

Solid deformation by material point method

YUNTAO JIANG, TAO YANG, AND JIAN CHANG

Solid materials are responsible for many interesting phenomena. There are various types of them, such as deformable objects and granular materials. In this paper, we present an MPM based framework to simulate the wide range of solid materials. In this framework, solid mechanics is based on the elastoplastic model following small deformation theory. We use von Mises criterion for deformable objects, and the Drucker-Prager model with non-associated plastic flow rules for granular materials. As a result, we can simulate different kinds of deformation of deformable objects and sloping failure for granular materials.

1. Introduction

Solid materials exist everywhere in our daily life, and are responsible for many interesting phenomena. Deformable objects, such as chewing gum, toothpaste, and bread dough, undergo elastic and plastic deformation when pressed or stretched. Another kind of solid materials are granular materials, including many kinds of geomaterials such as sand, soil and rubbles, consist of numerous grains colliding and sliding against each other. The numerical simulation of these different materials has been a problem of long standing interest and challenge.

There have been many research on the simulation of deformation objects, either mesh-based methods like Finite Element Method (FEM) or meshless methods like Smoothed Hydrodynamic Particle (SPH) and Material Point Method (MPM). As for granular materials, given the extremely large number of grains in realistic scenes, it's not practical to simulate individual grain as a Lagrangian particle. Thus the continuum based approaches like SPH or MPM are better choices. Since we want to build a unified framework to simulate different kinds of materials, SPH and MPM approach seems to be the best choice. Here we choose the MPM approach, because the hybrid of Lagrangian particles and Eulerian grids brings us a more stable performance.

In engineering field, deformable objects and granular materials can both be described as elastoplastic materials, and they share the same elastic model when the deformation remains in the elastic range. Their behavior under plastic deformation, however, correspond with different constitutive models. We use the linear model for elastic deformation for all solid materials, and use different yield criterion for the plastic deformation of different materials, Von Mises criterion for deformable objects[17], and Drucker-Prager model with non-associated flow rule for granular materials[1].

In this work, we follow the small deformation theory to compute the strain in the materials. This approach has been applied widely in SPH based simulations in engineering field, due to its simple mathematical form while remaining as a good approximation to the real physical process. We also show the constitutive models which have been applied in SPH framework can be easily adapted to MPM framework and still work well. Besides, we adopt the APIC method [7] to transfer physical quantities between grid and particles, which preserves the linear and angular momentum and results in an ignorable dissipation rate.

2. Related work

Both SPH and MPM use a Lagrangian particle representation of materials, and they share very similar discretizations of physical equations. So here we first briefly introduce some SPH based works, and then some MPM based works. Mesh-based methods such as FEM, mass-string method, and Eulerian grid based method are beyond the scope, so are not discussed here.

2.1. SPH simulation

In engineering field, SPH was first applied to simulate elastic solid materials by Libersky and Petschek [9]. Gray et al. [6] extended this early work with a method for overcoming the tensile instability, which could otherwise lead to numerical fracture. These works are based on the small deformation theory, and the strain in solid materials is updated additively with velocity gradient.

To simulate different plastic materials, different yield criterions are used. Cleary et al. [4] used Von Mises plasticity and linear isotropic hardening to simulate elastoplastic deformation of deformable objects. Bui et al. [3] implemented the Drucker-Prager model with associated and non-associated plastic flow rules to simulate large deformation and post-failure of granular materials, and An et al. [1] extended this work to 3D cases.

In computer graphics, SPH has been widely used to simulate fluids([10], [12]), while it has also been recognized as a very potential method to simulate solid materials. A common choice is to directly compute the strain of solid materials by comparing the current shape of the solid materials with a reference shape, which is the original shape of the object without any force exerted on it. Müller et al. [11] proposed a particle-based method for elastic, plastic and melting solid materials. They use the SPH method to compute the mass, density and volume of phyxels, and use Green-Saint-Venant strain to determine the stress tensor. To approximate the Jacobian of the deformation vector field, a Moving Least Squares approach is employed. Solenthaler et al. [13] replaced the MLS approach with an SPH method, which can handle coarsely sampled and coplanar particle configurations. Becker et al. [2] extended their work with a corotational approach to correctly handle rotations. The methods using the reference shape can maintain the original shape well, but face additional difficulties when handling extremely large deformations and topological changes.

Yan et al. [18] presented an SPH approach to uniformly handle the interaction between elastoplastic solid and multiple fluids. Their approach follows the works in engineering field, and updates the strain with velocity gradient. Here we also follow this idea, and adapt the solid part to our MPM framework.

2.2. MPM simulation

MPM [16] has been used to simulate a wide range of solid materials in the past two decades, but is only applied in computer graphics in recent years. Stomakin et al. [14] first introduced MPM into graphics for simulating snow, using an elastoplastic constitutive model. Their framework is based on the finite deformation theory, and an energy related with deformation gradient tensor is used to derive the shear force. Following this work, they introduced a novel MPM method for heat transport, melting and solidifying materials[15].

Jiang et al. [7] tuned the model in [14] to simulate granular materials. Later Klár et al. [8] introduced the Drucker-Prager plastic flow model into MPM to simulate sand dynamics, and Daviet et al. [5] presented a semi-implicit scheme for granular materials.

Our MPM method is similar to the works mentioned above, but here we follow the small deformation theory, and adopt the constitutive model which has been used in the engineering field, especially in SPH based works.

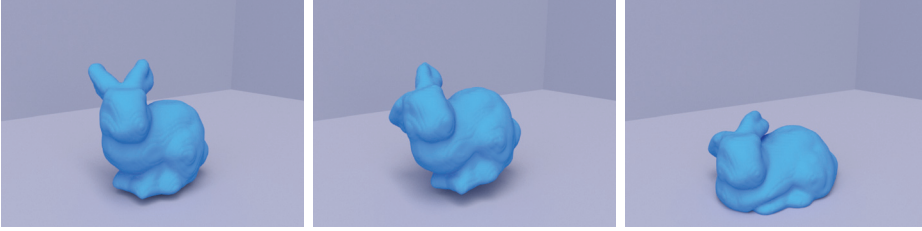


Figure 1: The elastic and the elastoplastic bunny.

We show that MPM based method is equally flexible and more stable than SPH for these applications.

The grid/particle transfer in MPM typically follows the method used in PIC or FLIP, or a weighted combination of both. While the transfer in PIC suffers from the loss in angular momentum, the transfer in FLIP brings more noise into the system and makes it unstable. Jiang et al. [7] resolved this problem by coming up with the APIC method, which greatly increases the stability while still keeps a very low dissipation rate. This has been widely applied in the following MPM based works([8],[5], etc), and we also follow this method in our work.

3. Solid mechanics

3.1. Deformable objects

The motion of solid materials obeys conservation of mass and conservation of momentum. Since the conservation of mass is naturally preserved by the particle representation, we here only focus on conservation of momentum

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

where \mathbf{v} is the velocity, ρ is the density, \mathbf{g} is the gravity, and $\boldsymbol{\sigma}$ is the Cauchy stress tensor determined by the constitutive model of solid materials.

3.2. Linear elastic model

In this section we briefly introduce the elastic constitutive model used by all the solid materials in our framework, and leave the plasticity to Section 3.1.

The stress tensor $\boldsymbol{\sigma}$ can be divided into two parts

$$(2) \quad \boldsymbol{\sigma} = -P\mathbf{I} + \mathbf{s}$$

where $-P\mathbf{I}$ is called the volumetric stress tensor, and \mathbf{s} is the deviatoric stress tensor. P is in fact the hydrostatic pressure. These two parts have the following relations

$$(3) \quad P = -\frac{1}{3}Tr(\boldsymbol{\sigma})$$

$$(4) \quad \mathbf{s} = \boldsymbol{\sigma} + P\mathbf{I}$$

where $Tr(\cdot)$ is the trace operator. The volumetric stress tensor describes the volume change of the deformed body, while the deviatoric part describes pure shear deformation.

To calculate the stress tensor $\boldsymbol{\sigma}$, we need a pair of tensors related with the velocity gradient

$$(5) \quad \begin{aligned} \dot{\boldsymbol{\epsilon}} &= \frac{1}{2}(\nabla\mathbf{v} + \nabla\mathbf{v}^T) \\ \boldsymbol{\omega} &= \frac{1}{2}(\nabla\mathbf{v} - \nabla\mathbf{v}^T) \end{aligned}$$

where $\dot{\boldsymbol{\epsilon}}$ is the strain rate tensor and $\boldsymbol{\omega}$ is the Jaumann rotation tensor. The first describes the rate of change of the strain tensor, while the second is used to correct the stress tensor when rotation happens.

According to the Hooke's law, \mathbf{s} is updated with the strain rate tensor

$$(6) \quad \dot{\mathbf{s}} = 2G \left(\dot{\boldsymbol{\epsilon}} - \frac{1}{3}Tr(\dot{\boldsymbol{\epsilon}})\mathbf{I} \right)$$

where G is the shear modulus.

The pressure P can be computed either with the equation of state (EOS), or with the volumetric change according to the strain rate tensor. Here we choose the later approach,

$$(7) \quad \dot{P} = -KTr(\dot{\boldsymbol{\epsilon}})$$

where K is the bulk modulus.

Considering the effect of the rotation, the final equation for updating the stress tensor $\boldsymbol{\sigma}$ is

$$(8) \quad \dot{\boldsymbol{\sigma}} = 2G \left(\dot{\boldsymbol{\epsilon}} - \frac{1}{3} \text{Tr}(\dot{\boldsymbol{\epsilon}}) \mathbf{I} \right) + K \text{Tr}(\dot{\boldsymbol{\epsilon}}) \mathbf{I} + \boldsymbol{\omega} \boldsymbol{\sigma} - \boldsymbol{\sigma} \boldsymbol{\omega}.$$

It is worth mentioning that this approach follows the small deformation theory, and will lose some accuracy when handling large deformations. However this inaccuracy is insignificant here, since we are updating with a reasonably small timestep [3].

When the deformation of deformable objects goes beyond a threshold, the objects are unable to recover their initial shape, and the irreversible part of the deformation is called plastic deformation. The criterion to decide when and how the plastic deformation will take place is called yield criterion.

For deformable objects, the Von Mises criterion is commonly applied, given as

$$(9) \quad f(J_2) = J_2 - Y^2$$

where Y is a parameter determining the yield stress, and J_2 is the second principal invariant of deviatoric stress tensor

$$(10) \quad J_2 = \frac{1}{2} \mathbf{s}_{\alpha\beta} \mathbf{s}_{\alpha\beta}.$$

Similar to [18], we assume the solid material has an elastic response at first, and calculate a trial stress tensor $\boldsymbol{\sigma}$ according to Equation (8). Then we can get the trial deviatoric stress tensor

$$(11) \quad \mathbf{s} = \boldsymbol{\sigma} - \frac{1}{3} \text{Tr}(\boldsymbol{\sigma}) \mathbf{I}.$$

If $f(J_2) > 0$, yield happens, and we have to project \mathbf{s} to the yield surface

$$(12) \quad \hat{\mathbf{s}} = \mathbf{s} \sqrt{\frac{Y}{J_2}}$$

The volumetric stress tensor remains unchanged, and the final stress tensor $\hat{\boldsymbol{\sigma}}$ is

$$(13) \quad \hat{\boldsymbol{\sigma}} = \hat{\mathbf{s}} + \frac{1}{3} \text{Tr}(\boldsymbol{\sigma}) \mathbf{I}.$$

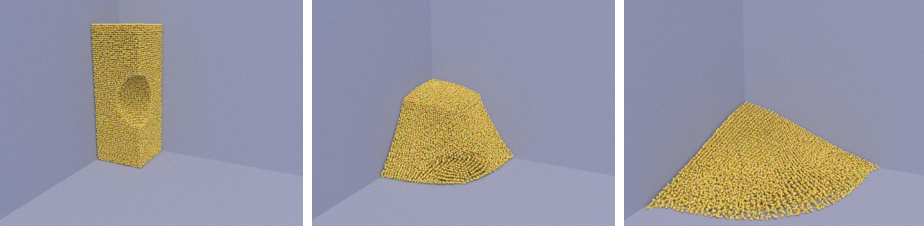


Figure 2: Sliding and sloping failure of granular materials.

3.3. Granular materials

When dealing with deformable objects, we can directly project the shear stress back to the yield surface. To deal with granular materials, however, we need to compute the increase of plastic strain explicitly.

The strain rate tensor $\dot{\epsilon}$ can be divided into two parts, namely the elastic and the plastic strain rate tensor

$$(14) \quad \dot{\epsilon} = \dot{\epsilon}_e + \dot{\epsilon}_p.$$

Only the elastic part $\dot{\epsilon}_e$ is responsible for the stress, and Equation (8) becomes

$$(15) \quad \dot{\sigma} = 2G \left(\dot{\epsilon}_e - \frac{1}{3} \text{Tr}(\dot{\epsilon}_e) \mathbf{I} \right) + K \text{Tr}(\dot{\epsilon}_e) \mathbf{I} + \omega \sigma - \sigma \omega.$$

As in [1], we use the Drucker-Prager criterion to describe the plastic deformation of granular materials, which has the following expression

$$(16) \quad f(I_1, J_2) = \sqrt{J_2} + \alpha_\phi I_1 - k_c$$

where I_1 and J_2 are, respectively, the first and second invariants of the stress tensor

$$(17) \quad \begin{aligned} I_1 &= \sigma_{xx} + \sigma_{yy} + \sigma_{zz} \\ J_2 &= \frac{1}{2} \mathbf{s}_{\alpha\beta} \mathbf{s}_{\alpha\beta}. \end{aligned}$$

α_ϕ and k_c are Drucker-Prager's constants, which are related to the Coulomb's material constants c (cohesion) and ϕ (internal friction). In 3D

cases, they are given as

$$(18) \quad \begin{aligned} \alpha_\phi &= \frac{\tan\phi}{\sqrt{9 + 12\tan^2\phi}} \\ k_c &= \frac{3c}{\sqrt{9 + 12\tan^2\phi}} \end{aligned}$$

The plastic strain rate tensor is determined by the plastic flow rule

$$(19) \quad \dot{\epsilon}_p = \dot{\lambda} \frac{\partial g}{\partial \boldsymbol{\sigma}}$$

where $\dot{\lambda}$ is the rate of change of plastic multiplier λ , and g is the plastic potential function.

Following the non-associated flow rule in [1], g is given as

$$(20) \quad g(I_1, J_2) = \sqrt{J_2} + \alpha_\psi I_1 - C$$

where α_ψ has the same expression as α_ϕ , but related with the dilatancy angle ψ

$$(21) \quad \alpha_\psi = \frac{\tan\psi}{\sqrt{9 + 12\tan^2\psi}}.$$

The dilatancy angle is assumed to be zero in this paper. A zero dilatancy angle indicates that the material is plastically incompressible.

The value of λ is determined by the consistency condition, which states that

$$(22) \quad df = \frac{\partial f}{\partial \boldsymbol{\sigma}_{\alpha\beta}} d\boldsymbol{\sigma}_{\alpha\beta} = 0.$$

This assures that the new stress state after loading is still on the yield surface.

Combining Equations (14), (15), (19) and (22), the stress-strain relationship is given by

$$(23) \quad \begin{aligned} \dot{\boldsymbol{\sigma}} &= 2G \left(\dot{\epsilon} - \frac{1}{3} Tr(\dot{\epsilon}) \mathbf{I} \right) + K Tr(\dot{\epsilon}) + \boldsymbol{\omega} \boldsymbol{\sigma} - \boldsymbol{\sigma} \boldsymbol{\omega} \\ &\quad - \dot{\lambda} \left(3\alpha_\psi K \mathbf{I} + \frac{G}{\sqrt{J_2}} \mathbf{s} \right) \end{aligned}$$

where $\dot{\lambda}$ has the expression

$$(24) \quad \dot{\lambda} = \begin{cases} \frac{3\alpha_\phi K Tr(\dot{\epsilon}) + (G/\sqrt{J_2}) \mathbf{s}_{\alpha\beta} \dot{\epsilon}_{\alpha\beta}}{9\alpha_\phi \alpha_\psi K + G} & f(I_1, J_2) > 0 \\ 0 & f(I_1, J_2) \leq 0 \end{cases}$$

The detailed deduction can be found in [3].

3.4. Stress projection

For granular materials, due to numerical errors in computation, the stress state may lie outside the yield surface, and the behavior of the materials is no longer consistent with the model. In such a circumstance, we have to use a return mapping algorithm to numerically return the stress state to the yield surface. This involves the following two procedures.

3.4.1. Tension cracking treatment. If the stress state of soil moves beyond the apex of the yield surface, e.g. to the stress state at point F in Figure 3, which satisfies the following condition

$$(25) \quad -\alpha_\phi \mathbf{I}_1 + k_c < 0$$

We shall adjust the hydrostatic stress component to the apex of the yield surface, according to the following equations

$$(26) \quad \hat{\boldsymbol{\sigma}} = \boldsymbol{\sigma} - \frac{1}{3} \left(\mathbf{I}_1 - \frac{k_c}{\alpha_\phi} \right) \mathbf{I}$$

3.4.2. Stress-scaling back procedure. When an elastic-perfectly plastic material experiences plastic deformation, the stress state must always lie on the yield surface during plastic loading. However, numerical errors may result in a stress state that lies far away from the yield surface. Therefore, a stress-rescaling procedure should be employed to return the stress state to the yield surface. The procedure is shown in Figure 3. The scaling factor is

$$(27) \quad r = \frac{-\alpha_\phi \mathbf{I}_1 + k_c}{\sqrt{J_2}}$$

When the stress state exceeds the yield surface, the Drucker-Prager yield criterion gives

$$(28) \quad \sqrt{J_2} + \alpha_\phi \mathbf{I}_1 - k_c > 0$$

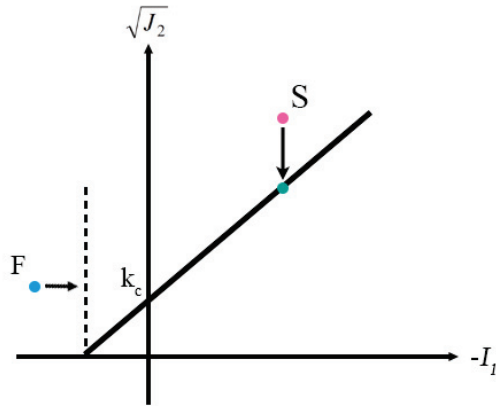


Figure 3: Stress state projection.

Then the stress is modified as

$$(29) \quad \hat{\boldsymbol{\sigma}} = r\boldsymbol{s} + \frac{1}{3}\boldsymbol{I}_1$$

where \boldsymbol{s} is the deviatoric stress tensor. This procedure is similar with the way we apply the Von Mises plastic criterion to deformable objects.

Actually when the dilatency angle ψ is zero, Equation (23) becomes

$$(30) \quad \dot{\boldsymbol{\sigma}} = 2G \left(\dot{\boldsymbol{\epsilon}} - \frac{1}{3}Tr(\dot{\boldsymbol{\epsilon}})\boldsymbol{I} \right) + KTr(\dot{\boldsymbol{\epsilon}}) + \boldsymbol{\omega}\boldsymbol{\sigma} - \boldsymbol{\sigma}\boldsymbol{\omega} - \dot{\lambda} \frac{G}{\sqrt{J_2}}\boldsymbol{s}$$

In this specific case, the plastic strain is in the same direction as the scaling back procedure, and the Drucker-Prager criterion can be applied solely with a projection.

3.5. Hardening model

Here we adopt the hardening model of [8], where plastic deformation increases the friction between granular particles. Here we define a hardening

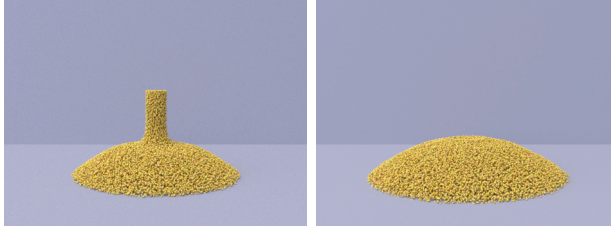


Figure 4: Pouring sands.

state q_p , and the hardening has the following expressions

$$(31) \quad q_p^{n+1} = q_p^n + \delta q_p$$

$$(32) \quad \phi = h_0 + (h_1 q_p^{n+1} - h_3) e^{-h_2 q_p^{n+1}}$$

$$(33) \quad \alpha_\phi = \frac{\tan \phi}{\sqrt{9 + 12 \tan^2 \phi}}$$

(32) models a curve with a maximum and asymptote. Plausible values of ϕ lie in $[0, 90)$, and feasible hardening parameters satisfy $h_0 > h_3 > 0$ and $h_1, h_2 \geq 0$.

The hardening change δq_p depends on the accumulated plastic strain ϵ_p

$$(34) \quad \delta q_p = \|\dot{\epsilon}_p\|_F \Delta t$$

where $\dot{\epsilon}_p$ can be determined by (19), and in this work has the expression

$$(35) \quad \dot{\epsilon}_p = \dot{\lambda} \left(\frac{1}{2\sqrt{J_2}} \mathbf{s} + \alpha_\psi \mathbf{I} \right).$$

We also need to take the stress-scaling back procedure into account, which gives

$$(36) \quad \delta q'_p = \|\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}\|_F$$

4. Material point method

In MPM approaches, particles (material points) are used to track the physical quantities of the materials. Specifically, particle p holds mass m_p , position \mathbf{x}_p , velocity \mathbf{v}_p , and affine momentum \mathbf{B}_p . The extra matrix \mathbf{B}_p stored on each particle is used for APIC transfers [7]. Up to a constant scale, this

quantity approximates the spatial derivative of grid velocity at the end of the previous time step ([8]). We also store the stress tensor $\boldsymbol{\sigma}_p$ and the hardening state variable q_p on each particle.

The grid works as a scratch sheet, and is reset to the initial state without any deformation at the beginning of each timestep. We first transfer the physical quantities such as mass and momentum from particles to the grid. The stress on each particle is then evaluated with the grid nodes' information. The forces is actually calculated on the grid nodes, and the deformation also happens on the grid. And the new positions and velocities are transferred back to the particles at the end of each timestep.

Following are the details of our approach.

4.1. Kernel

To transfer the quantities between particles and the grid, we use a weight function $\omega_{ip} = N(\mathbf{x}_p - \mathbf{x}_i)$ to determine how much a particle and a grid node would contribute to each other during interpolation, where \mathbf{x}_p is the particle's position and \mathbf{x}_i is the node's position.

In FLIP a multilinear kernel is typically used as the weight function, however here we follow [14] and use a cubic B-spline kernel. This is because the derivative of multilinear kernel $\nabla\omega_{ip}$ is discontinuous, and would produce discontinuous forces. Besides, it may be far from zero when $\omega_{ip} \approx 0$, leading to unexpected large forces. Compared with multilinear kernel, the cubic spline kernel gives more smooth results.

The kernel we use here is

$$(37) \quad \hat{N}(x) = \begin{cases} \frac{1}{2}|x|^3 - x^2 + \frac{2}{3}, & 0 \leq |x| < 1 \\ -\frac{1}{6}|x|^3 + x^2 - 2|x| + \frac{4}{3}, & 1 \leq |x| < 2 \\ 0, & 2 \leq |x| \end{cases}$$

$$(38) \quad N(\mathbf{u}) = \hat{N}\left(\frac{\mathbf{u}_x}{h}\right) \hat{N}\left(\frac{\mathbf{u}_y}{h}\right) \hat{N}\left(\frac{\mathbf{u}_z}{h}\right)$$

where h is the grid spacing.

4.2. Particle/grid transfer

In each step, we transfer the mass from particles to the grid nodes

$$(39) \quad m_i^n = \sum_p \omega_{ip}^n m_p$$

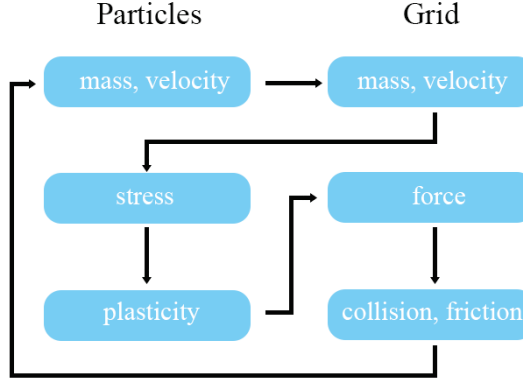


Figure 5: An overview of our MPM approach.

where n indicates the n th timestep.

We use APIC method in [7] to transfer the velocity state from particles to the grid. The velocity state of a particle is represented by \mathbf{v}_p and \mathbf{B}_p , where \mathbf{B}_p is an affine momentum. The main concept of APIC is to idealize the velocity field as locally affine on each particle, and the local velocity represented by a particle at the grid node \mathbf{x}_i can be written as $\mathbf{v}_p + \mathbf{C}_p(\mathbf{x}_i - \mathbf{x}_p)$. With the following transfer scheme, the affine velocity field is preserved, and both linear and angular momentum are automatically preserved as well.

The transfer from particles to grid is

$$(40) \quad m_i^n \mathbf{v}_i^n = \sum_p \omega_{ip}^n m_p (\mathbf{v}_p^n + \mathbf{B}_p^n (\mathbf{D}_p^n)^{-1} (\mathbf{x}_i - \mathbf{x}_p^n)),$$

where $\mathbf{C}_p^n = \mathbf{B}_p^n (\mathbf{D}_p^n)^{-1}$ and \mathbf{D}_p^n is analogous to an inertia tensor, given by

$$(41) \quad \mathbf{D}_p^n = \sum_i \omega_{ip}^n (\mathbf{x}_i - \mathbf{x}_p^n) (\mathbf{x}_i - \mathbf{x}_p^n)^T.$$

The corresponding transfer from the grid back to particles is

$$(42) \quad \mathbf{v}_p^{n+1} = \sum_i \omega_{ip}^n \mathbf{v}_i^{n+1}$$

$$(43) \quad \mathbf{B}_p^{n+1} = \sum_i \omega_{ip}^n \mathbf{v}_i^{n+1} (\mathbf{x}_i - \mathbf{x}_p^n)^T.$$

For the cubic B-spline kernel in this work, \mathbf{D}_p^n has a surprisingly simple form, $\mathbf{D}_p^n = \frac{1}{3}h^2\mathbf{I}$, which is actually a constant scale factor.

4.3. Full method

Here we outline the full update procedure.

- 1) **Rasterize particle data to the grid.** Transfer the mass and momentum from particles to the grid, as in Section 4.2.
- 2) **Compute grid node densities.** The density of the grid node is computed as $\rho_i^n = m_i^n/h^3$.
- 3) **Compute particle velocity gradients.** Giving the grid cell that a particle p lies in, the velocity gradient at the particle is computed with the velocities of the cell's eight nodes: $\nabla\mathbf{v}_p^n = \sum_i \mathbf{v}_i^n (\nabla\omega_{ip}^n)^T$.
- 4) **Update particle stress σ_p^n** with the constitutive model in Section 3, such as (8) or (23), and apply the proper yield criterion.
- 5) **Update velocities on grid $\hat{\mathbf{v}}_i^{n+1}$** with $\frac{Dv}{Dt} = \frac{1}{\rho_i^n} \sum_p \sigma_p^n \nabla\omega_{ip}^n + g$.
- 6) **Apply grid-based body collisions** on $\hat{\mathbf{v}}_i^{n+1}$, and get the final velocity \mathbf{v}_i^{n+1} of the nodes .
- 7) **Update particle velocities and positions.** Transfer the velocity \mathbf{v}_p^{n+1} from grid to particles and update affine momentum \mathbf{B}_p^{n+1} as in Section 4.2. The position of the particle is updated as $\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t\mathbf{v}_p^{n+1}$.

4.4. Grid-based collision

In MPM approaches, the collisions between objects are naturally handled with the background Eulerian grids. When different objects are colliding, the nodes near the colliding interface would have the total momentum of both objects. The node velocities represent the velocity of the colliding interface. And the velocity gradient would correctly lead to the increase of the strains inside the colliding objects. Since the movement of particles are interpolated from the deformation of grid nodes, the penetration is automatically prevented.

When the objects are colliding with boundaries, we apply the following types of boundary condition.



Figure 6: Interaction between an elastic bunny and granular materials.

Stiky. If a node locates outside the boundary, stiky boundary requires $\hat{\mathbf{v}}_i^{n+1} = \mathbf{v}_b^{n+1}$, where \mathbf{v}_b^{n+1} is the velocity of the boundary object.

Separating. A separating boundary prevents a node from getting into the boundary, but fully permit the movement along and away from the collision surface. This can be written as the following expressions

$$(44) \quad \tilde{\mathbf{v}}_n = \min(\mathbf{v}_i^{n+1} \cdot \mathbf{n}, 0)\mathbf{n}$$

$$(45) \quad \mathbf{v}_i^{n+1} = \hat{\mathbf{v}}_i^{n+1} - \tilde{\mathbf{v}}_n$$

where $\tilde{\mathbf{v}}_n$ is the velocity along the normal of the collision surface, and \mathbf{n} is the normal unit vector pointing outward the surface.

Sliding. A sliding condition assures the node inside the boundary moving along the surface, but does not allow either moving inward or outward.

$$(46) \quad \tilde{\mathbf{v}}_n = (\hat{\mathbf{v}}_i^{n+1} \cdot \mathbf{n})\mathbf{n}$$

$$(47) \quad \mathbf{v}_i^{n+1} = \hat{\mathbf{v}}_i^{n+1} - \tilde{\mathbf{v}}_n$$

4.4.1. Friction. For stiky surfaces, the velocity of materials are set to be equal with the collision objects. While for the other two cases, sometimes we need additional frictions to stop materials from sliding on the surface.

Through the collision handling procedures above we can get the normal velocity change $\Delta v = \|\tilde{\mathbf{v}}_n\|$ and tangential velocity $\mathbf{v}_t = \hat{\mathbf{v}}_i^{n+1} - (\hat{\mathbf{v}}_i^{n+1} \cdot \mathbf{n})\mathbf{n}$. The amount of friction is controlled by a friction factor μ , and if $\mu\Delta v \geq \|\mathbf{v}_t\|$, the tangential movement is stopped by friction, and we need to set $\mathbf{v}_t = \mathbf{0}$. Otherwise, we apply the friction as

$$(48) \quad \mathbf{v}_i^{n+1} = \hat{\mathbf{v}}_i^{n+1} - \tilde{\mathbf{v}}_n - \frac{\mu\Delta v}{\|\mathbf{v}_t\|}\mathbf{v}_t$$

case	particle number	timestep(s)	ms/timestep
bunny	15.4k	5×10^{-4}	63
notch pile	50.2k	2×10^{-4}	102
coupling	99.6k	2×10^{-4}	200
pouring	107.7k	5×10^{-4}	210

Table 1: Simulation performance.

5. Results

We implemented the framework based on Nvidia CUDA and tested the program on a computer with an Nvidia GTX1080 graphic card.

We have simulated several examples to demonstrate the effectiveness of our method, including deformable objects of different plasticity, granular materials, and the interaction between them.

In Figure 1, a deformable bunny object is dropped to the ground(left). If it undergoes pure elastic deformation, it can bounce up and recover its initial shape (middle). But when plastic deformation happens (right), it can no longer maintain its initial shape and fails to bounce up.

To examine the model for granular materials, we set up a notched sand block, which is also one of the cases in [8]. In Figure 2, the block is initially placed at the corner, and begins to fall down as the simulation goes. In the end it forms a pile of sand, and the final shape is related to the inner friction angle. And in Figure 4 we show when the granular materials are being poured from above, they will form a cone.

In Figure 6 we show a bunny dropping onto a pile of granular materials, where the bunny has a density 5 times greater than that of the granular materials. The granular materials are gradually starting to slide when the bunny hits the top on it, and the grains are made to spread away. The materials are bounded within a relatively small box, which is not rendered in the pictures.

6. Discussion and conclusion

Comparison with SPH methods. MPM approaches are more stable than common SPH approaches, because of the use of background grids. In

MPM approaches, physical quantities such as mass and velocity are transferred from particles to the background grids, and after the calculation of each timestep, velocities are transferred back to particles. These transfers act like filtering processes, and can smooth out the noise due to the irregular spatial distribution of particles.

Therefore in MPM approaches, there is no need for the artificial viscosity, which is however necessary in SPH approaches to eliminate the tensile and numerical instability. The APIC transfers used here does not harm the stability of MPM approaches, but can still preserve the linear momentum and angular momentum. This leads to a low dissipation rate, and the details of movement are well preserved. Besides, detecting collisions is trivial on the grid, and penetrations are automatically prevented.

Since SPH and MPM approaches share the lagrangian representation of materials, the way of discretizing the constitutive models are quite similar. So the constitutive models formerly implemented in SPH method can be easily adapted to MPM.

However, not all SPH applications can be replaced with MPM. For example, when simulating multiphase materials with volume fraction information stored on the particles, the exchange of different phase between particles can only be achieved by particle-particle interactions, and an SPH scheme is thus necessary.

Limitations. The explicit time integration we used here limits the timestep. A timestep too large would lead to unrealistic phenomena and numerical errors. This can be solved by using an implicit integration, and the efficiency can be further improved.

Following small deformation theory, we directly update the Cauchy stress tensor with velocity gradient. While this is simple in the mathematical form, and is also a good approximation, in some cases it could be better to follow the finite deformation theory.

Also we would like to investigate more constitutive models, and the way to describe the behavior of the mixture of different kinds of materials. Besides, the visualization of different materials and their combination requires further study.

Conclusion. We have presented an MPM framework for simulating various solid materials including deformable objects and granular materials. The constitutive models applied in previous SPH based works can be easily adapted to our framework. The collision of different objects are handled in a stable way with the use of the grid.

Acknowledgements

The earlier version of this paper was presented in workshop of the European Unions Seventh Framework Programme at Bournemouth University. This work was supported by the National Key Technology R&D Program (Project Number 2017YFB1002701) and the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme FP7/2007-2013/ under REA grant agreement No. 612627.

References

- [1] Y. An, Q. Wu, C. Shi, and Q. Liu, *Three-dimensional smoothed-particle hydrodynamics simulation of deformation characteristics in slope failure*, *Geotechnique* **66** (2016), 670–680.
- [2] M. Becker, M. Ihmsen, and M. Teschner, *Corotated sph for deformable solids*, in: *Proceedings of the Fifth Eurographics Conference on Natural Phenomena, NPH'09*, pages 27–34, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association.
- [3] H. H. Bui, R. Fukagawa, K. Sako, and S. Ohno, *Lagrangian mesh-free particles method (sph) for large deformation and failure flows of geomaterial using elastic-plastic soil constitutive model*, *International Journal for Numerical and Analytical Methods in Geomechanics* **32** (2008), no. 12, 1537–1570.
- [4] P. W. Cleary and R. Das, *The potential for SPH modelling of solid deformation and fracture*, pages 287–296, Springer Netherlands, Dordrecht, 2008.
- [5] G. Daviet and F. Bertails-Descoubes, *A semi-implicit material point method for the continuum simulation of granular materials*, *ACM Trans. Graph.* **35** (2016), no. 4, 102:1–102:13.
- [6] J. Gray, J. Monaghan, and R. Swift, *Sph elastic dynamics*, *Computer Methods in Applied Mechanics and Engineering* **190** (2001), no. 49, 6641–6662.
- [7] C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin, *The affine particle-in-cell method*, *ACM Trans. Graph.* **34** (2015), no. 4, 51:1–51:10.

- [8] G. Klár, T. Gast, A. Pradhana, C. Fu, C. Schroeder, C. Jiang, and J. Teran, *Drucker-prager elastoplasticity for sand animation*, ACM Trans. Graph. **35** (2016), no. 4, 103:1–103:12.
- [9] L. D. Libersky and A. G. Petschek, *Smooth particle hydrodynamics with strength of materials*, in: H. E. Trease, M. F. Fritts, and W. P. Crowley, editors, *Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method*, Volume 395 of Lecture Notes in Physics, Berlin Springer Verlag, pages 248–257, 1991.
- [10] M. Müller, D. Charypar, and M. Gross, *Particle-based fluid simulation for interactive applications*, in: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '03*, pages 154–159, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [11] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa, *Point based animation of elastic, plastic and melting objects*, in: *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '04*, pages 141–151, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.
- [12] B. Ren, Y.-T. Jiang, C.-F. Li, and M. C. Lin, *A simple approach for bubble modelling from multiphase fluid simulation*, *Computational Visual media* **1** (2015), no. 2, 171–181.
- [13] B. Solenthaler, J. Schflfi, and R. Pajarola, *A unified particle model for fluid-solid interactions*, *Computer Animation and Virtual Worlds* **18** (2007), no. 1, 69–82.
- [14] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle, *A material point method for snow simulation*, ACM Trans. Graph. **32** (2013), no. 4, 102:1–102:10.
- [15] A. Stomakhin, C. Schroeder, C. Jiang, L. Chai, J. Teran, and A. Selle, *Augmented mpm for phase-change and varied materials*, ACM Trans. Graph. **33** (2014), no. 4, 138:1–138:11.
- [16] D. Sulsky, Z. Chen, and H. Schreyer, *A particle method for history-dependent materials*, *Computer Methods in Applied Mechanics and Engineering*, **118** (1994), no. 1, 179–196.

- [17] M. L. Wilkins, B. University of California, and L. L. Laboratory, *Calculation of elastic-plastic flow*, University of California Lawrence Radiation Laboratory, Livermore, CA, 1963. 1 v. (various pagings).
- [18] X. Yan, Y.-T. Jiang, C.-F. Li, R. R. Martin, and S.-M. Hu, *Multiphase sph simulation for interactive fluids and solids*, ACM Trans. Graph. **35** (2016), no. 4, 79:1–79:11.

DEPARTMENT OF COMPUTER SCIENCE
TSINGHUA UNIVERSITY, BEIJING, CHINA
E-mail address: `jhodinger@gmail.com`

DEPARTMENT OF COMPUTER SCIENCE
TSINGHUA UNIVERSITY, BEIJING, CHINA
E-mail address: `yangtao9009@gmail.com`

THE NATIONAL CENTRE FOR COMPUTER ANIMATION
BOURNEMOUTH UNIVERSITY, BOURNEMOUTH, UK
E-mail address: `JChang@bournemouth.ac.uk`

RECEIVED OCTOBER 12, 2017
ACCEPTED NOVEMBER 11, 2017