A Unified Particle System Framework for Multi-Phase, Multi-Material Visual Simulations

TAO YANG, Tsinghua University
JIAN CHANG, Bournemouth University
MING C. LIN, University of North Carolina, Chapel Hill
RALPH R. MARTIN, Cardiff University
JIAN J. ZHANG, Bournemouth University
SHI-MIN HU, Tsinghua University

Fig. 1. Frying an egg on a hot pan, achieved by enabling the diffusion of phases but disabling the diffusion of concentrations.

We introduce a unified particle framework which integrates the phase-field method with multi-material simulation to allow modeling of both liquids and solids, as well as phase transitions between them. A simple elasto-plastic model is used to capture the behavior of various kinds of solids, including deformable bodies, granular materials, and cohesive soils. States of matter or *phases*, particularly liquids and solids, are modeled using the non-conservative Allen-Cahn equation. In contrast, materials—made of different substances—are advected by the conservative Cahn-Hilliard equation. The distributions of phases and materials are represented by a *phase* variable and a *concentration* variable, respectively, allowing us to represent commonly observed fluid-solid i nteractions. T he d ynamics o f a m ulti-phase, multi-material system are governed by a unified Helmholtz free-energy density. This framework provides the first method in computer graphics capable of modeling a continuous interface between phases. It is versatile and can be readily used in many scenarios challenging to simulate. Examples are provided to demonstrate the capabilities and effectiveness of this approach.

CCS Concepts: • **Computing methodologies → Physical simulation**;

Additional Key Words and Phrases: Phase-Field, Multi-Material Simulation, Elastoplastic, Smoothed Particle Hydrodynamics, Phase Transition

0730-0301/2017/11-ART00 $15.00

1 INTRODUCTION

Physically-based simulation of fluids and solids has recently gained much attention in computer graphics, as it is central to reproducing realistic visual effects for a wide range of real-world phenomena. Researchers have developed physically-based modeling techniques using various grid- or particle-based frameworks. The smoothed-particle hydrodynamics (SPH) method [Monaghan 1992] is widely used as a particle-based method due to its mass-conservation property, and its flexibility in handling topological changes. The SPH methods have been widely used to simulate fluid flow [Müller et al. 2003], and it has been further adapted to simulate the dynamics of both deformable bodies [Gerszewski et al. 2009; Müller et al. 2004] and granular materials [Alduán and Otaduy 2011].

Since multiple fluids and solids often co-exist in the real world, multi-material methods have also been introduced. Previous work can be broadly divided into two categories: methods handling fluid-solid interaction, and those only handling multiple fluids. Many approaches for fluid-solid interaction have been proposed, covering fluid to rigid solid coupling [Akinci et al. 2012], fluids interacting with elastoplastic objects [Keiser et al. 2005; Solenthaler et al. 2007], fluids interacting with granular materials [Lenaerts and Dutré 2009], and fluids in porous materials [Lenaerts et al. 2008]. In other cases, the entire system may comprise different miscible or immiscible fluids. Solenthaler and Pajarola [2008] simulated fluids using density contrast. By adopting the concept of volume fraction [Müller et al. 2005], Ren et al. [2014] and Yang et al. [2015] managed to capture a wide range of multi-fluid flow phenomena with rich visual effects, using a mixture model and Helmholtz free energy. They focused on interactions between multiple fluids, and solids were not considered. Yan et al. [2016] extended the multi-fluid SPH framework to incorporate solids, achieving impressive results. However, their

method is based on drift velocity, which requires small time steps, and cannot easily or intuitively capture the evolution of phenomena based on energy considerations [Yang et al. 2015]. Furthermore, it cannot simulate phase-change phenomena such as melting and solidification.

Phase transitions are commonly observed in the physical world, such as dissolving, melting and solidification. Previous particle-based simulation methods model this process mainly using either a concentration criterion [Yan et al. 2016] or a temperature criterion [Stomakhin et al. 2014], so that a given particle changes its phase if its concentration or temperature exceeds a pre-defined threshold. Although this approach is straightforward and easy-to-implement, it suffers from two problems. First, the criteria are inconsistent: it is problematic to model phase change phenomena involving both concentration- and temperature-related variables. Secondly, the interfaces between phases are discontinuous, as a given particle can only be in a single phase. To address these problems, we represent the phase separately using an extra variable to describe it. This phase variable is governed by a unified energy density function related to both concentration and temperature.

In computer graphics, the terms 'fluid', 'phase' and 'material' have been widely used, but not always consistently or accurately. We formally define them here: fluids are substances with zero shear modulus; phases are states of substances, and materials are different kinds of substances (which may exist in any state or phase). For simplicity, we only consider the three most commonly observed phases of matter, i.e., solid, liquid and gas, and largely focus on the first two for demonstration of concepts in this paper.

Our approach significantly extends the energy-based multi-fluid approach proposed by Yang et al. [2015] to represent 'solid' phases, enabling modeling of a variety of fluid-solid interactions. We treat phases and materials independently, thereby the unified multi-material simulation framework can model a much wider range of real-world phenomena. It is straightforward to describe the state of a particle with a phase variable and a concentration variable. For simplicity, we assume that the phase variable acts on the whole particle which can be composed of multiple materials.

In summary, the main contributions of this work are as follows:

- A simple elastoplastic model is introduced to simulate various solids, including deformable objects, granular materials, and cohesive soils. It is capable of capturing a variety of solid phenomena that have been largely ignored in the literature, such as collapsing granular columns with varying aspect ratios, landslides, and dry/wet sand.
- By treating materials and phases separately, a much wider range of challenging real-world phenomena involving multiple materials and phases can be simulated, including complex fluid-solid interactions, such as dissolving, melting, etc.
- Our phase-field method defines the phase boundary using a phase variable, providing a *continuous* interface between phases.
- A unified Helmholtz free-energy density for both the non-conservative Allen-Cahn equation and the conservative Cahn-Hilliard equation is proposed to guide the evolution of concentrations and phases, respectively.

In the rest of the paper, Section 2 discusses related work. We present our elastoplastic model in Section 3. We introduce our unified multi-phase, multi-material simulation framework in Section 4. Some applications of this new unified framework are shown in Section 5, with illustrative examples in Section 6. Finally, limitations and future work are discussed in Section 7.

2 RELATED WORK

In computer graphics, multi-material fluid simulation has been extensively investigated during the last decade. Again, here we classify multi-material fluid simulations as those involving only liquids, and those involving solids and liquids. We start by introducing particle-based solid simulation, then providing an overview of the relevant works closest to this paper.

2.1 Solid Simulation

In computer graphics, particle-based methods are commonly used to simulate fluids [Monaghan 1992; Müller et al. 2003]; they also work well for simulating solid materials, including rigid objects, deformable bodies, and granular materials. Gray et al. [2001] simulated elastic objects using a linear model by defining the artificial stress based on the signs of the principal stresses. Although this gave plausible results, their model suffers from accumulated numerical error. To capture the dynamics of truly elastic materials or large elastic deformations, researchers later developed a variety of solutions, such as storing the initial state [Müller et al. 2004], calculating the deformation gradient tensor [Gerszewski et al. 2009], adopting an implicit framework [Zhou et al. 2013], or use of 'embedded space' [Jones et al. 2014]. Müller et al. [2004] animated elastic, plastic, and melting objects using a physical model derived from continuum mechanics. This method needs to store the initial state in order to calculate stress. Gerszewski et al. [2009] proposed a new way to compute the deformation gradient, avoiding the need to store the initial rest configuration. Zhou et al. [2013] extended this work, using an implicit numerical integrator to achieve more stable simulation of elastoplastic materials. Jones et al. [2014] introduced *embedded space*—the least-squares best fit of the material's rest state in 3D—and used it to handle extreme elastic and plastic deformations. Jiang et al. [2015a] animated deformable objects, as well as collision handling, using an *affine particle-in-cell* method (APIC) with a Lagrangian force model.

Zhu and Bridson [2005] adopted an existing fluid solver for simulating granular materials, decomposing the spatial domain according to the strain rate tensor. Bell et al. [2005] used non-spherical particles to represent discrete elements of simulated materials, such as sand and rigid bodies. Narain et al. [2010] solved both the internal pressure and frictional stresses in granular materials, achieving two-way coupling between granular materials and solid bodies. Using a friction and cohesion model, Alduán and Otaduy [2011] simulated granular flows in a *predictive-corrective incompressible SPH* (PCISPH) framework; this work was extended by Ihmsen et al. [2012], using a two-scale framework to simulate granular materials with high-quality visual detail. Recently, the *material point method* (MPM) has gained popularity as a hybrid method to capture a variety of fluid and solid phenomena. Klár et al. [2016] recreated a wide range of visual sand phenomena by combining Drucker-Prager plastic

flow and Hencky-strain-based hyperelasticity. Daviet et al. [2016] presented a continuum-based method for simulating non-smooth viscoplastic granular rheology using a material point method.

2.2 Multi-Fluid Simulation

A multi-fluid system is composed of more than one miscible or immiscible fluids. Particle-based multi-fluid simulations have attracted much attention in the last decade. To simulate immiscible fluids, Solenthaler and Pajarola [2008] derived a modified density calculation method to deal with density contrast more precisely. Peer et al. [2015] incorporated this approach into an *implicit incompressible SPH* (IISPH) to model multiphase highly viscous fluids. Macklin et al. [2014] simulated immiscible fluids with a density ratio using position based dynamics, capturing the Rayleigh-Taylor instability. For miscible fluids, the volume fraction [Müller et al. 2005] is widely used to represent the spatial distribution of different components. Liu et al. [2011] integrated the volume fraction with an SPH solver and mixed components using concentration differences. To capture the underlying interactions between components due to flow motions and force distributions, Ren et al. [2014] took the drift velocity into consideration. By doing so, they could simulate a wide range of multi-fluid phenomena, including mixing/unmixing of miscible and immiscible fluids, and chemical reactions. Park et al. [2008] showed that energy can be used to drive multi-fluid simulations, and adopted the Cahn-Hilliard equation to handle both miscible and immiscible fluids using a lattice Boltzmann method (LBM). Their approach was later extended by Yang et al. [2015], who achieved impressive results for problems such as extraction (i.e. separating a substance from a mixture) and mixing of an egg yolk and egg white. All these multi-fluid simulations only consider fluids, but do not model fluid-solid interactions.

2.3 Fluid-Solid Interactions

Fluid-solid interactions have attracted much attention in computer animation. Commonly observed fluid-solid interactions fall into four categories: fluid-solid coupling, solid wetting, phase transitions between solid and fluid, and dissolving of solids.

Solenthaler et al. [2007] used a unified particle model to handle the interactions between fluids and various solids including deformable bodies. Ihmsen et al. [2010] focused on rigid-fluid couplings and showed that adaptive timesteps are required for boundary handling in a PCISPH framework. Akinci et al. [2012] proposed a versatile SPH-based approach for two-way fluid-solid coupling using per-particle volume correction. More recently, Shao et al. [2015] combined PCISPH and geometric lattice shape matching to achieve two-way fluid-solid coupling with large time steps.

For solid wetting, previous methods mainly focus on porous materials. Lenaerts et al. [2008] succeeded in capturing a fluid flowing through a porous deformable material; the porous flow is governed by Darcy's Law. This work was later extended to simulate interactions between sand and fluids by using a *unified SPH framework* [Lenaerts and Dutré 2009]. Recently, Tampubolon et al. [2017] presented a multi-species model to capture the interactions between porous sand and water, using a continuum mixture theory where species individually obey conservative equations and are coupled through a momentum exchange term.

Phase transitions are highly correlated with temperature or internal energy. SPH has been widely used to simulate melting objects [Becker et al. 2009; Chang et al. 2009; Keiser et al. 2005; Müller et al. 2004]. Keiser et al. [2005] animated solids and fluids, as well as phase transitions, by combining the equations of solid mechanics with the Navier-Stokes equations. Stomakin et al. [2014] introduced a dilational /deviatoric splitting of the constitutive model for heat transport, melting and solidifying materials. However, dissolving of solids, which also involves a phase change, is largely overlooked by such methods.

Dissolving of soluble materials is commonly observed in daily life. Jiang et al. [2015b] proposed an energy-based method for real-time simulation of such phenomena. Yan et al. [2016] extended the work of Ren et al. [2014] to model solid phases. The distribution and shapes of both fluids and solids are uniformly represented by their volume fraction function and are governed by the conservation of mass and momentum within different phases. The approach impressively captures various fluid-solid interactions, but it cannot capture the evolution of phenomena based on energy, or provide flexible fluid control. Furthermore, it cannot model phase-change phenomena such as melting and solidification.

In such work, solid particles are insoluble in fluid-solid coupling scenarios. It is also possible to model solid wetting by considering wettable solid particles to be slightly soluble while they remain in a solid state, rather than by considering them to be porous materials [Lenaerts et al. 2008]. To simulate pure phase transitions, particles only change their states of matter. When solids dissolve, the solid particles change phase in the liquid.

These four categories cover a wide range of real-world phenomena, and almost all phenomena modeled by previous research. In contrast to the previous work, we adopt a concentration variable *and* a phase variable to describe the separate evolution of materials and phases, allowing us to handle all four categories of interaction (see Table 1 later) *together* in a versatile way. While many previous works have considered multi-material simulations for solids and liquids, none can handle all four types of interaction simultaneously.

3 SOLID MECHANICS

As a basis for our approach, we first introduce an elastoplastic model for particle-based solid simulation. To solve the problem of inconsistent pressure forces when handling interactions, we treat the hydrostatic pressure of solids in a similar way to that of fluids. To allow realistic simulation of granular materials, we adopt a *tension cracking treatment* and a *stress scaling-back procedure* to handle the numerical errors in computational plasticity. This elastoplastic model underpins the solid simulations in our approach, allowing modeling of deformable bodies, wet and dry granular materials, cohesive soils, and various interactions.

3.1 Constitutive Model

The momentum equation for solids can be expressed as:

$$\frac{Du}{Dt} = \frac{1}{\rho}\nabla \cdot \sigma + f_{\text{ext}}, \tag{1}$$

where $D/Dt = \partial/\partial t + u \cdot \nabla$ is the substantial derivative, u is the velocity, ρ is the density, f_{ext} is the external force, and σ is the

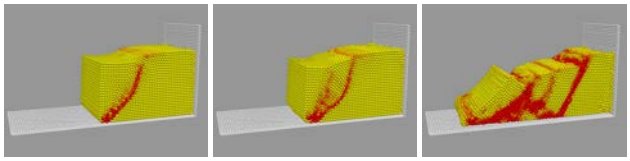

Fig. 2. Particle views of a landslide coloured according to total deviatoric plastic strain, for different Young's modulus and Poisson ratio. These lead to different failure patterns.

Fig. 3. A particle view demonstrating the benefit of using Jaumann stress rate. Left: with it, particles belonging to an elastic cube stay attached during rotation. Right: without it, particles at the corner detach as the cube rotates.

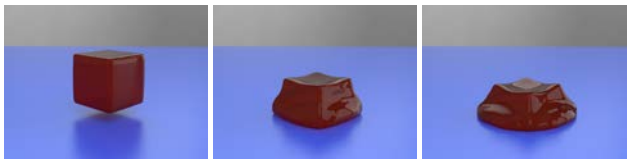

Fig. 4. Deformable jelly. Left: a purely elastic jelly. Middle and Right: plastic jellies, with different degrees of plasticity.

Cauchy stress tensor of the solid, which is normally divided into two parts in SPH: hydrostatic pressure p and deviatoric shear stress s, i.e., $\sigma = -p\delta + s$. We rewrite Equation (1) as:

$$\frac{Du}{Dt} = \frac{1}{\rho}(-\nabla p + \nabla \cdot s) + f_{\text{ext}}. \tag{2}$$

For consistency, the hydrostatic pressure is determined for both fluids and solids using Tait's equation [Becker and Teschner 2007]. To determine s, the Drucker-Prager model is adopted in our work. The yield condition f and plastic potential function ∂ are respectively given by:

$$f = \frac{1}{\sqrt{2}}S - 3\alpha p - \gamma, \tag{3}$$

$$\partial = \frac{1}{\sqrt{2}}S - 3\beta p, \tag{4}$$

where $S = \|s\|$, α and γ are Drucker-Prager's constants, which are related to ϕ (internal friction) and the Coulomb constant k_c (cohesion); the later can be treated as a function of the accumulated plastic strain to handle plastic hardening and softening. For simplicity, we just set k_c to a constant. β has a similar expression to α and is related to the dilation angle ψ of materials such as soils. The yield condition determines where plastic deformation occurs; the plastic potential function is used to obtain the degree of plasticity (see the Appendix A). For simplicity, we formulate the deviatoric shear rate $\dot{s}$ as in the works of Bui et al. [2008] and Chen et al. [2012]:

$$\dot{s} = 2G\left(\dot{e} - \dot{\lambda}\frac{s}{2S}\right), \tag{5}$$

(see the Appendix for the derivation A) where '˙' denotes the derivative with respect to time, G is the shear modulus, $\dot{\epsilon} = (\nabla u + \nabla u^T)/2$ denotes the total strain rate tensor, and $\dot{e} = \dot{\epsilon} - \text{Tr}(\dot{\epsilon})/3\delta$ is the deviatoric shear strain rate tensor. The scalar function $\dot{\lambda}$ denotes the rate of change of the plastic multiplier λ, and is given by:

$$\dot{\lambda} = \begin{cases} \frac{1}{G + 9K\alpha}\left(\frac{\sqrt{2}G}{S}s : \dot{e} + 3K\alpha\,\text{Tr}(\dot{\epsilon})\right), & f > 0 \\ 0, & f \le 0 \end{cases}, \tag{6}$$

where K is the bulk modulus. The shear and bulk moduli are related to the Young's modulus E and Poisson ratio v by:

$$K = \frac{E}{3(1-2v)}, \quad G = \frac{E}{2(1+v)}. \tag{7}$$

In our implementations, the deviatoric shear stress s in Equation (2) is obtained by integrating its time derivative $\dot{s}$ in Equation (5). We use the leapfrog algorithm for numerical integration; s and S on the right-hand side of Equations (5) and (6) are obtained at the half-time step.

In this work, we use a fixed coordinate system and integrate the stress rate. As a result, the rotation of a material introduces changes in Cauchy stress. To eliminate this consequence, an extra stress rate, invariant with respect to material rotation, is employed to describe the material response. The Jaumann stress rate is implemented, and works well for small strains and large material rotations (see Figure 3).

$$\dot{s}_J = \dot{s} + s \cdot \dot{\omega} - \dot{\omega} \cdot s, \tag{8}$$

where $\dot{\omega} = (\nabla u - \nabla u^T)/2$ denotes the spin rate tensor; subscript J denotes the Jaumann rate.

Jaumann stress rate is a so-called *objective stress rate*, aiming to eliminate the mechanical response of a material with respect to the frame of reference. It has been previously used in the graphics community occasionally, e.g. by Yan et al. [2016]. They simply employ such a term, but did not illustrate how it benefits simulation; our examples do so. As illustrated in Figure 3 (and the supplementary video), the rotation of a deformable cube introduces extra changes of the Cauchy stress tensor due to the fixed frame of reference, which however should not be part of the mechanical response of a material, resulting in particles at the corner detaching. By adopting the Jaumann stress rate, this issue could be largely prevented.

To smoothly approximate the Mohr-Coulomb hexagon on the deviatoric stress plane (π plane), several strategies have been proposed to determine the Drucker-Prager cone parameters [Wang and Sitar 2004] (see the Appendix B). In 3D, the parameters are determined by [Chen et al. 2012]:

$$\alpha = \frac{2\sin\phi}{\sqrt{3}(3-\sin\phi)}, \quad \beta = \frac{2\sin\psi}{\sqrt{3}(3-\sin\psi)}, \gamma = \frac{6k_c\cos\phi}{\sqrt{3}(3-\sin\phi)}.$$

ψ is set to 0 in our examples. The implementation is given in Algorithm 1 later.

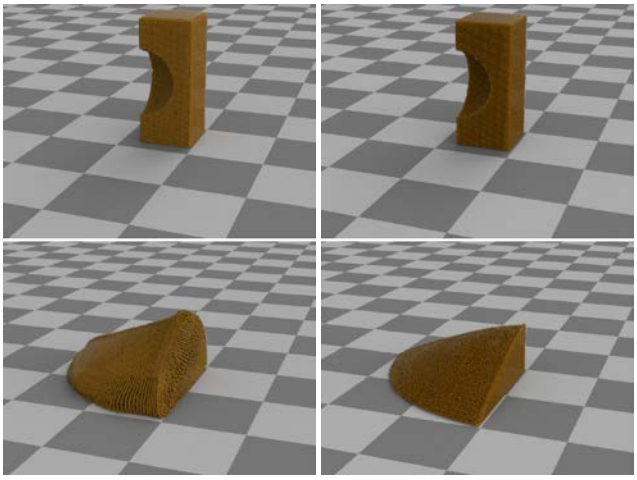

Fig. 5. Comparison with an MPM-based solution by revisiting [Zhu and Bridson 2005]'s column collapse. The friction angle is 20°; 8000 particles were used in both cases. Our SPH-based approach is about 4 times faster than the MPM-based one. Left: MPM solution achieves 22 fps. Right: SPH-solution achieves 80 fps. Without treating tension cracking or using stress scaling-back, the collapse of the sand pile exhibits cohesive behavior.

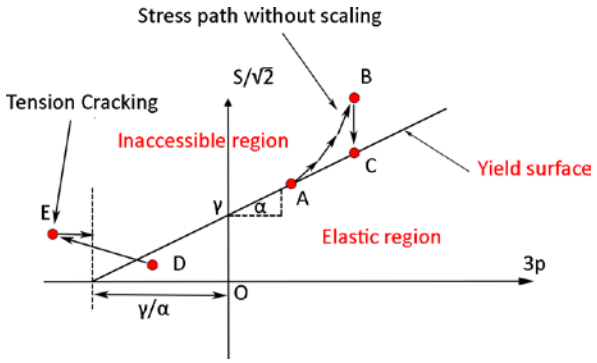

Fig. 6. 2D illustration of the Drucker-Prager yield criterion and numerical errors in computational plasticity.

3.2 Granular Materials

The proposed elastoplastic model can be used to capture the dynamics of purely elastic objects or plastic ones, depending on whether or not the Drucker-Prager yield criterion is satisfied (see Figure 4). This model is also capable of simulating granular materials [Bui et al. 2008; Chen et al. 2012]. However, the granular materials directly simulated by this model are cohesive (even if the cohesion k_c is set to 0), acting like wet sands (see Figure 5). We have implemented this model using SPH and MPM [Sulsky et al. 1995], using CUDA; our observation is that an SPH-based approach is around 4 times faster than the MPM-based one. See Figure 5.

We have introduced an elastic-perfectly plastic model. Theoretically, the stress state should not lie outside the yield surface (the inaccessible region in Figure 6) when plastic deformation occurs. However, due to numerical errors during computation, the stress state may leave the elastic region as shown in Figure 6. In this case, a return mapping algorithm is usually used to numerically return the stress state to the yield surface [Bui et al. 2008; Chen and Mizuno 1990]. We consider two kinds of errors.

Tension cracking treatment: if the stress state moves beyond the apex of the yield surface (point E in Figure 6), a numerical error known as tension cracking occurs [Chen and Mizuno 1990]. This error shares some similarities with SPH tensile instability, resulting in unrealistic fracturing or particle clumps. Following Chen and Mizuno [1990], if the stress state exceeds the apex of the yield surface, it should satisfy the condition below:

$$3\alpha p + \gamma < 0. \tag{9}$$

To avoid tension cracking, it is crucial to shift the hydrostatic pressure to the apex of the yield surface, so the treatment is as follows:

$$\tilde{\sigma}_{yy} = \sigma_{yy} + (p + \frac{\gamma}{3}), \tag{10}$$

where $y \in \{1, 2, 3\}$. If the cohesion coefficient k_c is set to 0, γ equals 0. In this case, the condition in Equation (9) is satisfied only if the hydrostatic pressure p is negative. The treatment in Equation (10) ensures the pressure is non-negative, which also solves the SPH tensile instability problem.

Stress-scaling back procedure: when plastic deformation occurs, numerical errors during computation may lead to a stress state far from the yield surface (the path AB in Figure 6). In such circumstances, a stress rescaling procedure may be used with the help of a scaling factor r [Bui et al. 2008]. For the Drucker-Prager yield criterion, this scaling factor is given by:

$$r = \frac{\sqrt{2}3\alpha p + \gamma}{S}. \tag{11}$$

Accordingly, using the Drucker-Prager yield criterion, when the stress state exceeds the yield surface, it corresponds to the following condition:

$$3\alpha p + \gamma < \frac{1}{\sqrt{2}} S. \tag{12}$$

The scaling factor r acts on the deviatoric shear stress components and the hydrostatic pressure is left unchanged, so:

$$\tilde{s} = rs. \tag{13}$$

We have discussed two kinds of errors as well as their return mapping algorithms (Equations (10) and (13)). Since the leapfrog algorithm is used in this work, the above return mapping algorithms are applied during the half-time step.

We have modelled granular materials using the proposed elastoplastic model in conjunction with the return mapping algorithms, achieving plausible visual results. Our model can simulate the collapses of dry granular columns with different friction angles; see Figures 7b–7d. It can further capture the varying patterns when granular columns with different aspect ratios collapse. When $H/R = 2.0$, the entire column starts to flow immediately. At first, the upper free surface of the column remains undeformed and horizontal, but later, the top deforms to form a dome: see Figure 7d. When $H/R = 0.9$, a circular area in the upper surface of the column is preserved undis- turbed: see Figure 7g. Figure 7 shows collapsing granular columns with varying friction angles and aspect ratios. We also demonstrate

Fig. 7. Sand collapses. (a) initial dry granular column with aspect ratio $H/R = 2.0$, for (b)–(e). (b)–(d) final states of collapse for friction angles φ of 20°, 30°, and 40°. (e) final states of collapse without tension cracking treatment for a friction angle φ of 40°. (f) initial dry granular column for (g) with aspect ratio $H/R = 0.9$. (g) final state of the collapse with friction angle of 40°.

the benefit of incorporating the tension cracking treatment: see Figure 7e. Without this term, the sand forms clumps and exhibits cohesive behavior.

4 PHASE-FIELD MULTI-MATERIAL SIMULATION

In this section, we first review the energy-driven multi-fluid simula-tion proposed by Yang et al. [2015], and then show how to extend this framework to handle other phases.

4.1 Phase-Field Method

The phase-field model was established mainly for solving interface problems. The method has been used to describe solidification processes, as well as microstructure evolution at the interfaces of solids and liquids. Yang et al. [2015] used the Cahn-Hilliard equation [Cahn and Hilliard 1958] to calculate the changes in the fraction of each material. The rate of change of each material is given by:

$$\frac{Dc}{Dt} = \nabla \cdot (L_{CH} \nabla \mu), \tag{14}$$

where c represents the concentration of the material (we use mass fraction like Yang et al.), L_{CH} denotes its degenerate mobility, and μ is its chemical potential, related to the Ginzburg-Landau free energy density. The latter is composed of a bulk part and an interface part, and can be expressed as:

$$\mu = \frac{\partial F}{\partial c} - \xi^2 \Delta c, \tag{15}$$

where F is the Helmholtz free energy density, and ξ is associated with diffuse interfaces; it introduces an internal length scale (interface thickness). Yang et al. [2015] integrated their model into the position-based fluids approach, achieving real-time performance.

As explained in Section 1, we introduce an extra phase variable η to handle multiple phases consistently and continuously. η is used to describe the current state of each particle in a continuous manner. For example, we may take the value one to mean solid and zero for liquid; across an interface, the value varies smoothly from one to zero. The introduced phase variable η is considered as the proportion of a state. Unlike the concentrations of materials, the phase-field variables are not conserved. Real-world phenomena evolve from high-energy states to low-energy states. We assume the system locally tends to minimize energy and to conserve concentrations of materials at the same time. Therefore, we represent the evolution of the phase variable using:

$$\frac{D\eta}{Dt} = L_{AC}(\xi^2 \Delta \eta - \frac{\partial F}{\partial \eta}), \tag{16}$$

where L_{AC} is the mobility. For simplicity, we adopt the same ξ as in the Cahn-Hilliard equation. This is known as the Allen-Cahn equation [Allen and Cahn 1972], which appears to have not been previously used in computer graphics.

4.2 Governing Equations

We make use of the momentum equation for motion, the Cahn-Hilliard equation for the evolution of the concentration variable, and the Allen-Cahn equation for the evolution of the phase variable. We assume the system has M materials with N possible phases.

$$\frac{Du}{Dt} = \frac{1}{\rho}(-\nabla p + \nabla \cdot s) + f_{\text{ext}}, \tag{17}$$

$$\frac{Dc_m}{Dt} = \nabla \cdot (L_{CH} \nabla \mu_m), \tag{18}$$

$$\mu_m = \frac{\partial F}{\partial c_m} - \xi^2 \Delta c_m, \tag{19}$$

$$\frac{D\eta_k}{Dt} = L_{AC}(\xi^2 \Delta \eta_k - \frac{\partial F}{\partial \eta_k}), \tag{20}$$

where s is the deviatoric stress for a solid (see Section 3) or the viscosity tensor for a fluid (when Equation (17) becomes the Navier-Stokes equation). c_m and η_k are respectively the m-th material's concentration and the k-th phase of a given particle. For simplicity, we set $\xi = 0.01$ regardless of the type of the material or the state of phase. L_{CH}, L_{AC} are both material dependent. However, unlike density and viscosity, the values of L_{CH}, L_{AC} are set constant during simulation for simplicity. The external force f_{ext} includes gravity, as well as interactive forces between materials [Yang et al. 2015], etc. For a consistent labeling convention, we use subscripts k, l to represent phases, m for materials, and i, j for particles.

The concentrations of the materials and the phase variables must satisfy the constraints:

$$\sum_{m=0}^{M-1} c_m = 1, \quad \sum_{k=0}^{N-1} \eta_k = 1, \quad c_m, \eta_k \in [0, 1]. \tag{21}$$

To meet the constraints, we adopt a post-correction process, as do Ren et al. [2014], Yang et al. [2015] and Yan et al. [2016]. For simplicity, we use c, η to represent the vector of materials and phases of a given particle in the rest of this paper. It is a vector of three elements, whose entries with indices 0, 1, 2 represent the proportions of solid, liquid and gas respectively. Since we focus on the first two phases, η_2 is always 0 in this work.

Temperature is important when simulating phenomena involving phase change. To model heat flow, we use a similar evolution

equation to that used by Stomakhin et al. [2014], adding an extra source term for the phase transition:

$$C_H \frac{dT}{dt} + Z \frac{d\eta}{dt} = \nabla \cdot (C_T \nabla T), \tag{22}$$

where C_H is the heat capacity per unit volume, Z is the latent heat per unit volume, and C_T is the thermal conductivity. For each particle at a given time, the phase change is considered to occur between two phases, e.g., phase k, l (in this work, solid and liquid); the η in Equation (22) can be either η_k or η_l, while Z should change its sign accordingly.

Every particle is considered to (potentially) be composed of multiple materials and multiple phases, but unlike Ren et al. [2014] and Yan et al. [2016], we simply take it as a whole. We thus compute aggregate values for every particle at the beginning of each time step, for each of the required quantities: density, viscosity, heat capacity per unit volume, latent heat per unit volume, and thermal conductivity. For simplicity, we use weighted averages:

$$A = \sum_{m=0}^{M-1} c_m A_m, \tag{23}$$

$$B = \sum_{k=0}^{N-1} \eta_k B_k, \tag{24}$$

where m and k are respectively the material and phase indices. A and B are continuous variables; the former is mainly influenced by the type of the material, and the latter by the state of the phase, respectively. In this paper, only the deviatoric stress is of the latter type.

4.3 Unified Helmholtz Free Energy Density

We use c and η respectively to describe the concentrations of materials and the states of phases. As explained in Section 4.2, the evolutions of c and η are driven by the Helmholtz free energy density F. In contrast to the work of Yang et al. [2015], our Helmholtz free energy density works on both c and η. We thus call it the *unified Helmholtz free energy density*. It plays a key role in guiding phase-field multi-material simulations.

The unified Helmholtz free energy density must be related to both the concentration fraction c and phase fraction η. Since phase change typically depends on temperature, it must be taken into consideration. Garcke et al. [2004] defined the bulk free energies for individual phases. Since our simulations involve multiple phases, with the help of the interpolating function q, we define the final free energy function for phase-field multi-material simulation to be:

$$F = \sum_{k,l=0,k<l}^{N-1} \sum_{m=0}^{M-1} \left(c_m (Z_{k,l})_m \left(\frac{T - (\tilde{T}_{k,l})_m}{(\tilde{T}_{k,l})_m} \right) q(\eta_k) - q(\eta_l) \right) + \frac{R_G}{V_M} \sum_{m=0}^{M-1} \left(T c_m \ln(c_m) - C_V T \left(\ln(T) - 1 \right) \right), \tag{25}$$

where $(Z_{k,l})_m$ and $(\tilde{T}_{k,l})_m$ are respectively the latent heat per unit volume and the phase transition temperature of the pure material m when a phase transition occurs from phase k to phase l. $(Z_{k,l})_m = (Z_{l,k})_m$, $(\tilde{T}_{k,l})_m = (\tilde{T}_{l,k})_m$ for simplicity. C_V is the specific heat, which is assumed to be independent of c and η. The molar volume V_M is taken to be a constant. R_G is the gas constant. If c_m is smaller than a threshold (10^{-4} in our examples), the particle is considered to be free of the m-th material. q is a monotone function on $[0, 1]$ used for interpolation, satisfying $q(0) = 0$ and $q(1) = 1$; it is chosen according to the situation. For example, if we use $q(\eta_k) = \eta_k^2 (3 - 2\eta_k)$, $k \in \{0, 1, 2\}$, it satisfies $q'(0) = q'(1) = 0$, which ensures that the minima of the free energy are at $\eta_k = 0$ and $\eta_k = 1$ (i.e. pure phase states) regardless of the temperature. However, when simulating a single material's phase transition, the phase-change process will not be triggered if only one phase exists in the initial stage. We then instead use $q(\eta_k) = \eta_k$.

The proposed model can be used to capture simulations with multiple phases including solid, liquid and gas. In this paper, we simply focus on two phase examples involving solid and liquid. If additional constraint are introduced [Macklin et al. 2014], our approach can be extended to consider gaseous phase, which is not the main focus of this work. Thus, N in Equation (25) is set to 2 in our examples. Then only $(Z_{0,1})_m$ and $(T_{0,1})_m$ appear in Equation (25), to model phase change between liquid and solid.

4.4 Continuous Interfaces between Phases

Materials in different phases obey different physical laws. For instance, in Equation (17), s denotes the deviatoric stress for solids, but the viscosity tensor for fluids. To obtain this value, the velocity gradient is needed. When calculating the velocity gradient in standard SPH, only neighboring particles in the same state are taken into consideration. This is problematic if particles have mixed phase. To calculate the aggregate deviatoric stress, deviatoric stresses for both liquid and solid phases are needed. Each depends on its own velocity gradient, while the particle only has a bulk velocity. To address this issue, previous works simply set each particle to a particular phase according to its concentration fraction, temperature and so on. However, we adopt a continuous phase interface: a particle at the interface can be in multiple phases. We resolve this issue by computing the velocity gradient for phase k of particle i as follows:

$$(\nabla u_i)_k = \sum_j V_j \frac{(\eta_k)_i + (\eta_k)_j}{2} (u_j - u_i) \nabla W_{ij}, \tag{26}$$

where W_{ij} is a symmetric kernel function; we use a spiky kernel for gradient calculation following the work by Müller et al. [2003]. V_j is the volume of the neighboring particle j. $(\eta_k)_i$ denotes the k-th phase variable for particle i. As we only consider solids and liquids, we use $(\nabla u_i)_k$ to calculate the appropriate tensor for phase k. Finally, as we assume that the deviatoric stress is mainly influenced by phase, we use Equation (24) to obtain the aggregate deviatoric stress. The whole process is summarised in Algorithm 1.

The use of continuous interfaces is of benefit in the implementation of a multi-phase, multi-material framework, as no extra parameter is needed to label the phase (solid, liquid or gas). This feature also avoids the computational effort needed to update such labels.

The phases of particles can be mixed, thereby closer to real physical behavior. Although existing methods can produce similar visual effects, our method offers a unified treatment to seamlessly and easily handle multi-phase, multi-material simulations including transitions between phases.

Algorithm 1: Aggregate Deviatoric Stress Computation

Input: η, u , s from the last step
Output: s for the current step
1: **for all** particles **do**
2: **for all** phases k **do**
3: compute $(\nabla u)_k$ according to Equation (26)
4: compute s_k using $(\nabla u)_k$: if phase k is solid, use Equation (5), if liquid, use the standard viscosity tensor
5: **end for**
6: compute aggregate s using Equation (24)
7: **end for**

Algorithm 2: Simulation Loop for Multi-Phase, Multi-Material Modelling Framework

1: **for all** particles **do**
2: compute aggregate values using Equation (23)
3: **end for**
4: **for all** particles **do**
5: compute thermal diffusion according to Equation (22), and update the temperature
6: compute pressure using Tait's equation
7: **end for**
8: **for all** particles **do**
9: update c and η using the Cahn-Hillard Equation (18) and the Allen-Cahn Equation (20) respectively
10: **end for**
11: **for all** particles **do**
12: compute pressure force using standard SPH approach
13: compute deviatoric stress using Algorithm 1
14: compute artificial viscosity using Equation (27) and stress following [Monaghan 2000]
15: compute external forces
16: **end for**
17: **for all** particles **do**
18: advance the particle
19: **end for**

4.5 Artificial Viscosity and Stress

When simulating solids including deformable bodies and granular materials using particle-based methods, physical oscillations are often observed due to instability of the numerical solutions. They appear mostly in the initial stages of simulation, particularly if the Courant–Friedrichs–Lewy (CFL) condition is not satisfied (e.g., a particle travels more than a certain fraction of its support radius in one time step). To improve numerical stability and to damp out undesirable oscillations, a dissipative term may be introduced into the pressure term, as suggested by Yan et al. [2016], who pointed out that viscosity plays an important role in preventing particle penetration. We therefore also apply an artificial viscosity term to handle fluid-solid coupling. This term is given by:

$$\Pi_{ij} = \begin{cases} \nu_{ij} \dfrac{-ac_s\Omega_{ij} + b\Omega_{ij}^2}{\rho_{ij}}\delta, & \mathbf{V}_{ij}\cdot\mathbf{x}_{ij} < 0 \\ \mathbf{0}, & \mathbf{V}_{ij}\cdot\mathbf{x}_{ij} \geq 0 \end{cases}, \quad (27)$$

where i, j are particle indices, $\Omega_{ij} = h\mathbf{V}_{ij}\cdot\mathbf{x}_{ij}/(|\mathbf{x}_{ij}|^2 + 0.01h^2)$, $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$, $\mathbf{V}_{ij} = \mathbf{V}_i - \mathbf{V}_j$, a, b are viscosity coefficients, c_s is the speed of sound, and ν_{ij}, ρ_{ij} are respectively the average viscosity coefficient and density of particles i and j. We precompute the aggregate viscosity and density of every particle at the beginning of every step, to avoid treating miscible and immiscible particles in different ways [Ren et al. 2014; Yan et al. 2016].

Tensile instability is commonly observed in SPH simulations for both liquids and solids. We overcome this problem by using the artificial stress term derived by Monaghan [2000].

The final SPH formulation for the momentum equation in conjunction with the artificial viscosity and stress of particle i is given by:

$$\frac{Du_i}{Dt} = \sum_j \left(\frac{\sigma_i}{\rho_i^2} + \frac{\sigma_j}{\rho_j^2} - \Pi_{ij} + (R_i + R_j)\right)\nabla W_{ij} + (f_{\text{ext}})_i, \quad (28)$$

where R_i is the artificial stress of particle i. We adopt the approximations of spatial derivatives for SPH simulations suggested by Ihmsen et al. [2014].

4.6 Overall Algorithm

Pulling all of the above ideas together, the main loop of our simulation approach is as given in Algorithm 2.

5 APPLICATIONS

We have proposed use of a concentration variable c and a phase variable η to describe how concentration and phase evolve separately. Table 1 shows how these may be controlled to simulate various challenging real-world phenomena. For instance, to capture phase transitions such as melting and solidification, we enable the evolution of phase, while not allowing concentrations to evolve(i.e.

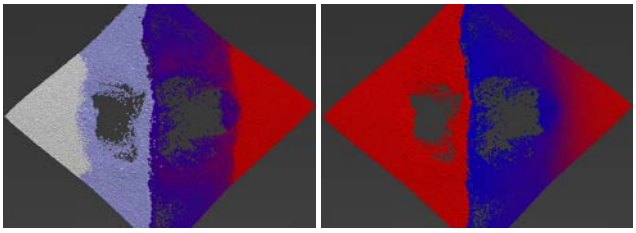

Fig. 8. Concentration and phase evolution. Water is poured onto two piles of soluble and insoluble grains. The water particles are invisible. Left: diffusion state of concentration at some given time (colors: blue: water, red: soluble granular material, white: insoluble but wettable granular material). Right: diffusion state of phase at the same time (color: blue: liquid, red: solid).

Table 1. Controlling concentration and phase evolution for various phenomena.

Phenomena	Concentration Evolution	Phase Evolution
Solid-Fluid Coupling	✗	✗
Phase Transition	✗	✓
Wetting	✓	✗
Dissolving	✓	✓

Table 2. Performance; particle numbers change in some examples as water is added; our method is about 3 times faster than [Yan et al. 2016]

Example	materials	phases	particles	ms/step
Deformable Jelly	1	1	50k	9.6
Dry Sand ($H/R = 0.9$)	1	1	48k	10.2
Dry Sand ($H/R = 2.0$)	1	1	100k	15.0
Butter Bunny	1	2	276k	24.1
Egg	2	2	180k	20.8
Dissolving Bunny (our method)	2	2	110k–260k	36.2
Dissolving Bunny ([Yan et al. 2016])	2	2	110k–260k	96.4
Soluble/wettable Granules	3	2	200k–400k	46.4
Bubble Tea	4	2	320k–600k	61.0

we set the mobility to 0). We enable both concentration and phase to evolve when simulating solid dissolution.

To simulate phase-change phenomena, we set the original phase to 0, and the target phase to 1. The temperature T in Equation (25) triggers the evolution of phase. For cases including solid wetting and dissolution, the temperature does not play a key role, so we set it to a constant. The evolution of concentration occurs at interfaces according to the Cahn-Hillard equation (18). We set the monotone function $q(\eta)$ in Equation (25) to $\eta_k^2(3-2\eta)$, so the evolution of phase starts at the interfaces of phases thanks to the Laplacian term in Equation (20) (The SPH formulation of the Laplacian operator contains a gradient term [Ihmsen et al. 2014]). To capture phenomena like dissolution of solids, the evolution of concentration interacts with the evolution of phase, and both only occur at interfaces. Figure 8 shows a snapshot of the differences in concentration and phase evolution in the same scenario as Figure 11 at a given time.

Decoupling the concentration and the phase leads to a straightforward and flexible approach to capture a variety of multi-material and multi-phase interactions as illustrated in Table 1.

6 RESULTS

6.1 Examples

In this section, we present various examples to demonstrate the range of simulations that our method can handle; supplementary videos are also provided. Key parameters used in the simulations are given here; further details of parameters can be found in the supplementary material.

Deformable Jelly: the proposed elastoplastic model can simulate both purely elastic and plastic object behavior, as shown in Figure 4. An elastic jelly (left) and two plastic jellies (middle and right) are dropped onto a plane. The elastic jelly rebounds and recovers its shape, while the plastic jellies deform irreversibly with different degrees of plasticity.

Dry and Wet Sand: Figures 5 and 7 shows that our elastoplastic model is capable of simulating both dry, non-cohesive sand, and cohesive, wet sand by choosing whether to use return mapping algorithms or not. Our SPH-based approach is about 4 times faster than the MPM-based one. Our method can simulate the collapses of dry granular columns with different angles (see Figures 7b–7d). We also demonstrate that the behavior of the collapse of an initially vertical cylinder of granular materials depends on the aspect ratio H/R, where H and R are its initial height and radius respectively [Lube et al. 2004] (see Figures 7d and 7g).

Melting and Solidification: our multi-phase, multi-material model can be used to capture a wide range of phase-changing phenomena including melting and solidification. Figure 10 presents the melting of a butter bunny on a hot pan. The simulation involves one material (butter) with two phases (solid and liquid). We capture melting by enabling diffusion of the phase variable. Since the initial conditions are single-phase, we set the monotone function $q(\eta_k) = \eta_k$. Figure 1 demonstrates the frying of an egg. This is a two-material (egg white and egg yolk) system with two phases (solid and liquid). The egg white and egg yolk do not mix in this case, so any given particle contains only one material. We initialize the egg white and the egg yolk with different values of thermal conductivity [Yang et al. 2016] to ensure different speeds of spread of phase change.

Dissolving Bunny: Figure 9 demonstrates the dissolution process of a bunny. We also provides a comparison with results by Yan et al. [2016] (see the supplementary video). Our method achieve comparable results and is about 3 times faster (see Table 2), as we avoid the use of computationally expensive drift velocity and an hypoplastic model.

Soluble and Wettable Granular Materials: Figure 11 show advantages of our model over previous methods. Two piles of soluble coffee granules and sands interact with water from above poured onto them. To simulate soluble coffee granules, we enable diffusion of both concentration and phase variables, but to simulate insoluble sand grains, we disable the diffusion of the phase variable. (The density ratio of granular material to water is $2:1$ in each case). Here, we assume the wettable sand and water slightly mix with each other until reaching a saturation point: we assume a sand particle can contain at most 5% water.

Bubble Tea: to demonstrate challenging real-world phenomena with multiple materials and phases, as shown in Figure 12, we simulate the making of bubble tea, a drink that contains soluble coffee granules, wettable elastoplastic tapioca pearls, and two liquids (water and milk). Our method can capture a variety of complex interactions in a versatile way. (The density ratios of coffee granules vs. tapioca pearls vs. water vs. milk are $2:1.5:1:1$). In this scenario, we use particle boundary; to prevent particle penetration, larger viscosity and artificial viscosity are adopted. We add vorticity confinement [Fedkiw et al. 2001; Macklin and Müller 2013] to make the fluid less viscous.

We implemented our algorithm using CUDA on an NVIDIA GeForce GTX1080 11GB GPU and 8 Intel Core i7-3770 CPUs; the CPUs are relatively unimportant. Performance for the examples described above is given in Table 2. The particle numbers shown include both fluid and boundary particles. Timing presented are averaged running time over the entire simulation. The time step is set to around 10^{-3} s, which is sufficient to achieve stable simulation and prevent particle penetration.

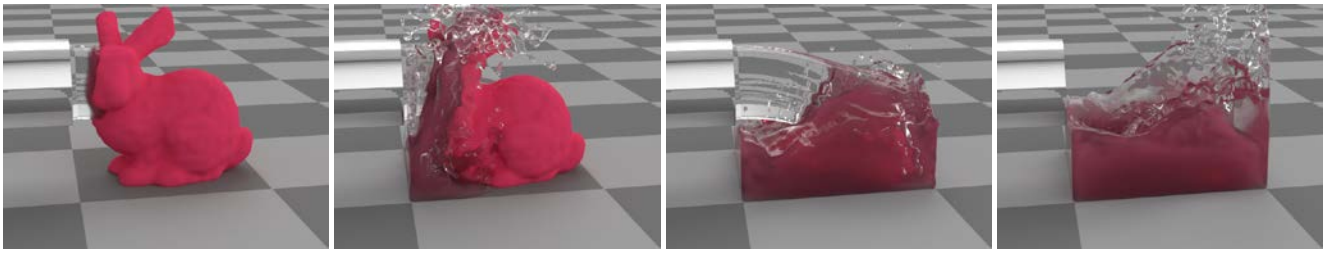

Fig. 9. Soluble bunny. A deformable bunny dissolves in the water pouring from the pipe, coloring the water red.

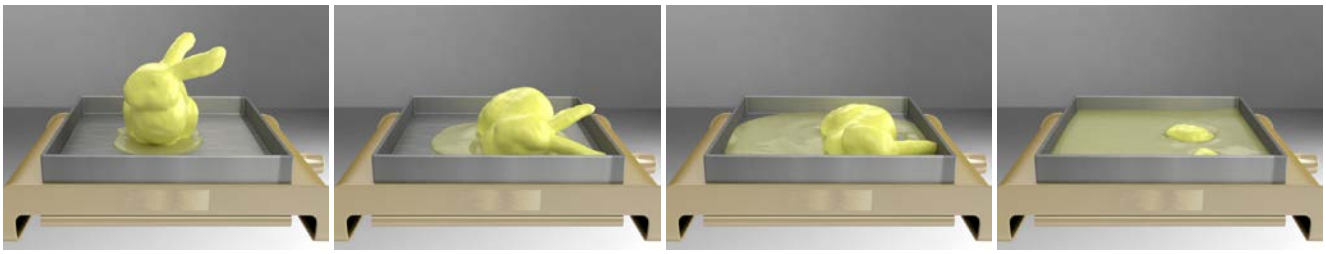

Fig. 10. Melting a butter bunny on a hotplate. Phenomena such as wave-like spread and free sliding behavior of the unmelted butter can be observed.

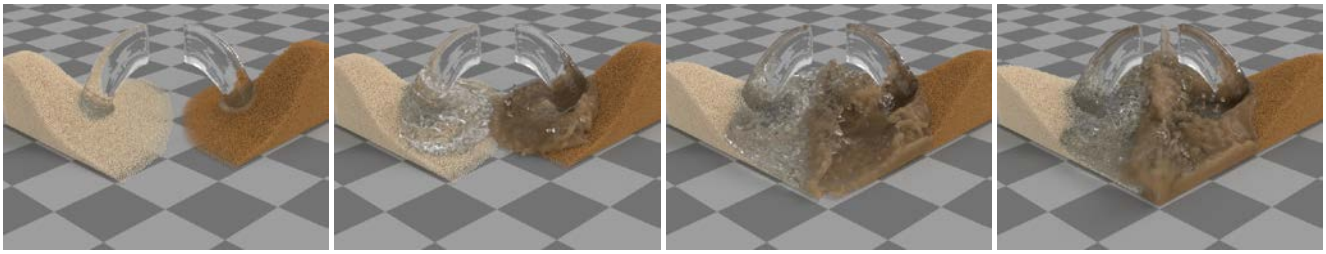

Fig. 11. Soluble and wettable granular materials. Piles of coffee granules (brown) and sands (yellow) interact with water poured from above onto them. Left: the sand is merely wettable if phase evolution is disabled. Right: the coffee granules dissolve, achieved by enabling the evolution of both phase and concentration.

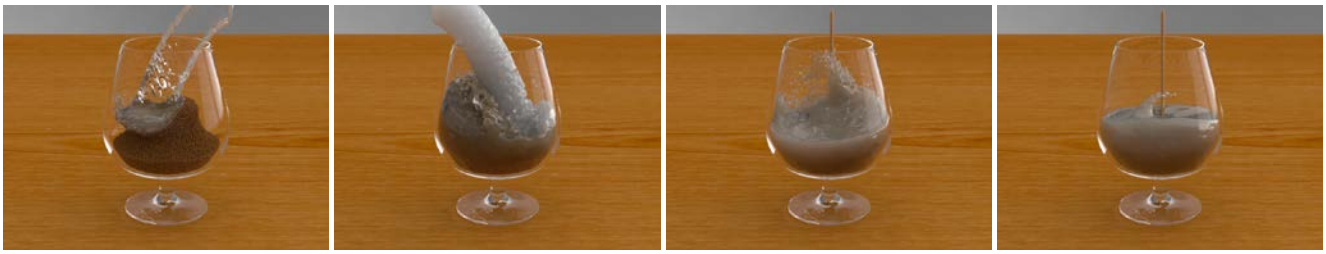

Fig. 12. Making of Bubble Tea. A granular coffee column collapses in the goblet, then dissolves in water (pouring from right) and milk (pouring from left).

6.2 Discussion

Yang et al. [2015] also adopted Cahn-Hilliard equation. Their work focused on interactions among multiple fluids; no solids are considered. Furthermore, they proposed a user-defined Helmholtz free energy density function to artistically control the interactions; while in this paper, a unified Helmholtz free energy density function is introduced to drive the evolutions of both concerntration and phase.

Although our examples show some similarities with those of Yan et al. [2016], the methods used to produce the results are completely different. Yan et al. use drift velocity to capture the diffusion of con-centration; thus, in every particle, each concentration has its own velocity. Our method however uses a unified energy to drive the dif-fusion of both concentration and phase. Every particle is considered as a whole; each concentration within a single particle shares the

same velocity. Our method is much simpler to implement and is also more efficient, overall being about 3 times faster than Yan et al.'s method (see Table 2). Furthermore, Yan et al. use an elastoplastic model to simulate deformable bodies and a hydroplastic model for granular materials, which adds complexity. By improving the elastoplastic model to make it capable of modeling granular materials, we are able to use a single general formulation for both cases. The proposed method models landslides with different failure patterns as well as collapsing granular columns with varying aspect ratios, which have been largely overlooked in the graphics community. All the multi-material, multi-phase examples presented could not previously be achieved in such a straightforward, easy-to-implement, fast and unified way.

7 CONCLUSION AND FUTURE WORK

We have extended the energy-based multi-fluid model [Yang et al. 2015] to simulate a wider range of multi-phase, multi-material system including both liquids and solids, by incorporating the phase-field method. Our proposed method can distinguish between phases and materials, and treats them independently. By doing so, it becomes simpler to capture challenging real-world phenomena involving multiple materials and multiple phases. Our examples demonstrate the effectiveness of the proposed unified simulation framework.

We use a linear elastoplastic model, as it is simple and versatile, and enables modeling of a large variety of solid behavior, including deformable bodies, granular materials, and cohesive soils. However, when animating purely elastic objects, this model is unstable to capture large elastic deformation using SPH, due to numerical error of stress-rate accumulation, as simulation continues. In our experiments, smaller time steps can ensure longer time of stable simulations, but it cannot solve the problem. More investigation is needed to address this issue.

The proposed model is in principle capable of capturing gaseous phenomena. However, special techniques are needed for smoke simulation in current SPH based solvers [Macklin et al. 2014; Ren et al. 2014], and these are not compatible with the governing equations for liquids or solids, making it tricky to incorporate smoke within our new framework. Since only two phases, e.g., liquid and solid are considered in this work, it is possible to use a single value of η (see explanations in Section 4.1) rather than a vector η to represent the state of materials. We use the vector of phase η to leave room for future extensions.

In this work, we have adopted the simplifying assumption that the phase variable acts on the whole particle. For instance, suppose a given particle is determined as 50% material A and 50% material B, and as 50% solid and 50% liquid; we do not distinguish how much of A or B is solid or liquid, but just describe this mixed particle as 50% solid and 50% liquid overall. Our experiments show that this simplification (compared to strictly using a separate phase variable for each material) does not prevent us from capturing many commonly observed real-world phenomena. However, phenomena with multiple phases and multiple materials are actually more complex. For instance, the given particle above could be that all material A is solid and all material B is liquid, or vice versa, or various combinations in between. In the future, we hope to investigate the degree to which it is beneficial to use a more precise representation.

Crystallization is also commonly observed in daily life, e.g. in ice formation on a window. The process of crystal growth involves both concentration and phase evolution, so in principle, is within the scope of our model. Further effort is needed in future to capture this phenomenon.

We use the weakly compressible SPH (WCSPH) in our work, as it can readily integrate into our method. However, it suffers from compressibility compared with the MPM solver (see Figure 5 and the supplementary video). The position based fluids (PBF) [Macklin and Müller 2013] and MPM have gained popularity in recent years. It would be promising to integrate our method to these two frameworks.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of China (Project Number 61521002), Research Grant of Beijing Higher Institution Engineering Research Center, EU FP7 project Dr.Inventor (FP7-ICT-611383) and FP7 Marie Curie IRSES project AniNex (612627), the U.S. National Science Foundation, and an EPSRC UK travel grant.

REFERENCES

Nadir Akinci, Markus Ihmsen, Gizen Akinci, Barbara Solenthaler, and Matthias Teschner. 2012. Versatile rigid-fluid coupling for incompressible S PH. *ACM Trans. Graph.* 31, 4 (Jul. 2012), 62:1–62:8.

Iván Alduán and Miguel A. Otaduy. 2011. SPH granular flow with friction and cohesion. *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2011), 25–32.

S.M Allen and J.W Cahn. 1972. Ground state structures in ordered binary alloys with second neighbor interactions. *Acta Metallurgica* 20, 3 (1972), 423–433.

Markus Becker, Markus Ihmsen, and Matthias Teschner. 2009. Corotated SPH for deformable solids. *Proceedings of the Fifth Eurographics Conference on Natural Phenomena* (2009), 27–34.

Markus Becker and Matthias Teschner. 2007. Weakly compressible SPH for free surface flows. *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), 209–217.

Nathan Bell, Yizhou Yu, and Peter J. Mucha. 2005. Particle-based simulation of granular material. *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), 77–86.

Ha H. Bui, Ryoichi Fukagawa, Kazunari Sako, and Shintaro Ohno. 2008. Lagrangian meshfree particles method (SPH) for large deformation and failure flows of geomaterial using elastic-plastic soil constitutive model. *Int. J. Numer. Anal. Meth. Geomech.* 32 (2008), 1537–1570.

John W. Cahn and John E. Hilliard. 1958. Free energy of a nonuniform system. I. Interfacial free energy. *Journal of Chemical Physics* 28, 2 (1958), 258–267.

Yuanzhang Chang, Kai Bao, Youquan Liu, Jian Zhu, and Enhua Wu. 2009. A particle-based method for viscoelastic fluids animation. (2009), 111–117.

W. Chen and E. Mizuno. 1990. *Nonlinear Analysis in Soil Mechanics: Theory and Implementation*. Elsevier, Amsterdam.

Wei Chen, Tong Qiu, and M. ASCE. 2012. Numerical Simulations for Large Deformation of Granular Materials Using Smoothed Particle Hydrodynamics Method. *International Journal of Geomechanics* 12, 2 (2012), 127–135.

Gilles Daviet and Florence Bertails-Descoubes. 2016. A Semi-Implicit Material Point Method for the Continuum Simulation of Granular Materials. *ACM Trans. Graph.* 35, 4 (2016), 102:1–102:13.

R. Fedkiw, J. Stam, and H. W. Jensen. 2001. Visual simulation of smoke. *ACM Trans. Graph.* (2001), 15–22.

Harald Garcke, Britta Nestler, and Bjorn Stinner. 2004. A diffuse interface model for alloys with multiple components and phases. *SIAM J. APPL. MATH* 64, 3 (2004), 775–799.

Dan Gerszewski, Haimasree Dhattacharya, and Adam W. Bargteil. 2009. A point-based method for animating elastoplastic solids. *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), 133–138.

J. Gray, J. Monaghan, and R. Swift. 2001. SPH elastic dynamics. *Computer Methods in Applied Mechanics and Engineering* 190, 49C50 (2001), 6641–6662.
Markus Ihmsen, Nadir Akinci, Marc Gissler, and Matthias Teschner. 2010. Boundary handling and adaptive time-stepping for PCISPH. *VRIPHYS* (2010), 79–88.
Markus Ihmsen, Jens Orthmann, Barbara Solenthaler, Andreas Kolb, and Matthias Teschner. 2014. SPH Fluids in Computer Graphics. *In Eurographics 2014 - State of the Art Reports* (2014), 21–42.
Markus Ihmsen, Arthur Wahl, and Matthias Teschner. 2012. High resolution simulation of granular material with SPH. *VRIPHYS* (2012), 53–60.
C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin. 2015a. The Affine Particle-In-Cell Method. *ACM Trans. Graph.* 34, 4 (2015), 51:1–51:10.
Min Jiang, Richard Southern, and Jian J. Zhang. 2015b. A Particle-based Dissolution Model using Chemical Collision Energy. *Proceedings of the 10th International Conference on Computer Graphics Theory and Applications* (2015), 285–293.
Ben Jones, Stephen Ward, Ashok Jallepalli, Joseph Perenia, and Adam W. Bargteil. 2014. Deformation Embedding for Point-Based Elastoplastic Simulation. *ACM Trans. Graph.* 33, 2 (Apr. 2014), 21:1–21:9.
Richard Keiser, Bart Adams, Dominique Gasser, Paolo Bazzi, Philip Dutré, and Markus Gross. 2005. A unified lagrangian approach to solid-fluid animation. *Proceedings of the Second Eurographics/IEEE VGTC Conference on Point-Based Graphics* (2005), 125–133.
Gergely Klár, Theodore Gast, Andre Pradhana, Chuyuan Fu, Craig Schroeder, Chenfanfu Jiang, and Joseph Teran. 2016. Drucker-Prager Elastoplasticity for Sand Animation. *ACM Trans. Graph.* 35, 4 (Jul. 2016), 103:1–103:12.
Toon Lenaerts, Bart Adams, and Philip Dutré. 2008. Porous flow in particle-based fluid simulations. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 49:1–49:8.
Toon Lenaerts and Philip Dutré. 2009. Mixing fluids and granular materials. *Computer Graphics Forum* 28, 2 (2009), 213–218.
Shiguang Liu, Qiguang liu, and Qunsheng Peng. 2011. Realistic simulation of mixing fluids. *The Visual Computer* (Mar. 2011), 241–248.
Gert Lube, Hernert E. Huppert, R. Stephen J. Sparks, and Mark A. Hallworth. 2004. Axisymmetric collapses of granular columns. *J. Fluid Mech.* 508 (2004), 175–199.
Miles Macklin and Matthias Müller. 2013. Position Based Fluids. *ACM Trans. Graph.* 32, 4 (2013), 104:1–104–12.
Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. 2014. Unified Particle Physics for Real-Time Applications. *ACM Trans. Graph.* 33, 4 (2014), 153:1–153:12.
J. J. Monaghan. 1992. Smoothed Particle Hydrodynamics. *Annual Review of Astronomy and Astrophysics* 30, 1 (1992), 543–574.
J. J. Monaghan. 2000. SPH without a tensile instability. *J. Comput. Phys.* 82, 1 (2000), 1–15.
Matthias Müller, David Charypar, and Markus Gross. 2003. Particle-Based Fluid Simulation for Interactive Applications. *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), 154–159.
M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. 2004. Point Based Animation of Elastic, Plastic and Melting Objects. *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), 141–151.
Matthias Müller, Barbara Solenthaler, Richard Keiser, and Markus Gross. 2005. Particle-Based Fluid-Fluid Interaction. *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), 237–244.
Rahul Narain, Abhinav Golas, and Ming C. Lin. 2010. Free-flowing granular materials with two-way solid coupling. *ACM Trans. Graph.* 29, 6 (Dec. 2010), 173:1–173:10.
Jinho Park, Younghwi Kim, Daehyeon Wi, Nahyup Kang, Sung Yong Shin, and Junyong Noh. 2008. A Unified Handling of Immiscible and Miscible Fluids. *Computer Animation and Virtual Worlds* 19, 3-4 (Sept. 2008), 455–467.
Andreas Peer, Markus Ihmsen, Jens Cornelis, and Matthias Teschner. 2015. An Implicit Viscosity Formulation for SPH Fluids. *ACM Trans. Graph.* 34, 4 (Jul. 2015), 114:1–114:10.
Bo Ren, Chenfeng Li, Xiao Yan, Ming C. Lin, Javier Bonet, and Shi-Min Hu. 2014. Multiple-fluid SPH simulation using a mixture model. *ACM Trans. Graph.* 33, 5 (Sept. 2014), 171:1–171:11.
H.F. Schwiger. 1994. On the Use of Drucker-Prager Failure Criteria for Earth Pressure Problems. *Computer and Geotechnics* 16, 3 (1994), 223–246.
X. Shao, Z. Zhou, N. Magnenat-Thalmann, and W. Wu. 2015. Stable and Fast Fluid-Solid Coupling for Incompressible SPH. *Comput. Graph. Forum* 34, 1 (Feb. 2015), 191–204.
Barbara Solenthaler and Renato Pajarola. 2008. Density contrast SPH Interfaces. *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2008), 211–218.
Barbara Solenthaler, Jürg Schläfli, and Renato Pajarola. 2007. A unified particle model for fluid-solid interactions. *Comput. Animat. Virtual Worlds* 18, 1 (Feb. 2007), 69–82.
Alexey Stomakhin, Craig Schroeder, Chenfanfu Jiang, Lawrence Chai, Joseph Teran, and Andrew Selle. 2014. Augmented MPM for phase-change and varied materials. *ACM Trans. Graph.* 33, 4 (Jul. 2014), 138:1–138:11.
Deborah Sulsky, Shi-Jian Zhou, and Howard L. Schreyer. 1995. Application of particle-in-cell method to solid mechanics. *Computer Physics Communications* 87, 1-2 (1995), 236–252.
A. Pradhana Tampubolon, T. Gast, G. Klar, C. Fu, J. Teran, C. Jiang, and K. Museth. 2017. Multi-species simulation of porous sand and water mixtures. *ACM Trans. Graph.* 36, 4 (2017), 105:1–105:11.
Gang Wang and Nicholas Sitar. 2004. NUMERICAL ANALYSIS OF PILES IN ELASTO-PLASTIC SOILS UNDER AXIAL LOADING. *17th ASCE Engineering Mechanics Conference* (2004), 13–16.
Xiao Yan, Yun-Tao Jiang, Chen-Feng Li, Ralph R. Martin, and Shi-Min Hu. 2016. Multiphase SPH Simulation for Interactive Fluids and Solids. *ACM Trans. Graph.* 35, 7 (Jul. 2016), 79:1–79:11.
Tao Yang, Jian Chang, Bo Ren, Ming C. Lin, Jian Jun Zhang, and Shi-Min Hu. 2015. Fast multiple-fluid simulation using helmholtz free energy. *ACM Trans. Graph.* 34, 6 (Oct. 2015), 201:1–201:11.
Tao Yang, Ming C. Lin, Ralph. R. Martin, Jian Chang, and Shi-Min Hu. 2016. Versatile Interactions at Interfaces for SPH-Based Simulations. *Proceedings of the 2016 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2016), 57–66.
Yahan Zhou, Zhaoliang Lun, Evangelos Kalogerakis, and Rui Wang. 2013. Implicit integration for particle-based simulation of elasto-plastic solids. *Computer Graphics Forum* 32, 7 (2013), 215–223.
Yongning Zhu and Robert Bridson. 2005. Animating sand as a fluid. *ACM Trans. Graph.* (2005), 965–972.

A DERIVATION OF THE ELASTOPLASTIC MODEL

In this work, we adopt an elastic-perfectly plastic model. The total strain rate tensor $\dot{\epsilon}$ is commonly composed of two parts, i.e., elastic strain rate $\dot{\epsilon}^e$ and plastic strain rate $\dot{\epsilon}^p$:

$$\dot{\epsilon} = \dot{\epsilon}^e + \dot{\epsilon}^p. \tag{29}$$

The generalized Hooke's law states that:

$$\dot{\sigma} = \mathbf{D}^e : \dot{\epsilon}^e = \mathbf{D}^e : (\dot{\epsilon} - \dot{\epsilon}^p). \tag{30}$$

where $\mathbf{D}^e$ is the fourth-order elasticity tensor, which can be formulated as:

$$\mathbf{D}^e = 2G\mathbf{T} + K\boldsymbol{\delta} \otimes \boldsymbol{\delta}, \tag{31}$$

where $\mathbf{T} = \mathbf{I} - \frac{1}{3}\boldsymbol{\delta} \otimes \boldsymbol{\delta}$ is the fourth-order deviatoric tensor. The plastic potential function ∂ specifies the direction to which the plastic strain develops. The plastic strain rate tensor may be computed by using the plastic flow rule:

$$\dot{\epsilon}^p = \dot{\lambda}\frac{\partial \partial}{\partial \sigma}. \tag{32}$$

Thus, substituting Equation (32) into Equation (30) gives:

$$\dot{\sigma} = \left(\mathbf{D}^e : \dot{\epsilon} - \dot{\lambda}\frac{\partial \partial}{\partial \sigma}\right). \tag{33}$$

During the plastic flow, the stress state must remain on the yield surface (see Figure 6). Thus, the consistency condition must be satisfied:

$$df = \frac{\partial f}{\partial \sigma} : \dot{\sigma} = 0. \tag{34}$$

This equation ensures that the new stress state $\sigma + d\sigma$ still satisfies the yield criterion after loading:

$$f(\sigma + d\sigma) = f(\sigma) + df = f(\sigma). \tag{35}$$

We use N and Q to represent the partial derivatives of f and ∂ (see Equations (3) and (4)) with respect to σ for convenience:

$$N := \frac{\partial f}{\partial \sigma} = \frac{1}{\sqrt{2}S}\mathbf{s} + \alpha\boldsymbol{\delta}, \quad Q := \frac{\partial \partial}{\partial \sigma} = \frac{1}{\sqrt{2}S}\mathbf{s} + \beta\boldsymbol{\delta}. \tag{36}$$

Substituting Equation (33) into Equation (34) gives:

$$N : \left(\mathbf{D}^e : \left(\dot{\epsilon} - \dot{\lambda}Q\right)\right) = 0, \tag{37}$$

and thus:

$$\lambda = \frac{N : D^e : \dot{\epsilon}}{N : D^e : Q} = \frac{1}{G + 9K\alpha\beta} \left(\sqrt{2G} : e - 3K\alpha \, \mathrm{Tr}(\dot{\epsilon}) \right) \quad (38)$$

Substituting Equation (38) into Equation (33), it gives:

$$\dot{\sigma} = 2G\dot{e} + \mathrm{Tr}(\dot{\epsilon})\delta - \dot{\lambda}\left(\sqrt{2G} + K\beta\delta \right). \quad (39)$$

The deviatoric part of $\dot{\sigma}$ is as follows:

$$\dot{s} = 2G\dot{e} - \dot{\lambda}\sqrt{2G}\, s. \quad (40)$$

B DRUCKER-PRAGER APPROXIMATIONS

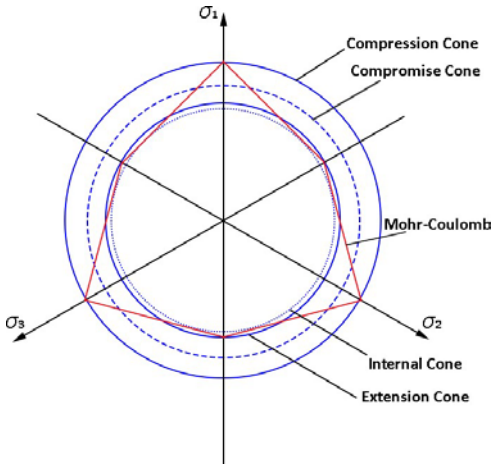

Fig. 13. Drucker-Prager approximations.

The Drucker-Prager yield criterion is a pressure-dependent model to estimate the stress state for determining whether a material has failed or undergone irreversible plastic deformation. The Drucker-Prager yield surface is a smooth version of the Mohr-Coulomb yield surface. A variety of Drucker-Prager approximations have been proposed to match the Mohr-Coulomb yield surface as shown in Figure 13.

- **Compression cone:** matching Mohr-Coulomb yield surface in triaxial compression;
- **Extension cone:** matching Mohr-Coulomb yield surface in triaxial extension;
- **Internal cone:** inscribed inside Mohr-Coulomb yield surface;
- **Compromise cone:** an average between extension and compression approximations.

Table 3 summarizes determination of model parameters with respect to friction angle ϕ, cohesive coefficient k_c, and dilatancy angle ψ [Wang and Sitar 2004].

Experiments show that different failure patterns arise using different approximations [Schwiger 1994]. The appropriate choice of approximation is application-dependent. When using an elastoplastic model, elastic and plastic deformations often co-occur. In this work, we use the compression cone model, as small elastic deformation can be ignored in comparison with plastic deformation when simulating granular materials.

Table 3. Model parameter determination.

D-P approximation	α	β	γ
Compression Cone	$\frac{2\sin\varphi}{\sqrt{3}(3-\sin\varphi)}$	$\frac{2\sin\psi}{\sqrt{3}(3-\sin\psi)}$	$\frac{6k_c\cos\varphi}{\sqrt{3}(3-\sin\varphi)}$
Compromise Cone	$\frac{2\sin\varphi}{3\sqrt{3}}$	$\frac{2\sin\psi}{3\sqrt{3}}$	$\frac{6k_c\cos\varphi}{3\sqrt{3}}$
Extension Cone	$\frac{2\sin\varphi}{\sqrt{3}(3+\sin\varphi)}$	$\frac{2\sin\psi}{\sqrt{3}(3+\sin\psi)}$	$\frac{6k_c\cos\varphi}{\sqrt{3}(3+\sin\varphi)}$
Internal Cone	$\frac{\sin\varphi}{\sqrt{3}(3+\sin^2\varphi)^{1/2}}$	$\frac{\sin\psi}{\sqrt{3}(3+\sin^2\psi)^{1/2}}$	$\frac{3k_c\cos\varphi}{\sqrt{3}(3+\sin^2\varphi)^{1/2}}$