Vision, Modeling, and Visualization (2022)
J. Bender, M. Botsch, and D. Keim (Eds.)

A Moving Least Squares Material Point Method for varied porous material interactions and non-sticky coupling of phases

A. M. Nilles[1] and Prof. Dr. Stefan Müller[1]

[1]University of Koblenz · Landau

Abstract

The Material Point Method (MPM) has become very popular in computer graphics due to its capability to handle a variety of materials and ease of coupling for multi-material simulations. However, MPM suffers from numerical stickyness, which is especially apparent in fluid-solid coupling. Furthermore, the free-flowing nature of fluids can cause issues for simulating immiscible fluids, leading to improper separation of phases especially at lower resolutions. Furthermore, some MPM formulations suffer from unwanted dissipation, noise or instability. Our MPM framework solves the latter using the Moving Least Squares MPM, while the former is addressed by coupling two grids only on contact, with an optional friction term for tangential coupling. This is further enhanced with a buoyancy penalty force that can achieve clean separation of immiscible fluids even at low resolutions. We combine this with a method for porous solids which we generalize in order to allow for highly varied material interactions.

CCS Concepts

• ***Computing methodologies*** → ***Physical simulation;***

1. Introduction

MPM is a hybrid Lagrangian/Eulerian method that has become increasingly popular in computer graphics. Particle advection addresses mass conservation, while momentum equations are handled by the grid. Due to the momentum transfer of particles through the grid and back, the method automatically handles coupling of different materials and can also be easily coupled with other simulations [JSS*15]. Usage of multiple grids allows for overlapping phases, i.e. to model porous materials [TGK*17]. These properties make MPM very useful for varied multi-material simulations. Common problems of MPM such as its dissipative nature and the *ringing instability* can be addressed using the Affine Particle-In-Cell method [JSS*15] (APIC) or MLS-MPM [HFG*18]. Another issue with MPM is its inherent numerical stickyness of the coupling [SSJ*14].

In this work, we present a multi-grid MLS-MPM framework for varied material interaction. We adopt and extend the ideas from [TGK*17] for porous sand, generalizing their ideas to new forms of interactions. We derive a generalized coupling between grids similiar to [YLCH18] which also supports tangential coupling via friction, removing the problem of numerical stickyness. To achieve better separation of immiscible fluids, we introduce a buoyancy penalty force that is effective even for simulations at lower resolutions. We adapt the approaches to miscible fluids and dissolution from [YLCH18] to our own framework and extend this by allowing dissolution of porous materials. Our framework allows great freedom in the specification of how materials interact and change their properties based on another phase, allowing for the simulation of varied effects.

2. Related Work

Stomakhin, Schroeder, Chai, et al. [SSC*13] introduced MPM to computer graphics. They used the method in order to model snow, for which they combined the fixed-corotated [SHST12] elasticity with a new plasticity model. Momentum transfer was handled as a blend between the Fluid Implicit Particle [BKR88] (FLIP) and Particle-In-Cell [HER55] (PIC) formulations. Later on, their elasto-plastic constitutive model was extended and used in a MPM for melting, heat transport and phase-change, which also supported incompressible materials [SSJ*14].

MPM has been used for a variety of material behaviours. This includes porous interaction of wet and dry sand with water [TGK*17], viscoelastic fluids, foams, sponges and even cloth [RGJ*15; YSB*15; JGT17] and many more materials and multi-material interactions [FQL*20; WFL*19; NSS*19; FLGJ19]. MPM continues to be actively researched to this date [LCS22; FCK22; TM22; FHW21].

Jiang, Schroeder, Selle, et al. [JSS*15] address the dissipative nature of PIC as well as the noisy, unstable nature of FLIP with their APIC method by describing particle velocities locally affine instead of constant. MLS-MPM [HFG*18] works very similiarly, but reduces computational costs by merging steps of the computation and avoiding the computation of gradients. The au-

DOI: 10.2312/vmv.20221214

thors additionally achieved two-way coupling with rigid bodies, as well as material cutting. GAO, WANG, WU, et al. [GWW*18] parallelized MLS-MPM on the GPU and achieved faster computations than with FLIP.

The MPM coupling is numerically sticky. YAN, LI, CHEN, and HU [YLCH18] address this by using multiple grids and coupling only when collisions occur, leaving tangential velocities discontinuous. Their approach also allows for miscible fluids via diffusion, as well as dissolution of solids and fluids. FANG, QU, LI, et al. [FQL*20] use a more sophisticated approach to address stickyness with a ghost matrix operator-splitting scheme. Their method supports coupling non-linear solids and incompressible fluids and is not restricted to explicit time integration.

3. Material point method

In computer graphics, MPM is most commonly used to simulate deformable solids. The laws of conservation of mass and momentum for this scenario can be expressed with the following governing equations:

$$\frac{D\rho}{Dt} = -\rho(\nabla \cdot \boldsymbol{v}), \quad \rho\frac{D\boldsymbol{v}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \rho\boldsymbol{g} \tag{1}$$

where ρ is the density, $\boldsymbol{v}$ the velocity, $\boldsymbol{\sigma}$ the Cauchy stress and $\boldsymbol{g}$ is the gravity. For incompressible non-viscous fluids, the governing equations are instead:

$$\frac{D\rho}{Dt} = 0, \quad \rho\frac{D\boldsymbol{v}}{Dt} = -\nabla p + \rho\boldsymbol{g} \tag{2}$$

where p is pressure. Equation (2) can be viewed as a special case of Eq. (1), where we define $\boldsymbol{\sigma} = -p\boldsymbol{I}$, along with the additional incompressibility constraint. The divergence of $\boldsymbol{\sigma}$ is then equivalent to the pressure gradient in Eq. (2) [TGK*17]. Section 3.2 will outline how we extend this to handle viscosity.

In the context of MPM, constitutive models are often defined using an elasto-plastic energy density function $\Psi(\boldsymbol{F}_E, \boldsymbol{F}_P)$, where $\boldsymbol{F}_E$ and $\boldsymbol{F}_P$ are the elastic and plastic portions of the deformation gradient $\boldsymbol{F}$. This relates to the Cauchy stress as

$$\boldsymbol{\sigma} = \frac{1}{\det(\boldsymbol{F}_E)} \frac{\partial\Psi}{\partial\boldsymbol{F}_E} \boldsymbol{F}_E^T. \tag{3}$$

MPM solves these equations using a hybrid Lagrangian/Eulerian approach. Velocity, mass and deformation gradient are stored in particles (the *material points*), which makes the material derivates in Eq. (1) simple to solve, but complicates the computation of derivatives such as $\nabla \cdot \boldsymbol{\sigma}$. These derivatives are instead computed using an Eulerian grid with grid basis functions that interpolate between particles and the grid. As such, the method consists of a particle-to-grid (P2G) transfer step that evaluates forces according to Eq. (1), followed by a grid-to-particle (G2P) transfer step that translates the grid forces back onto particles and evolves the deformation gradients. In the following, we outline our single-grid MPM framework in more detail, which provides the core for our multi-grid simulation described in Section 4.

3.1. Single-grid material point method

Our MPM framework is based on MLS-MPM [HFG*18] and the work of STOMAKHIN, SCHROEDER, CHAI, et al. [SSC*13] who introduced the method to computer graphics. We only use explicit time integration. Our algorithm is structured into four major steps: particle initialization, P2G, grid operations and G2P.

Particle initialization We sample particles using poisson disk sampling with a default relative spacing of 0.5 grid units per dimension, yielding 8 particles per cell. Masses m_p are determined by evenly distributing the mass of the sampled volume among particles. Each particle p is assigned a constitutive model Ψ_p, which includes the material parameters for that model. Furthermore, the user can specify an initial velocity, which can either be a constant or a function of the initial particle position. The initial volume V_p^0 of each particle is then estimated by computing grid cell masses $m_{\boldsymbol{i}}$ using a P2G step, estimating the cell density with the grid spacing h as $\rho_{\boldsymbol{i}}^0 = m_{\boldsymbol{i}}^0/h^3$ and transferring this back in a G2P step:

$$\rho_p^0 = \sum_{\boldsymbol{i}} \rho_{\boldsymbol{i}}^0 \omega_{\boldsymbol{i}p}^0, \quad V_p^0 = m_p/\rho_p^0 \tag{4}$$

where $\omega_{\boldsymbol{i}p}$ is the weight between cell $\boldsymbol{i}$ and particle p according to the grid basis functions.

Particle-to-grid The P2G step transfers masses and velocities to the grid and computes grid forces. Masses are computed as:

$$m_{\boldsymbol{i}}^n = \sum_p m_p \omega_{\boldsymbol{i}p}^n. \tag{5}$$

MLS-MPM allows to combine the momentum transfer and force calculation by first calculating the matrix

$$\boldsymbol{Q}_p = \Delta t M_p^{-1} V_p^0 \frac{\partial\Psi_p}{\partial\boldsymbol{F}_E} (\boldsymbol{F}_{Ep}^n)^T + m_p \boldsymbol{C}_p^n \tag{6}$$

where M_p is $\frac{1}{4}h^2$ or $\frac{1}{3}h^2$ for quadratic and cubic B-spline basis functions, respectively. $\boldsymbol{C}_p^n$ is the additional affine matrix from APIC [JSS*15]. Using $\boldsymbol{Q}$, grid momentum is estimated as

$$m_{\boldsymbol{i}}^n \boldsymbol{v}_{\boldsymbol{i}}^\star = \sum_p \omega_{\boldsymbol{i}p}^n \left(m_p \boldsymbol{v}_p^n + \boldsymbol{Q}_p(\boldsymbol{x}_{\boldsymbol{i}} - \boldsymbol{x}_p^n) \right). \tag{7}$$

Grid operations Grid velocities are computed by dividing Eq. (7) by grid mass and gravity is applied. We then apply the same algorithm described in [SSC*13] to achieve one-way coupling with rigid bodies using animated level sets. The result of this are the final collided grid velocities $\boldsymbol{v}_{\boldsymbol{i}}^{n+1}$. STOMAKHIN, SCHROEDER, CHAI, et al. also apply this collision algorithm directly on particles at the end of the simulation step. While we implemented this, we do not use it because this bypasses the deformation gradient update. The errors introduced by this are mostly apparent for free-flowing materials such as fluids and lead to clumping of particles around collision surfaces (see Fig. 1).

Grid-to-particle The G2P step transfers grid momentum onto particles, evolves the deformation gradients and calculates the affine matrix $\boldsymbol{C}_p^{n+1}$. Velocities are updated according to APIC, which is equivalent for regular PIC for this step:

$$\boldsymbol{v}_p^{n+1} = \sum_{\boldsymbol{i}} \omega_{\boldsymbol{i}p}^n \boldsymbol{v}_{\boldsymbol{i}}^{n+1}. \tag{8}$$

The affine matrix is updated as

$$\boldsymbol{C}_p^{n+1} = M_p^{-1} \sum_{\boldsymbol{i}} \omega_{\boldsymbol{i}p}^n \boldsymbol{v}_{\boldsymbol{i}}^{n+1} (\boldsymbol{x}_{\boldsymbol{i}} - \boldsymbol{x}_p^n)^T \tag{9}$$

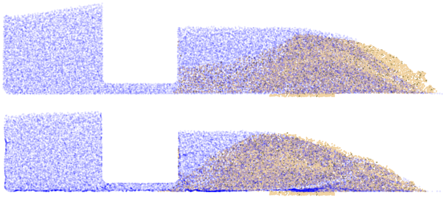

Figure 1: *A 2D dam-break scenario. The top does not use additional particle collisions, while the bottom does. The errors caused by this can be seen as dark blue spots where many particles are clumped together.*

and deformation gradients as

$$\hat{\boldsymbol{F}}_{Ep}^{n+1} = (\boldsymbol{I} + \Delta t \boldsymbol{C}_p^{n+1})\boldsymbol{F}_{Ep}^{n}, \tag{10}$$

which uses $\boldsymbol{C}_p^{n+1}$ to estimate $\nabla \boldsymbol{v}_p^{n+1}$ according to MLS-MPM. $\hat{\boldsymbol{F}}_{Ep}^{n+1}$ is then treated as defined by the plasticity model to yield the actual elastic portion of the deformation gradient and update the plastic portion. Note that for fluids, full deformation gradients are not necessary and this step can be simplified to an update of only the determinant (see [TGK*17] for details). At the end of the step, particles are advected according to their velocities.

3.2. Constitutive models

For hyperelasticity, we implemented the Neo-Hookean model, a model based on Hencky strain, as well as the fixed corotated energy density from [SHST12]. We use two different plasticity models, namely the snow model from [SSC*13], which is paired with the fixed corotated model, although we allow to use any of the other hyperelastic models in combination with it. Furthermore, we implemented the model for dry and wet sand from [TGK*17], using the regular energy density function instead of the unilateral one that was necessary due to their semi-implicit scheme.

We handle non-viscous fluids the same way as TAMPUBOLON, GAST, KLÁR, et al. [TGK*17] in a weakly-compressible manner. In order to extend this with viscosity, first consider the respective governing equations for incompressible fluids:

$$\frac{D\rho}{Dt} = 0, \quad \rho \frac{D\boldsymbol{v}}{Dt} = -\nabla p + \mu \nabla \cdot (\nabla \boldsymbol{v}) + \rho \boldsymbol{g} \tag{11}$$

where μ is the dynamic viscosity. As MPM computes $\nabla \cdot \boldsymbol{\sigma}$, we can define

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \mu \nabla \boldsymbol{v} \tag{12}$$

and due to our usage of MLS-MPM, $\nabla \boldsymbol{v}$ is simply the affine matrix $\boldsymbol{C}$. This approach is however very simple and does not support highly viscous fluids without requiring unreasonably small timesteps.

4. Multi-grid coupling for MPM

While single-grid MPM is already a powerful method for multi-material simulations due to automatically handling collisions between different materials via the grid, it does come with some limitations. The numerical scheme of MPM causes collisions to be sticky, i.e. fluid particles can get stuck on a solid object because its momentum will have a much bigger impact on the grid momentum compared to a few individual fluid particles. For very similiar reasons, the method cannot handle the separation of different free-flowing phases very well, because thin strands or individual particles of a fluid are not sufficiently sampled to stay neatly separated from another phase. Lastly, single-grid MPM does not allow for overlapping phases.

To address these issues, we start by adapting the approach from [TGK*17] for porous materials to our MLS-MPM framework with a few extensions (Section 4.1). We derive a relationship between the drag-based momentum exchange between two grids and the default single-grid coupling of MPM. Using these results, we then define a modification of the single-grid coupling for multiple grids which yields non-sticky coupling with control over friction and which allows for overlap (Section 4.2). We then adapt the approaches for fluid-solid coupling, immiscible and miscible fluid-fluid coupling and dissolution from [YLCH18] to our own coupling algorithm (Sections 4.2 to 4.5).

4.1. Porous interaction of fluids and solids

In order to facilitate porous materials that can overlap with a fluid phase, TAMPUBOLON, GAST, KLÁR, et al. [TGK*17] treat the solid and fluid phase separately, each with their own background grid. For our algorithm this means applying the method from Section 3.1 independently for both phases. Interaction between the two phases is governed by momentum exchange terms [TGK*17], which were adopted from [MAS*10]:

$$\boldsymbol{p}^s = c_E(\boldsymbol{v}^w - \boldsymbol{v}^s) + p^w \nabla \phi^w, \quad \boldsymbol{p}^w = -\boldsymbol{p}^{\boldsymbol{s}}. \tag{13}$$

Here, s denotes the solid phase and w the fluid phase. The drag coefficient c_E is derived based on material parameters of the sand as well as gravity (see [TGK*17] for details) and $\phi^w = \frac{\rho^w}{\rho^w + \rho^s}$ is the fluid volume fraction. The left part of the momentum exchange term is dissipative and similiar to Coulomb-friction [MAS*10] and models viscous forces due to sand particles that move through the fluid. Our discretization of this is equivalent to [TGK*17] and happens during the grid operations step by calculating an additional force:

$$\boldsymbol{f}_{\boldsymbol{i}}^s = c_E m_{\boldsymbol{i}}^{s,n} m_{\boldsymbol{i}}^{w,n} (\boldsymbol{v}_{\boldsymbol{i}}^{w,n} - \boldsymbol{v}_{\boldsymbol{i}}^{s,n}), \quad \boldsymbol{f}_{\boldsymbol{i}}^w = -\boldsymbol{f}_{\boldsymbol{i}}^s. \tag{14}$$

The right part is the buoyancy term. This term however is not dissipative and thus was not used in [MAS*10]. TAMPUBOLON, GAST, KLÁR, et al. [TGK*17] did implement it, but did not use for the majority of their examples. Due to our usage of MLS-MPM, which avoids computation of $\nabla \omega_{\boldsymbol{i}p}^n$, our discretization of this term is different from [TGK*17]:

$$\boldsymbol{f}_{\text{buoy}} = m_{\boldsymbol{i}}^{s,n} m_{\boldsymbol{i}}^{w,n} \sum_p -p_p^{w,n} M_p^{-1} \omega_{\boldsymbol{i}p}^n \frac{m_{\boldsymbol{i}}^{w,n}}{m_{\boldsymbol{i}}^{s,n} + m_{\boldsymbol{i}}^{w,n}} \left(\boldsymbol{x}_{\boldsymbol{i}} - \boldsymbol{x}_p^n\right). \tag{15}$$

This force is added to $\boldsymbol{f}_{\boldsymbol{i}}^w$ and subtracted from $\boldsymbol{f}_{\boldsymbol{i}}^s$. The non-dissipative nature of this can however easily lead to an unstable simulation. Furthermore, Eq. (15) is another P2G step, which is the most computationally expensive part of the method. For these reasons, we came to the same decision as previous authors and did not use it for the results produced in this paper.

An issue with Eq. (14) for our explicit time integration is that large values of c_E cause the drag exchange to be non-dissipative and instead accelerate the two phases away from each other. We derive a maximum allowed value $c_{E,\max,\boldsymbol{i}}$ by solving for the point at which the applied force causes both phases velocities to be equal:

$$\boldsymbol{v}_{\boldsymbol{i}}^{s} + \Delta t \cdot c_E m_{\boldsymbol{i}}^{w} (\boldsymbol{v}_{\boldsymbol{i}}^{w} - \boldsymbol{v}_{\boldsymbol{i}}^{s}) = \boldsymbol{v}_{\boldsymbol{i}}^{s} - \Delta t \cdot c_E m_{\boldsymbol{i}}^{s} (\boldsymbol{v}_{\boldsymbol{i}}^{w} - \boldsymbol{v}_{\boldsymbol{i}}^{s}). \tag{16}$$

The solution to this is

$$c_{E,\max,\boldsymbol{i}} = \frac{1}{\Delta t (m_{\boldsymbol{i}}^{w} + m_{\boldsymbol{i}}^{s})}, \quad \Delta t_{\max,\boldsymbol{i}} = \frac{1}{c_E (m_{\boldsymbol{i}}^{w} + m_{\boldsymbol{i}}^{s})} \tag{17}$$

and we clamp c_E to $c_{E,\max,\boldsymbol{i}}$ for each individual grid cell. An interesting result of this is that if we set the drag coefficient to always be the maximum, this is exactly the same coupling as single-grid MPM where we have simply partitioned the set of particles onto two grids which are later combined.

TAMPUBOLON, GAST, KLÁR, et al. [TGK*17] used their algorithm to model dam-break scenarios. The sand of the solid phase is assumed to be initially wet to a degree that leads to high cohesion, where any additional fluid instead lowers cohesion. This is facilitated by interpolating the initial cohesion $c_{Cp}^{s,0}$ with 0 based on the fluid volume fraction ϕ:

$$c_{Cp}^{s,n+1} = c_{Cp}^{s,0}(1 - \phi_p^{n+1}), \tag{18}$$

where the volume fraction is discretized as

$$\phi_{\boldsymbol{i}}^{n+1} = \begin{cases} 1 & m_{\boldsymbol{i}}^{w,n+1} > 0 \text{ and } m_{\boldsymbol{i}}^{s,n+1} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

$$\phi_p^{n+1} = \sum_{\boldsymbol{i}} \omega_{\boldsymbol{i}p}^{n} \phi_{\boldsymbol{i}}^{n+1}. \tag{20}$$

We generalized their ideas by first allowing both phases to consist of arbitrary and multiple materials with different material properties, thus not being restricted to sand and water. The user is given control over how the fluid volume fraction changes material parameters and which of the parameters are affected by specifying different material interactions to Eq. (18). This could be gaining instead of losing cohesion for sand, changes of the Lamé-parameters, or even using it alongside materials such as snow and changing parameters of the plasticity model, to name a few examples. Furthermore, we allow changes of the material parameters not only for the solid phase but also the fluid phase.

Another limitation of the model is that it does not consider capillary action and there is no means for sand to store water and slowly dry out over time. We thus introduce a new property, the *effective volume fraction* $\phi_{p,\text{eff}}$, which is used to change material parameters instead of the current volume fraction. $\phi_{p,\text{eff}}$ is stored on particles and can be computed freely based on its value in the previous timestep and the current volume fraction. Examples include:

$$\phi_{p,\text{eff}}^{n+1} = \max(\phi_{p,\text{eff}}^{n}, \phi_p^{n+1}), \tag{21}$$

which causes the solid to never dry and

$$\phi_{p,\text{eff}}^{n+1} = (1 - \alpha)\phi_{p,\text{eff}}^{n} + \alpha\phi_p^{n+1} \tag{22}$$

where α can be chosen based on the timestep, which causes the effective volume fraction to change slowly over time, allowing effects where the solid is slowly drying out. We give the user free control over how the effective volume fraction is calculated and combined with different material interaction terms, this leads to a wide variety of possible effects without overcomplicating the original method.

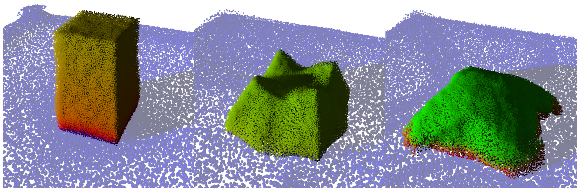

Figure 2: *A fluid column was initialized on top of a porous sand column and quickly drains. Sand cohesion is multiplied by $\sqrt{\phi_{eff}}$, which is visualized as the sand color and computed as in Eq. (22). Cohesion does not immediately change with the water fraction and the sand can retain its shape for longer with this method.*

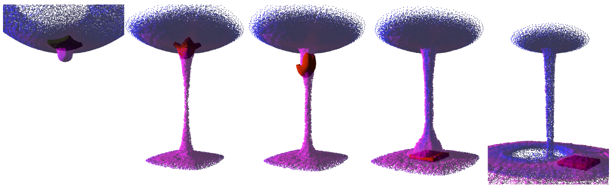

Figure 3: *A hyperelastic, porous sheet is placed over a drain. Contact with water slowly increases ϕ_{eff}, which increases elasticity, causing the sheet to eventually slip through the drain. At the same time, water gains viscosity due to contact with the sheet. ϕ_{eff} is calculated by combining Eqs. (21) and (22), i.e. it is slowly gained and then never decreases.*

4.2. Generalized multi-grid coupling

YAN, LI, CHEN, and HU [YLCH18] proposed a coupling approach that served as the motivation for our work and is very similiar to our approach. They derived the force of the collision process between two grids (phases) f_1 and f_2 to be

$$\boldsymbol{f}_{\boldsymbol{i},\text{col}}^{\prime} = \frac{\boldsymbol{p}_{\boldsymbol{i}}^{f_1} m_{\boldsymbol{i}}^{f_1} - \boldsymbol{p}_{\boldsymbol{i}}^{f_2} m_{\boldsymbol{i}}^{f_2}}{(m_{\boldsymbol{i}}^{f_1} + m_{\boldsymbol{i}}^{f_2})\Delta t} = \frac{\boldsymbol{v}_{\boldsymbol{i}}^{f_1} (m_{\boldsymbol{i}}^{f_1})^2 - \boldsymbol{v}_{\boldsymbol{i}}^{f_2} (m_{\boldsymbol{i}}^{f_2})^2}{(m_{\boldsymbol{i}}^{f_1} + m_{\boldsymbol{i}}^{f_2})\Delta t} \tag{23}$$

If we compare this to Eqs. (14) and (17), this looks very similiar to what we derived to be the correct force for collision based on the drag term from the previous section. However, if we insert Eq. (17) into Eq. (14), the correct result is actually:

$$\boldsymbol{f}_{\boldsymbol{i},\text{col}} = \frac{\boldsymbol{p}_{\boldsymbol{i}}^{f_1} m_{\boldsymbol{i}}^{f_2} - \boldsymbol{p}_{\boldsymbol{i}}^{f_2} m_{\boldsymbol{i}}^{f_1}}{(m_{\boldsymbol{i}}^{f_1} + m_{\boldsymbol{i}}^{f_2})\Delta t} \tag{24}$$

We believe Eq. (23) from [YLCH18] to be a typo. Using this equation lead to non-dissipative behaviour in our experiments and it does not seem correct from a physics viewpoint. In the original work, the authors ignore viscous forces during the collision, so their contact force was just a normal force. We generalized this idea and split our coupling into a normal and tangential force that can be controlled seperately:

$$\boldsymbol{f}_{\boldsymbol{i},\text{contact}}^{f_2} = -\boldsymbol{f}_{\boldsymbol{i},\text{contact}}^{f_1} = \beta(\boldsymbol{f}_{\boldsymbol{i},\text{col}} \cdot \boldsymbol{n}_{\boldsymbol{i}})\boldsymbol{n}_{\boldsymbol{i}} + \boldsymbol{f}_{\boldsymbol{i},\text{t}}. \tag{25}$$

Here, $\boldsymbol{n_i}$ is the interface normal, which is discretized differently from [YLCH18] in line with MLS-MPM as follows:

$$\bar{\boldsymbol{n}}_{\boldsymbol{i}}^{f_i} = \frac{\sum_{p\in P^{f_i}} m_p \omega_{\boldsymbol{i}p}^n \left(\boldsymbol{x_i} - \boldsymbol{x}_p^n\right)}{\left\|\sum_{p\in P^{f_i}} m_p \omega_{\boldsymbol{i}p}^n \left(\boldsymbol{x_i} - \boldsymbol{x}_p^n\right)\right\|} \tag{26}$$

$$\boldsymbol{n}_{\boldsymbol{i}}^{f_2} = -\boldsymbol{n}_{\boldsymbol{i}}^{f_1} = \frac{\bar{\boldsymbol{n}}_{\boldsymbol{i}}^{f_2} - \bar{\boldsymbol{n}}_{\boldsymbol{i}}^{f_1}}{\left\|\bar{\boldsymbol{n}}_{\boldsymbol{i}}^{f_2} - \bar{\boldsymbol{n}}_{\boldsymbol{i}}^{f_1}\right\|}, \tag{27}$$

where P^{f_i} is the set of particles associated with phase f_i. We left out M_p^{-1} in this discretization as it is constant for our grid basis functions and thus irrelevant due to the normalization.

The tangential portion $\boldsymbol{f}_{\boldsymbol{i},t}$ of our contact force is controlled by a friction parameter δ which allows for stickyness if the normal force is small enough:

$$\boldsymbol{f}_{\boldsymbol{i},n} = (\boldsymbol{f}_{\boldsymbol{i},\mathrm{col}} \cdot \boldsymbol{n_i})\boldsymbol{n_i} \tag{28}$$

$$\bar{\boldsymbol{f}}_{\boldsymbol{i},t} = \boldsymbol{f}_{\boldsymbol{i},\mathrm{col}} - \boldsymbol{f}_{\boldsymbol{i},n} \tag{29}$$

$$\boldsymbol{f}_{\boldsymbol{i},t} = \begin{cases} \bar{\boldsymbol{f}}_{\boldsymbol{i},t} & \text{if } \|\bar{\boldsymbol{f}}_{\boldsymbol{i},t}\| \leq \delta \cdot \|\boldsymbol{f}_{\boldsymbol{i},n}\| \\ \delta \cdot \|\boldsymbol{f}_{\boldsymbol{i},n}\| \frac{\bar{\boldsymbol{f}}_{\boldsymbol{i},t}}{\|\bar{\boldsymbol{f}}_{\boldsymbol{i},t}\|}, & \text{otherwise} \end{cases} \tag{30}$$

As a simpler alternative, $\boldsymbol{f}_{\boldsymbol{i},t}$ can be computed as $\delta\bar{\boldsymbol{f}}_{\boldsymbol{i},t}$. Similiar to [YLCH18], we can restrict the contact force to only apply when a collision is detected. In order to make our coupling more general we also allow to have it apply always, which makes it possible to represent the default single-grid coupling as well as the drag-based coupling from Section 4.1 in one unified coupling model. For solid/fluid or solid/solid coupling, our approach can be used to improve on the numerically sticky single-grid coupling by restricting the contact force to collisions and choosing $\beta = 1$ and $\delta \in [0,1]$. Unfortunately, this approach to coupling is not sufficient for immiscible fluids, which we adress in the following section.

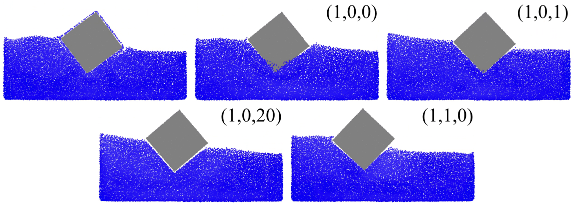

Figure 4: *A cube of low density hit by a water column. The upper left image uses fully coupled single-grid MPM, demonstrating the numerical stickyness. The remaining images use our generalized multi-grid coupling, where coupling is restricted to collisions. The parameters (β, δ, c_b) are shown in the upper right for each image.*

Yan, Li, Chen, and Hu [YLCH18] needed their approach to prevent penetration even for solid-solid and fluid-solid coupling. This is due to their FLIP/PIC interpolation, as FLIP allows particles to retain some of their velocities which circumvents the grid coupling. As we use MLS-MPM, which is similiar to APIC, our single-grid approach already properly couples all particles and penetration does not occur. Our reason for using this approach for solid-solid and fluid-solid coupling is solely in order to address numerical stickyness.

4.3. Immiscible fluids

For immiscible fluids, Yan, Li, Chen, and Hu [YLCH18] used $\beta = 0.3$ and only apply their contact force if a collision is detected. Applying this to our own coupling, this also means $\delta = 0$ as they did not have a tangential component for their contact force. These parameters effectively mean that the two fluid phases behave similiar to being porous on collisions, in other words the phases will have overlap. If we let $\beta = 1$ as we do for solid/fluid and solid/solid coupling, the overlap is avoided in theory. Unfortunately, this still suffers from similiar issues as attempting to handle immiscible fluids in single-grid MPM. The fluid tends to form thin strands or small droplets of liquid and this leads to particles from both phases ending up finely mixed. An accompanying issue is that the estimated normals become unreliable because of a more sparsely sampled volume. While restricting the coupling to collisions using normals does help in letting the denser liquid pass through to the bottom, proper separation is not achieved with any values for β, as the coupling is dissipative and thus cannot actively counteract overlap that already happened due to the described numeric limitations.

We solve this issue by introducing an *artificial buoyancy* by means of a penalty force counteracting gravity. Particles are enhanced with their respective rest density $\rho_{0,p}$ which is rasterized in a P2G step onto grid cells as

$$m_{\boldsymbol{i}}^{f_i}\rho_{0,\boldsymbol{i}}^{f_i} = \sum_{p\in P^{f_i}} \omega_{\boldsymbol{i}p}^n m_p \rho_{0,p}. \tag{31}$$

The buoyancy force is $\boldsymbol{f}_b = -\boldsymbol{g}\rho V$, given the density of the liquid ρ and the displaced volume V. We use the rest density and mass of one phase to estimate the displaced volume and the cell density of the other phase for ρ. This yields our buoyancy force as

$$\boldsymbol{f}_{\boldsymbol{i},b} = c_b \max\left(\rho_{\boldsymbol{i}}^{f_2}\frac{m_{\boldsymbol{i}}^{f_1}}{\rho_{0,\boldsymbol{i}}^{f_1}}, \rho_{\boldsymbol{i}}^{f_1}\frac{m_{\boldsymbol{i}}^{f_2}}{\rho_{0,\boldsymbol{i}}^{f_2}}\right)\boldsymbol{g} \tag{32}$$

where c_b is a parameter used to scale the strength of this penalty force. We apply this force to both phases with opposite directions, where the phase with the higher rest density is pushed in the direction of gravity. Smaller values for β allow particles to more easily slip past each other to separate, while also mitigating some of the artifacts caused by high values for c_b that are needed for quick separation. However, a small β results in more overlap in situations where the two phases collide in a direction orthogonally to gravity. Nevertheless, our approach facilitates perfect separation of phases as the fluids come to rest.

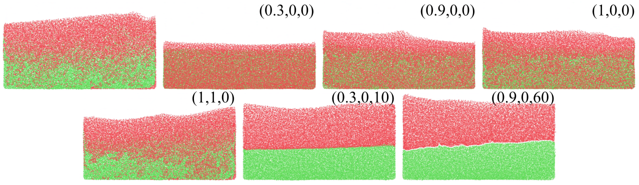

Figure 5: *Two columns of immiscible fluids with different densities collide. The upper left image uses fully coupled single-grid MPM. The remaining images use our generalized multi-grid coupling, where coupling is restricted to collisions. The parameters (β, δ, c_b) are shown in the upper right for each image.*

4.4. Miscible fluids

Yan, Li, Chen, and Hu [YLCH18] approach miscible fluids with the diffusion model

$$\frac{D\alpha^{f_i}}{Dt} = k_d \nabla^2 \alpha^{f_i}, \tag{33}$$

with diffusion coefficient k_d and concentration α^{f_i} of fluid phase f_i, where concentrations summed over all phases are 1. Our discretization of this is again different due to MLS-MPM:

$$m_{\boldsymbol{i}}^{f_i} \nabla \alpha_{\boldsymbol{i}}^{f_i} = \frac{\sum_p m_p \alpha_p^{f_i} \omega_{\boldsymbol{i}p}^n \left(\boldsymbol{x}_{\boldsymbol{i}} - \boldsymbol{x}_p^n\right)}{\left\| \sum_p m_p \alpha_p^{f_i} \omega_{\boldsymbol{i}p}^n \left(\boldsymbol{x}_{\boldsymbol{i}} - \boldsymbol{x}_p^n\right) \right\|} \tag{34}$$

$$\nabla^2 \alpha_p^{f_i} = \sum_{\boldsymbol{i}} M_p^{-1} \omega_{\boldsymbol{i}p}^n \nabla \alpha_{\boldsymbol{i}}^{f_i} \cdot \left(\boldsymbol{x}_{\boldsymbol{i}} - \boldsymbol{x}_p^n\right) \tag{35}$$

which involves an additional P2G and G2P step. Without the normalization of $\nabla \alpha_{\boldsymbol{i}}^{f_i}$ we experienced concentration changes even in situations where concentration was constant across particles. We update particle masses and rest densities the same way as [YLCH18]. Unlike the original paper, we also added viscosity to our fluids, which allows us to mix fluids of varying viscosity and blend their viscosity values as they mix.

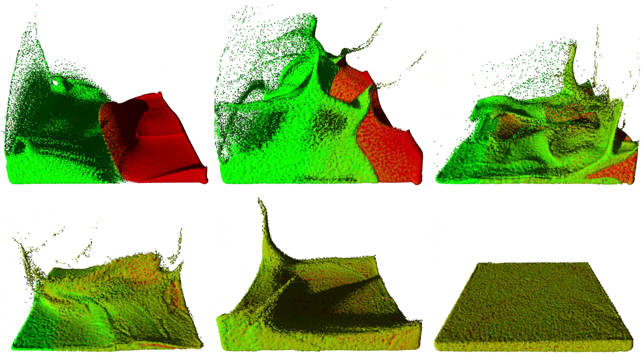

Figure 6: *Two columns of miscible fluids with different viscosity collide, resulting in a mixture with viscosity in between both. Artifacts of our diffusion method are visible as an unnatural pattern in which the concentrations seem to propagate and in the final picture (lower right), there are still particles left over that have not properly mixed.*

4.5. Dissolution

We adopt the dissolution approach of [YLCH18] as-is. The discretization stays the same for our MLS-MPM framework as it does not depend on gradients of the grid basis functions. For details, we refer to the original work in [YLCH18]. We use the approach for miscible fluids from Section 4.4 to diffuse the concentration of the dissolved solid inside the liquid, which allows for faster dissolution. Furthermore, the solid phase can use our algorithm for porous solids from Section 4.1. Since porous materials can be penetrated by the fluid phase, they are also able to dissolve faster. It is possible to change viscosity of the fluid phase based on the concentration of the dissolved solid.

5. Implementation

We parallelized our method on the GPU using OpenGL 4.6. P2G is done as a scattering algorithm using layered point rendering with additive blending, grid operations are done in a compute shader and G2P is a gathering algorithm in a compute shader. Our implementation is however not optimized and thus not the focus of this work. The P2G step tends to be the most expensive part of MPM, especially for our naive implementation, which means that computation of grid normals and concentration gradients is costly. On the other hand, the parts of our method that extend the G2P or grid operations have a smaller impact on performance. For this reason, performance largely depends on how well optimized the P2G is. Gao, Wang, Wu, et al. [GWW*18] massively reduced the cost of the P2G step, which would make our extensions significantly cheaper. More advanced and optimized GPU-parallel MPM algorithms can be found in the literature [GWW*18; WQS*20].

6. Results

We discuss our results using six different scenes, organized into the categories *porous materials*, *solid-fluid coupling*, *immiscible fluids*, *miscible fluids* and *dissolution*. The supplementary material contains the videos for all of the scenes and figures in this paper.

Porous materials For our first scene (Fig. 2), we initialize a sand column and water column at the same location. The effective volume fraction is initialized to 1 and changes according to Eq. (22) with $\alpha = \Delta t$. Sand cohesion is multiplied by $\sqrt{\phi_{\text{eff}}}$, which is opposite to what was done in [TGK*17]. As the water drains, the effective volume fraction decreases at a slower pace, as indicated by the color change from red to green in Fig. 2. Cohesion decreases over time after no water is present in the cell, and once it passes a threshold, the column starts collapsing. The sand continues to lose cohesion, causing further sandslides until cohesion reaches 0. With the original method from [TGK*17], the sand would instead immediately return to zero cohesion as the water drains and start collapsing right away.

Our second scene (Fig. 3) demonstrates how we extend porous interaction to other materials. A porous hyperelastic sheet is placed on top of a drain and water is poured on top if it. The sheet is initially capable of bridging the drain, but as ϕ_{eff} increases, we decrease its Young's modulus and it eventually slips through the gap. At the same time, fluid viscosity is increased with ϕ_{eff}, causing it to gain viscosity as it passes through the porous sheet. For this scene we compute ϕ_{eff} as

$$\phi_{p,\text{eff}}^{n+1} = \max(\phi_{p,\text{eff}}^n, (1-\Delta t)\phi_{p,\text{eff}}^n + \Delta t \phi_p^{n+1}). \tag{36}$$

Solid-fluid coupling We demonstrate our generalized multi-grid coupling using a 2D scene (Fig. 4). A hyperelastic cube is hit by a column of fluid, with a density ratio of 2 : 5. The default single-grid coupling of MPM leads to numeric stickyness and causes fluid particles to stick to the cube. If we restrict coupling to collisions and fully couple the normal forces, stickyness is prevented, but some fluid particles can penetrate the solid due to slight inconsistensies in the normals. This can be mitigated using our buoyancy penalty force that we proposed for immiscible fluids. If c_b is too large, this causes a gap between solid and fluid, and fluid particles that hit the top part of the solid are sucked through the solid. If we use our proposed tangential coupling, both penetration and numerical stickyness are prevented even without the additional buoyancy force.

Immiscible fluids We show how different parameters for our coupling approach affect immiscible fluids with a 2D scene containing two colliding water columns with a density ratio of 2 : 5 (Fig. 5). A fully-coupled single-grid simulation cannot cleanly separate the phases and instead leads to a soft gradient of overlapping phases. Using the approach from [YLCH18] with $\beta = 0.3, 0.9$ and 1 results in even more overlap of the two phases, while our approach with $\beta = 1$ and $\delta = 1$ performs slightly better, but still worse than a simple single-grid simulation. These results are in line with our expectations but contradict the results shown in [YLCH18]. This contradiction could be due to differences in our MPM discretization and fluid model. However, overlap with $\beta < 1$ is to be expected as it is similiar to the phases being porous, as described in Section 4.3. Restricting the coupling to only collisions has problems inside the volume, as normal directions are unreliable except for surfaces. It is possible that [YLCH18] accounts for this by not coupling the overlapping volumes, but this would require some form of interface tracking. Our buoyancy penalty force manages to cleanly separate both phases, with only temporary overlaps. For very large values of c_b, especially in conjunction with a large β, it causes artifacts in form of a gap between phases as well as continuously generating motion. Despite this, even with unreasonably large c_b, velocities did not increase infinitely, thus simulation was still stable without changing our timestep.

Miscible fluids In our fifth scene (Fig. 6) we mix two fluid columns with different viscosity, generating a mixture with a blend of the original viscosity levels. Our adapted MLS-MPM discretization for the concentration gradient seems to cause some artifacts. The concentrations mix in an unnatural pattern and there are leftover particles that do not have fully mixed concentrations. We thus propose to instead use the original discretization from [YLCH18] that uses $\nabla\omega$ instead of differences, which is more accurate, but more expensive to compute [HFG*18]. Despite that, our extended fluid model with viscosity allows for additional interesting effects not captured by the original algorithm.

Dissolution For our last scene (Fig. 7), we dissolve a bunny using the snow model from [SSC*13] in water. The solid is modeled to be porous and the dissolved concentration is diffused in the liquid. These two additions allow for much faster dissolution, as the diffusion dissipates the concentration, thus allowing the fluid to further dissolve the solid, while also being able to dissolve the inside of the volume instead of being limited to the exposed surface thanks to the porous solid.

7. Conclusion

MPM proves to be a very versatile and flexible method for computer graphics. The MLS-MPM discretization greatly improves momentum conservation, while mitigating the computational costs of APIC [JSS*15; HFG*18]. Support for viscous fluids was simple to add to the MPM formulation. Our extensions to the framework for porous materials from [TGK*17] allow for great freedom in designing material interactions and enable many interesting effects. Furthermore, our generalized multi-grid coupling with normal and tangential contact forces improves on the standard single-grid coupling, getting rid of the numerical stickyness that is a core issue in MPM. Our buoyancy penalty force is very effective in solving issues with overlapping phases, most notably for immiscible fluids, while still allowing for stable simulations despite being non-dissipative. Lastly, extension of the dissolution approach from [YLCH18] with our porous solid approach synergizes well and enables quicker dissolution.

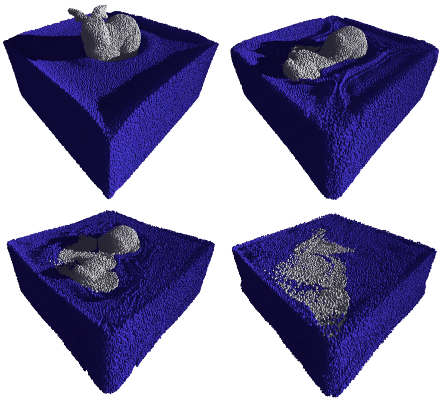

Figure 7: *A bunny with a snow material is dissolved in fluid. The snow was made slightly porous and diffusion of the dissolved solid in the liquid was enabled for this scene. In the lower right, the solid has fully dissolved and its particles are now part of the fluid phase.*

8. Limitations and future work

A core limitation of this work is the lack of support for very stiff and incompressible or nearly incompressible materials. Support for stiffer materials could be improved using a implicit or semi-implicit method, but this is non-trivial as [TGK*17] has shown that this can negatively impact material behaviour, which required an adapted energy density in their case. Incompressibility of the fluid would be simple using incompressible FLIP or APIC instead of our MPM formulation, which can easily be integrated with MPM thanks to the hybrid Lagrangian/Eulerian formulation. Solids are more complicated, but possible as shown in [SSJ*14], however it is not straightforward to extend their approach to arbitrary energy densities and materials.

The issues with our diffusion approach warrant further research. While we suspect the reason to be our modified discretization, we have yet to conclusively show that. Further research is required here, especially with respect to conservation of mass, as the current approach does not necessarily guarantee this.

Lastly, while dissolution works well, the current approach is one-way and the dissolved solid cannot return to a solid phase, i.e. precipitation is not possible. Furthermore, the change from solid to fluid phase is abrupt, as the particles switch their material immediately. Further research could focus on the blending and interpolation between different materials to make this process more smooth.

References

[BKR88] Brackbill, J.U., Kothe, D.B., and Ruppel, H.M. "Flip: A low-dissipation, particle-in-cell method for fluid flow". *Computer Physics Communications* 48.1 (1988), 25–38. issn: 0010-4655. doi: https://doi.org/10.1016/0010-4655(88)90020-3. url: https://www.sciencedirect.com/science/article/pii/0010465588900203 1.

[FCK22] Fan, Linxu, Chitalu, Floyd M., and Komura, Taku. "Simulating Brittle Fracture with Material Points". *ACM Trans. Graph.* 41.5 (May 2022). issn: 0730-0301. doi: 10.1145/3522573. url: https://doi.org/10.1145/3522573 1.

[FHW21] Feng, Kewei, Huang, Duruo, and Wang, Gang. "Two-layer material point method for modeling soil–water interaction in unsaturated soils and rainfall-induced slope failure". *Acta Geotechnica* 16.8 (Aug. 2021), 2529–2551. issn: 1861-1133. doi: 10.1007/s11440-021-01222-9. url: https://doi.org/10.1007/s11440-021-01222-9 1.

[FLGJ19] Fang, Yu, Li, Minchen, Gao, Ming, and Jiang, Chenfanfu. "Silly Rubber: An Implicit Material Point Method for Simulating Non-Equilibrated Viscoelastic and Elastoplastic Solids". *ACM Trans. Graph.* 38.4 (July 2019). issn: 0730-0301. doi: 10.1145/3306346.3322968. url: https://doi.org/10.1145/3306346.3322968 1.

[FQL*20] Fang, Yu, Qu, Ziyin, Li, Minchen, et al. "IQ-MPM: An Interface Quadrature Material Point Method for Non-Sticky Strongly Two-Way Coupled Nonlinear Solids and Fluids". *ACM Trans. Graph.* 39.4 (July 2020). issn: 0730-0301. doi: 10.1145/3386569.3392438. url: https://doi.org/10.1145/3386569.3392438 1, 2.

[GWW*18] Gao, Ming, Wang, Xinlei, Wu, Kui, et al. "GPU Optimization of Material Point Methods". *ACM Trans. Graph.* 37.6 (Dec. 2018). issn: 0730-0301. doi: 10.1145/3272127.3275044. url: https://doi.org/10.1145/3272127.3275044 2, 6.

[HER55] Harlow, Francis Harvey, Evans, Martha, and Richtmyer, Robert D. *A machine calculation method for hydrodynamic problems*. Los Alamos Scientific Laboratory of the University of California, 1955 1.

[HFG*18] Hu, Yuanming, Fang, Yu, Ge, Ziheng, et al. "A Moving Least Squares Material Point Method with Displacement Discontinuity and Two-Way Rigid Body Coupling". *ACM Trans. Graph.* 37.4 (July 2018). issn: 0730-0301. doi: 10.1145/3197517.3201293. url: https://doi.org/10.1145/3197517.3201293 1, 2, 7.

[JGT17] Jiang, Chenfanfu, Gast, Theodore, and Teran, Joseph. "Anisotropic Elastoplasticity for Cloth, Knit and Hair Frictional Contact". *ACM Trans. Graph.* 36.4 (July 2017). issn: 0730-0301. doi: 10.1145/3072959.3073623. url: https://doi.org/10.1145/3072959.3073623 1.

[JSS*15] Jiang, Chenfanfu, Schroeder, Craig, Selle, Andrew, et al. "The Affine Particle-in-Cell Method". *ACM Trans. Graph.* 34.4 (July 2015). issn: 0730-0301. doi: 10.1145/2766996. url: https://doi.org/10.1145/2766996 1, 2, 7.

[LCS22] Liang, Yong, Chandra, Bodhinanda, and Soga, Kenichi. "Shear band evolution and post-failure simulation by the extended material point method (XMPM) with localization detection and frictional self-contact". *Computer Methods in Applied Mechanics and Engineering* 390 (2022), 114530. issn: 0045-7825. doi: https://doi.org/10.1016/j.cma.2021.114530. url: https://www.sciencedirect.com/science/article/pii/S0045782521007210 1.

[MAS*10] Mackenzie-Helnwein, P., Arduino, P., Shin, W., et al. "Modeling strategies for multiphase drag interactions using the material point method". *International Journal for Numerical Methods in Engineering* 83.3 (2010), 295–322. doi: https://doi.org/10.1002/nme.2823. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2823. url: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2823 3.

[NSS*19] Nagasawa, Kentaro, Suzuki, Takayuki, Seto, Ryohei, et al. "Mixing Sauces: A Viscosity Blending Model for Shear Thinning Fluids". *ACM Trans. Graph.* 38.4 (July 2019). issn: 0730-0301. doi: 10.1145/3306346.3322947. url: https://doi.org/10.1145/3306346.3322947 1.

[RGJ*15] Ram, Daniel, Gast, Theodore, Jiang, Chenfanfu, et al. "A Material Point Method for Viscoelastic Fluids, Foams and Sponges". *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. SCA '15. Los Angeles, California: Association for Computing Machinery, 2015, 157–163. isbn: 9781450334969. doi: 10.1145/2786784.2786798. url: https://doi.org/10.1145/2786784.2786798 1.

[SHST12] Stomakhin, Alexey, Howes, Russell, Schroeder, Craig, and Teran, Joseph M. "Energetically Consistent Invertible Elasticity". *Proceedings of the 11th ACM SIGGRAPH / Eurographics Conference on Computer Animation*. EUROSCA'12. Lausanne, Switzerland: Eurographics Association, 2012, 25–32. isbn: 9783905674378 1, 3.

[SSC*13] Stomakhin, Alexey, Schroeder, Craig, Chai, Lawrence, et al. "A Material Point Method for Snow Simulation". *ACM Trans. Graph.* 32.4 (July 2013). issn: 0730-0301. doi: 10.1145/2461912.2461948. url: https://doi.org/10.1145/2461912.2461948 1–3, 7.

[SSJ*14] Stomakhin, Alexey, Schroeder, Craig, Jiang, Chenfanfu, et al. "Augmented MPM for Phase-Change and Varied Materials". *ACM Trans. Graph.* 33.4 (July 2014). issn: 0730-0301. doi: 10.1145/2601097.2601176. url: https://doi.org/10.1145/2601097.2601176 1, 7.

[TGK*17] Tampubolon, Andre Pradhana, Gast, Theodore, Klár, Gergely, et al. "Multi-Species Simulation of Porous Sand and Water Mixtures". *ACM Trans. Graph.* 36.4 (July 2017). issn: 0730-0301. doi: 10.1145/3072959.3073651. url: https://doi.org/10.1145/3072959.3073651 1–4, 6, 7.

[TM22] Telikicherla, Ram Mohan and Moutsanidis, Georgios. "Treatment of near-incompressibility and volumetric locking in higher order material point methods". *Computer Methods in Applied Mechanics and Engineering* 395 (2022), 114985. issn: 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.114985. url: https://www.sciencedirect.com/science/article/pii/S0045782522002328 1.

[WFL*19] Wolper, Joshuah, Fang, Yu, Li, Minchen, et al. "CD-MPM: Continuum Damage Material Point Methods for Dynamic Fracture Animation". *ACM Trans. Graph.* 38.4 (July 2019). issn: 0730-0301. doi: 10.1145/3306346.3322949. url: https://doi.org/10.1145/3306346.3322949 1.

[WQS*20] Wang, Xinlei, Qiu, Yuxing, Slattery, Stuart R., et al. "A Massively Parallel and Scalable Multi-GPU Material Point Method". *ACM Trans. Graph.* 39.4 (July 2020). issn: 0730-0301. doi: 10.1145/3386569.3392442. url: https://doi.org/10.1145/3386569.3392442 6.

[YLCH18] Yan, X., Li, C-F., Chen, X-S., and Hu, S-M. "MPM simulation of interacting fluids and solids". *Computer Graphics Forum* 37.8 (2018), 183–193. doi: https://doi.org/10.1111/cgf.13523. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13523. url: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13523 1–7.

[YSB*15] Yue, Yonghao, Smith, Breannan, Batty, Christopher, et al. "Continuum Foam: A Material Point Method for Shear-Dependent Flows". *ACM Trans. Graph.* 34.5 (Nov. 2015). issn: 0730-0301. doi: 10.1145/2751541. url: https://doi.org/10.1145/2751541 1.