

Energy-based Dissolution Simulation using SPH sampling

Min Jiang¹, Richard Southern*¹ and Jiang J. Zhang¹

¹National Centre for Computer Animation, Bournemouth University, UK

January 19, 2018

Abstract

A novel unified particle-based method is proposed for real-time dissolution simulation which is fast, predictable, independent of sampling resolution and visually plausible. The dissolution model is derived from Collision Theory and integrated into a Smoothed Particle Hydrodynamics (SPH) fluid solver. Dissolution occurs when a solute is submerged in solvent. Physical laws govern the local excitation of solute particles based on kinetic energy: when the local excitation energy exceeds a user specified threshold (activation energy), the particle will be dislodged from the solid. Solute separation during dissolution is handled using a new GPU-based region growing method. The use of SPH sampling for both solute and solvent guarantees a predictable and smooth dissolution process and provides user control of the volume change during the phase transition. A mathematical relationship between the activation energy and dissolution time allows for intuitive artistic control over the global dissolution rate. We demonstrate this method using a number of practical examples, including antacid pills dissolving in water, hydraulic erosion of non-homogeneous terrains and melting.

This is the peer reviewed version of the following article:

Jiang M, Southern R, Zhang JJ. Energy-based dissolution simulation using SPH sampling. *Comp Anim Virtual Worlds*. 2018;e1798. doi:10.1002/cav.1798

which has been published in final form at <https://doi.org/10.1002/cav.1798>. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Self-Archiving.

*rsouthern@bournemouth.ac.uk

1 Introduction

The reliance of films and games on realistic fluid simulations has led to the popularity of this research topic. Most related work has focused on the behaviour of fluid itself, such as spray, waves and bubbles [1, 2, 3] or the interaction with solids [4, 5, 6, 7]. Comparatively few researchers have tackled the challenging problem of the chemical reaction between fluid and solids, which, under the right conditions, can lead to the solid mixing with the liquid: a natural process known as *dissolution*. The solid in this case is called the *solute* and the liquid is the *solvent*. We can observe this process when an antacid tablet is dropped into water — bubbles are generated by a chemical reaction and the tablet shrinks due to mass transfer. However due to the complex physical and chemical behaviours of dissolution few publications have explored this problem.

In this paper we propose an energy-based dissolution model based on *chemical Collision Theory* for applications in visual effects and game production. A kinetic energy function is introduced to measure the particle excitation of the solute. The concept of *volume mass* for the energy calculation is proposed to ensure the consistent dissolution results even when the object sampling resolutions are different. This property of sampling resolution independence is advantageous when performing pre-visualization of the dissolution process, allowing animators to quickly preview the dissolution behaviour with a low resolution sampled solute.

To guarantee that the dissolution process is predictable and avoid the problem of the fluid penetrating the boundary of the solute, an even distribution of the solute particle is essential. We use the SPH sampling method [8] to compute the solute particle distribution. We demonstrate experimentally that the SPH sampling method achieves better visual effects than random sampling or simple grid sampling and also gives more control over volume change during phase transition than other blue noise sampling methods.

A solute particle will dislodge from the solid if the local energy exceeds a user specified *activation energy*. While physically justified, this threshold does not provide a user with an intuition of the total dissolution time. We find that the activation energy can be approximated as a function of the inter-facial surface area of the solvent/solute and the total dissolution time, providing animators with a tool to more accurately predict the total dissolution rate.

Our energy-based dissolution simulation coupled with an SPH sampled solute and solvent ensures that the dissolution behaviour is physically and chemically plausible, while supporting features such as object separation and sharp feature rounding. We demonstrate this using examples such as rotating and translating solids in fluids and pouring fluids onto dissolving objects. Due to the similarity between dissolution and hydraulic erosion, we are able to simulate erosion without any additional computational overhead. Our model can also approximate melting behaviour with the introduction of temperature. Our particle-based method exhibits real-time performance using a parallel implementation.

This work expands on our original paper [9] in the following ways:

- we provide a considerably more detailed and intuitive derivation of dissolution model based on Chemical Collision Theory;
- we investigate the effect of the solute sampling method on the quality of the result;
- we demonstrate qualitatively the effect of different sampling methods on the fluid–solid interface in particle-based fluid;
- we introduce a new parallel region growing algorithm which greatly accelerates the segmentation of particle-based object representations and has a wide variety of applications in graphics, and
- we extend the dissolution model to incorporate temperature and approximate melting behaviour.

2 Related Work

In most observable systems, the most frequent solvent is fluid and solute is solid. There have been many studies on the physical interactions between fluids and solids. G enevaux *et al.* [10] study the fluid–solid interaction by representing solid with linked point masses. Batty *et al.* [4] apply a pressure projection method with kinetic energy minimization to incorporate irregular boundary geometry into standard grid–based fluid simulations. However these methods are difficult to be integrated into a dissolution simulation.

Both [11] and [12] treat objects as if they are made of fluid. To maintain the object rigidity, Carlson *et al.* [11] identify the object by velocity field, and Amada [12] apply a corrective step to bring back the rigidity of the solid. Harada *et al.* [13] simulate the interaction using collision detection. To prevent fluid particles intersecting the solid boundary, both Amada [12] and Harada *et al.* [13] use penalty force methods which takes several frames to push the particles out. Becker *et al.* [14] use a predictor–corrector scheme with direct force to deal with the boundary conditions which ensures no–penetration, but this method requires two additional neighbour queries for collision detection which is expensive to compute.

Schechter and Bridson [6] achieve the no–separation and no–slip boundary condition by sampling the air and solid surface as ghost particles. Akinci *et al.* [15] use corrected density to compute the pressure and viscosity force to avoid the sticking artefacts. Both solve the boundary problem, making an extension to support dissolution simulation.

While these methods tackle the problem of physical interaction between the fluid and solid, they do not consider the chemical interaction between them. Relatively few attempts have been made to simulate the complex chemical reaction between fluid and solid.

Solenthaler *et al.* [16] use an elasticity model for particle–based objects and model temperature for phase changes, allowing them to simulate objects melting in hot liquid. Stomakhin *et al.* [17] introduce an augmented Material Point Method for phase change and simulated butter melting in pan. Melting offers a similar visual result to dissolution, but their underlying function is a heat model which differs from the chemical process presented in this paper. We extend our dissolution model to mimic a simplified melting effect in Sec. 8.3.

Kim *et al.* [18] simulate bubbles around dissolving objects, focusing on simulating the dispersed bubble flow. Their dissolution model is approximated by generating offsets of a level–set representation of the solute, which has little relationship with actual dissolution behaviors. Shin *et al.* [19] simulate solids dissolving in liquids by modelling solid mass transfer. They treat the solvent and solute as different fluids, and use multiple level–sets to track mixing surfaces which need to be updated whenever the reactions and interactions occurred — an expensive but necessary step. They also do not consider the object breaking into separate pieces during dissolution, a problem later solved by Wojtan and co–workers [20]. Both methods use a level–set representation to guide the object surface, making the dissolution result largely dependant on the level–set grid resolution. Furthermore these previous approaches did not consider the relationship between the dissolution time and the mass transfer of the solute, meaning that predicting the overall dissolution time was an exercise in trial and error. In this paper we propose an experimental model for predicting dissolution rate by activation energy, and the dissolution process is independent of solute sampling resolutions.

Dissolution also shares many properties with hydraulic erosion. Beneš *et al.* [21] propose a solution for erosion simulation by using Navier–Stokes equations, while Wojtan *et al.* [20] animate corrosion and erosion by driving the fluids with a finite difference simulation and presenting the solid by advecting the level sets inward. Kriřtof *et al.* [22] use an SPH–based method to simulate erosion, where the erosion model was adapted from an Eulerian approach. In this paper we demonstrate two practical erosion examples, including one which shows the specification of non–homogeneous solutes forming a natural layering stratum.

In previous work the erosion rate is derived from the power law [22, 20] which unfortunately requires a large number of parameters to control, making it impractical for use by an animator. Instead, our method requires only one single parameter which globally controls the rate of dissolution.

The basic dissolution model presented in this paper has been previously published in [9]. In this work we investigate the effect of the solute sampling, demonstrating qualitatively the properties of Blue Noise sampling which has broad reaching applications in any method utilising particle-based representations of solids. We introduce a new parallel region growing algorithm which greatly accelerates the segmentation of particle-based object representations. We also extend the dissolution model to incorporate temperature and melting behaviour.

3 Dissolution Model

In our dissolution model, the physical and chemical reactions between the solute and solvent are elegantly simulated within a single SPH solver. We derive a kinetic energy function to calculate the local excitation of the solute particle. The use of volume mass guarantees the dissolution independent of the sampling resolution. Solid solutes are distributed using SPH sampling method [8]. GPU-based region growing method is used for detecting the occurrence of separation during dissolution. We also experimentally deduce the relationship between the dissolution configuration and the dissolution time. The framework of our dissolution model is shown in Fig. 1. A list of symbols used in this paper is included in Table 1.

Symbol	Description	Defined	Symbol	Description	Defined
A	any scalar quantity	pg 6	\mathbf{x}	particle position	pg 6
W	kernel function	pg 6	h	smoothing length	pg 6
\mathbf{v}_r	relative velocity	pg 6	m_{vs}	volume mass	pg 22
ρ	density	pg 7	V	volume	pg 9
N	number of points	pg 9	θ	temperature	pg 19
ϵ	random thermal motion	pg 7	\mathbf{v}_l	linear velocity	pg 7
\mathbf{v}_w	angular velocity	pg 7	E	kinetic energy	pg 9
E_s	excitation energy	pg 9	E_0	activation energy	pg 9
T	total dissolution time	pg 19	S	inter-facial surface area	pg 19

Table 1: A list of symbols in this paper, with pages where they are first defined.

3.1 Chemical Collision Theory

Dissolution is a kinetic process, and in chemical engineering the dissolution of different solid substances is quantified by its dissolution rate [23]. In Computer Graphics, to mimic the dissolution process we need to consider two different aspects of this problem: which part of the solute has the highest probability of being dissolved first, and the speed of dissolution, or dissolution rate, represented by the mass transfer rate of the solute.

To solve these problems, we have to understand how chemical reactions take place and why chemical reaction rates differ [24], which are fortunately described by **Collision Theory**, firstly proposed by Trautz [25]. Collision theory is closely related to the kinetic-molecular theory: molecules are in constant motion, occasionally colliding as they move. For a chemical reaction to occur, molecules need to collide in the right orientation and with sufficient energy [26]. The relationship between energy and collision orientation is shown in Fig. 2. The energy required for the reaction to occur is referred to as the *activation energy* — necessary to break the existing chemical bonds between substrate molecules. If the energy produced

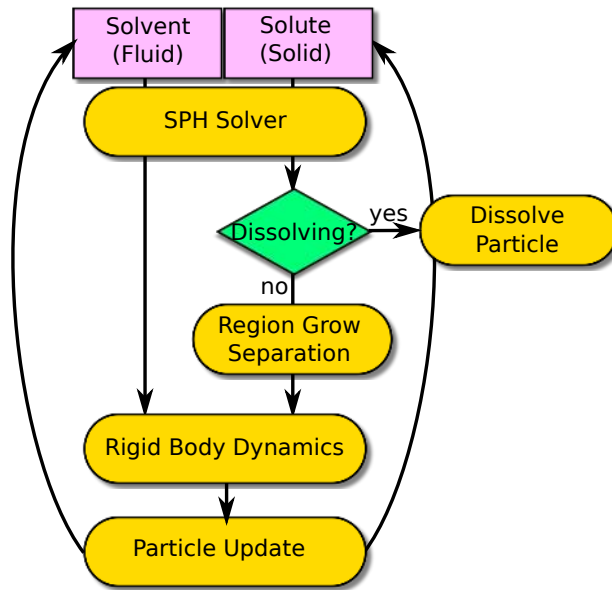


Figure 1: Framework of our dissolution model.

by the collision is less than the activation energy, the chemical reaction will not proceed.

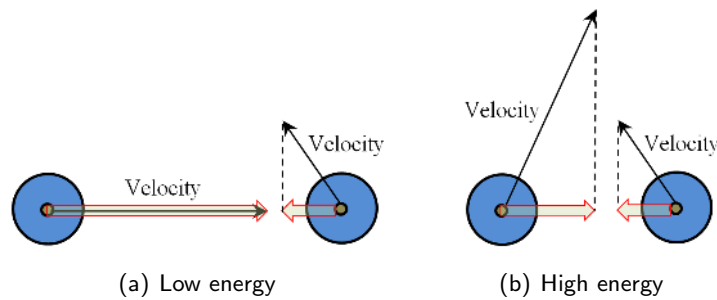


Figure 2: An intuitive depiction of the principle of particle Collision Theory.

Our dissolution model follows the Collision Theory maintaining the chemical plausibility of the simulation. We volumetrically sample both the solute and the solvent with particles, with each particle becoming a representative component of the corresponding substance. It is tempting to think of a particle as representing a molecule in collision theory, but in this case one particle might represent hundreds or thousands of molecules. The particle nature also makes the tracking of object separation possible.

We introduce a kinetic energy to describe the particle excitation of solute which governs the dissolving process (i.e. if the particle needs to be dissolved or not). When solute particles collide with the solvent particles, energy is accumulated with the solute particle. Each solute particle carries the energy information which is summed over all the collisions that have occurred. When the energy is larger than a threshold — the activation energy — the particle detaches from the object. This activation energy becomes an important factor in determining the dissolution rate. By detecting the relationship between the local particle excitation and the global dissolution time, we are able to control the dissolution rate during the process. The simulation continues until all the solid particles are dissolved into solvent. The position of the solute is updated using rigid body dynamics. This method allows us

to model the dissolution process in a physically and chemically correct manner.

3.2 SPH Interpolation

Most fluid simulation methods are governed by the Navier–Stokes equations [27]. From the viewpoints of the Lagrangian (particle–based) and Eulerian (grid–based) methods, particle–based methods seem to be a natural choice for dissolution because it is easy to implement with arbitrary boundaries and applied forces, while supporting fluid mixing and interaction with arbitrary rigid bodies. Smoothed Particle Hydrodynamics (SPH) was first introduced as a mesh–free Lagrangian method independently by both Gingold and Monaghan [28] and Lucy [29], and has since been established as one of the principle methods for fluid animation in computer graphics.

We use the standard SPH [30] for fluid simulation to demonstrate the dissolution framework, since the focus of this paper is on the chemical interaction between fluids and solids rather than fluid itself, but our method is compatible with other SPH models [31], such as PCISPH [16], and can be extended with other particle–based representations, such as Position Based Dynamics [32] or the Material Point Method [17].

We demonstrate the basic SPH method with a short introduction which is required to understand our dissolution method. In SPH, a scalar quantity A at location \mathbf{x} is computed by

$$A(\mathbf{x}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{x} - \mathbf{x}_j, h), \quad (1)$$

where $W(\mathbf{x}, h)$ is a smoothing kernel with smooth core radius h which also decides the neighbourhood size. The kernel satisfies $\int W(\mathbf{x}, h) d\mathbf{x} = 1$ and $W(\mathbf{x}, h) = 0$ when $\|\mathbf{x}\| > h$. Particle j is the neighbour of the particle at \mathbf{x} . m_j is the particle mass and A_j is the field quantity at \mathbf{x}_j .

Each individual fluid particle carries its own attribute values such as density and pressure, which are deduced from the neighbourhood of the fluid and solid particles. These values are calculated based on [30] and then used to calculate the physical motion and chemical behaviour happened to fluid. Neighbour searching of the solvent simulation is efficiently implemented using the framework of Hoetzlein [33].

To ensure the correctness of fluid–solid boundary interaction without causing any penetration or sticking artefacts, we use the corrected density computation of [15] to handle the boundary problem, except we sample the entire object rather than just the surface since the boundary and shape change as the solute dissolves.

3.3 Kinetic Energy

Based on Sec. 3.1, local collision energy of solute particle measures the particle excitation. Since the collision itself is a kinetic process, we derive our particle excitation equation from the kinetic energy $\frac{1}{2}m\mathbf{v}_r^2$, where \mathbf{v}_r is the *relative velocity*. It is incorporated into Eq. 1 to derive the expression for the particle excitation. Thanks to the use of relative velocity, the formulation yields asymmetric rules.

$$\begin{aligned} E_s &= \sum_f \frac{m_f}{\rho_f} E_f W(\mathbf{x}_s - \mathbf{x}_f, h), \\ &= \sum_f \frac{m_f}{\rho_f} \left(\frac{1}{2} m \mathbf{v}_r^2 \right) W(\mathbf{x}_s - \mathbf{x}_f, h), \\ &= \frac{1}{2} m \sum_f \frac{m_f}{\rho_f} [\|\mathbf{v}_l\| + \|\mathbf{v}_w\| + \epsilon]^2 W(\mathbf{x}_s - \mathbf{x}_f, h) \end{aligned} \quad (2)$$

where m is the volume mass of solute particle (introduced later in Eq. 4); f is the solvent particle within the neighbourhood of solute particle s ; m_f and ρ_f are the mass and density of solvent particle; $W(\mathbf{x}_s - \mathbf{x}_f, h)$ is a poly6 kernel [30], which smooths out the energy according to distance; other kernels may apply, such as cubic spline [31]. The relative velocity \mathbf{v}_r is split into three items \mathbf{v}_l , \mathbf{v}_w and ϵ .

\mathbf{v}_l is the relative linear velocity of solid and fluid particles. According to the Collision Theory, the dissolution only occurs when particles collide with a direct angle and at sufficient speed. In SPH, however, particles cannot collide as they will be repelled by the pressure between particles. Instead, we consider collision to have occurred once one SPH particle has entered the neighbourhood of the other, as they are representative thousands of molecules of solvent and solute. Particles outside of the neighbourhood are not computed. The magnitude of the velocity is affected by the angle of the collision, calculated as:

$$\|\mathbf{v}_l\| = \max((\mathbf{v}_f - \mathbf{v}_s) \cdot (\mathbf{x}_s - \mathbf{x}_f), 0), \quad (3)$$

where $\mathbf{v}_{s,f}$ represent the velocity of solid and fluid particles respectively. The direction of \mathbf{v}_l is $\mathbf{x}_s - \mathbf{x}_f$. This is illustrated in Fig. 3, where only f_2 and f_3 contribute to the collision.

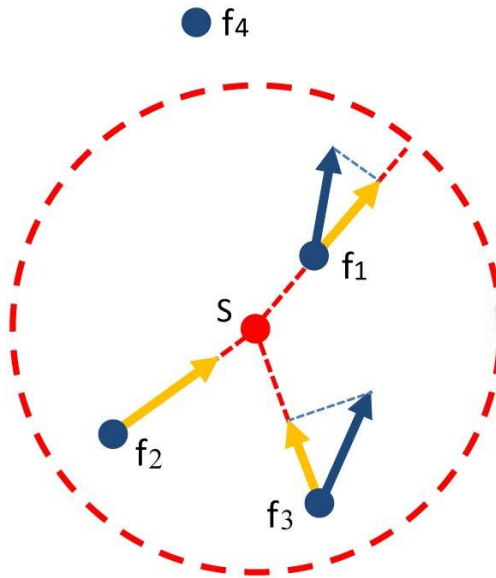


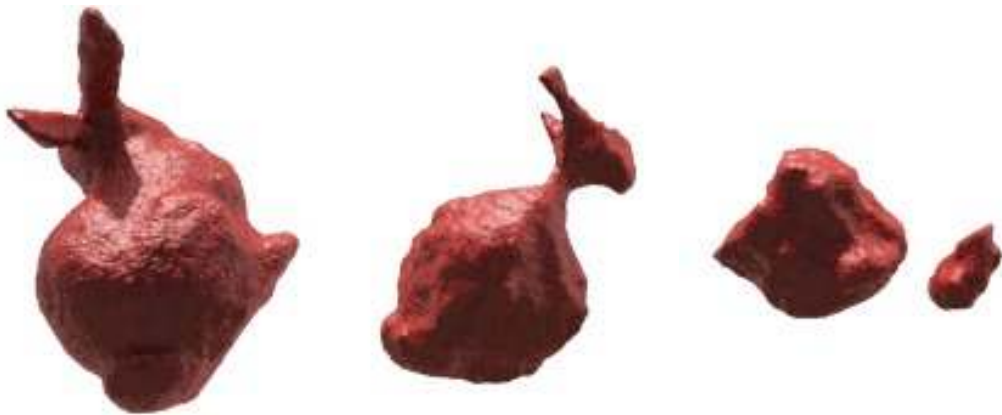
Figure 3: The collision domain. s represents a solid particle and $f_{1,2,3,4}$ are fluid particles, where f_4 is outside the neighborhood. The blue arrows indicate the actual relative velocity, while yellow ones are the calculated linear relative velocity \mathbf{v}_l . [9].

\mathbf{v}_w represents the angular velocity which contributes to the kinetic energy. \mathbf{v}_w is calculated by $\mathbf{w} \times \mathbf{r}$, where \mathbf{w} is the angular velocity and \mathbf{r} is the vector from the particle position to the centre of the object. This is only needed when there is obvious relative rotational motion between the object and fluid. Fig. 4 shows the different dissolution behaviours of the spinning bunny with and without angular velocity. It is easily noticed that the bunny with angular velocity is dissolving faster than the one without.

With the basic model described above no dissolution would occur if the solvent was at rest but in contact with the solute, for example if a puddle formed on the surface of the object. A small ϵ is added to demonstrate the random thermal motion among molecules which gives the possibility of dissolution when there is no macroscopic relative motion between the fluid and solid. In principle, this parameter can be defined according to the solubility of the solute in the solvent, but for simplicity $\epsilon = 0.00001$



(a) Dissolving without v_w



(b) Dissolving with v_w

Figure 4: A spinning bunny dissolving in water. Note that (b) demonstrates separation behaviour (will discuss in Sec. 6.1). The fluid is not rendered in these examples for better visualization of the solute. [9].

in our model. Note that even small values of ϵ will affect dissolution behaviour as E_s accumulates by time: when it is larger than a user specified activation energy E_0 , the particle dissolves into the solvent.

4 Sampling Resolution Independent Dissolution

Our method ensures the consistent dissolution result (within error tolerance) even when the sampling resolutions of the solute are different. This property has important applications in computer graphics: for the purposes of pre-visualisation, a solute with low sampling resolution or an adaptively sampled solute can be used to predict the dissolution behaviour of a very densely sampled solute, which improves the performance of real-time applications.

In previous methods [19, 20], shape alteration of a dissolving solute was achieved using mass transfer. Both methods used level-sets to represent the solute, causing the dissolution results to be largely affected by the grid resolution. We introduce the concept of **volume mass** which takes the volume and density of particle into consideration. It removes the assumption of the same mass for each particle, and calculates the mass of each particle according to its own representative volume and density. We use volume mass instead of mass in the energy calculation for each solute particle to ensure that our result is independent with the sampling resolution. The volume mass of each solid particle is calculated as:

$$m_{vs} = v_s \cdot \rho_s = \frac{V}{N} \sum_b m_s W(\mathbf{x}_s - \mathbf{x}_b, h), \quad (4)$$

where v_s and ρ_s are respectively the volume and the representative density of each solid particle. V is the volume of the entire object and m_s is the mass of each particle: both of which are constant once the object is initialised. N is the number of particles used to sample the object. It is easy to tell that $\rho_s = \sum_b m_s W(\mathbf{x}_s - \mathbf{x}_b, h)$ originates from the basic SPH density calculation in Eq. 1. b is the object particle within the solute neighbourhood h . Once the solute is sampled, the volume mass of each solute particle is set which does not change during the interactions. The volume mass of the boundary particles is set to the volume mass of the nearest interior particle to guarantee the consistency.

Fig. 5 demonstrates the property of sampling resolution independence by using different sampling resolutions, as well as non-homogeneous sampling resolutions, and compare these against previous dissolution methods [19, 20]. If we directly use the same mass for each solute, each collision will result in roughly the same energy, and the solute particle at overly sampled regions will have the same possibility to be dissolved with the solute particle at sparse sampled regions, causing discontinuities like Fig. 5(d). Instead, with our volume mass, as in Fig. 5(c), the solute particles in highly sampled area have larger volume mass (due to larger density) which means higher energy with each collision, thus will be easier to be dissolved according to Eq. 2.

To further demonstrate independence of sampling resolution, in Fig. 6 we pour liquid over a sphere, demonstrating the effect of pouring fluid at a targeted area. One of the spheres is sampled with 8538 particles while the other with 22274 particles. We calculate the percentage of the volume difference at each frame of these two spheres: with our method, the difference reaches a maximum of 3.4% of the entire sphere volume, while with the method of [20] and [19] the low sampled sphere dissolves completely before the high sampled sphere halves in size. To ensure that the fluid realistically adheres to the sides of the sphere, we use the surface tension model of [7] in this example.

Fig. 6(a) runs at 148 frames per second (FPS) in average while Fig. 6(b) runs at 48 FPS, both with 100k solvent particles. The performance is significantly improved with less solute particles, which demonstrates that our sampling independent dissolution model allows animators to preview the dissolution behaviour with a lower computational cost and is therefore useful for pre-visualization.

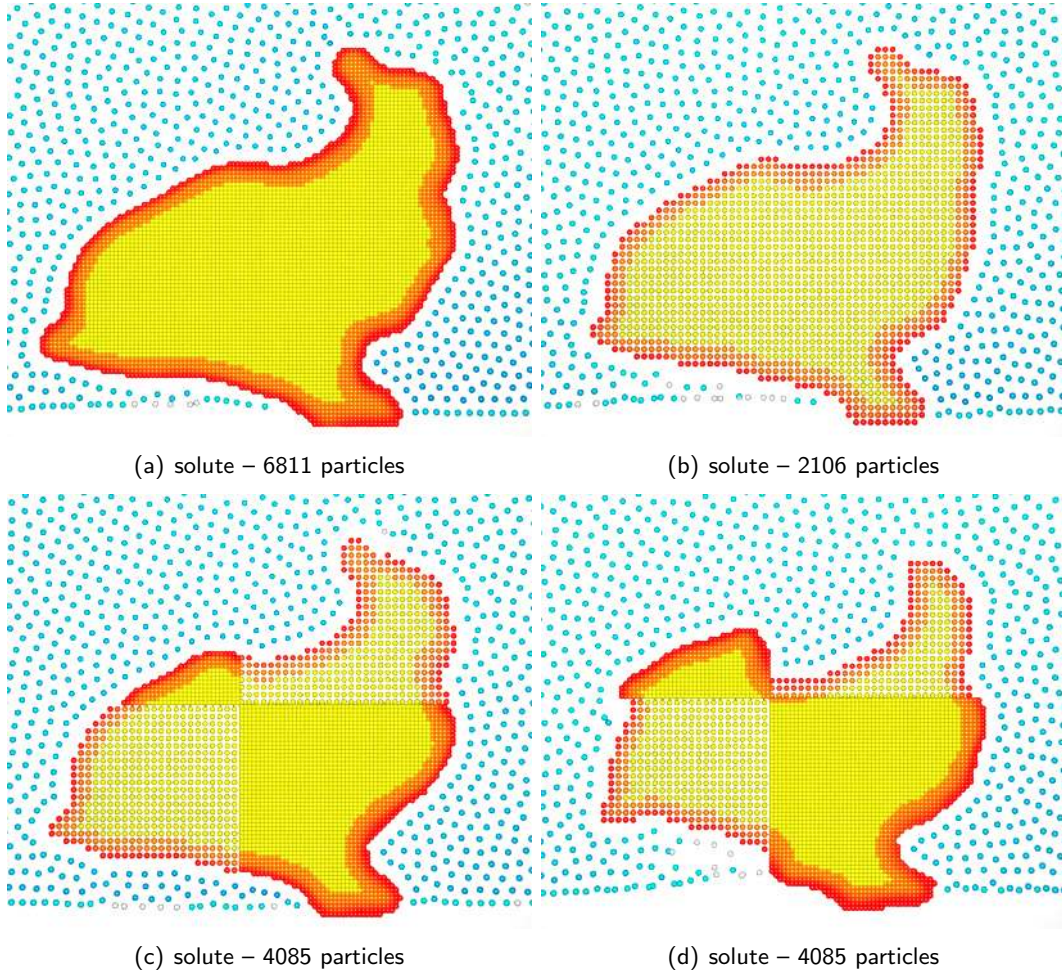


Figure 5: Bunny shaped solutes which are sampled with different resolutions dissolve in fluid. Using our method (in (a), (b) and (c)) the sampling resolution does not affect the dissolution process, even when the sampling is non-homogeneous. In (d) we demonstrate the result when using previous methods [19, 20]. All the images are taken from the same frame with the same initial conditions. The solute particles with high energy are coloured in red. [9].



(a) Sphere sampled with 8538 particles



(b) Sphere sampled with 22274 particles

Figure 6: Fluid is poured over spheres with different sampling resolutions, with almost identical results, except that the solute appears smoother with a higher sampling resolution [9].

5 Solute Sampling

Though our dissolution model is independent of the sampling resolution of the solutes, the distribution of the solute particles still affects the plausibility of the dissolution simulation. In the following section we investigate experimentally the influence of the sampling of the solute on the quality of the resulting simulation.

5.1 Random, Grid and Blue Noise Solute Sampling

During dissolution, solid particles will gradually transfer into the solution. Boundary particles are implicitly the first to be affected as they interact more frequently with the solvent, exposing inner particles that become the new boundary particles.

A completely random sampling of the solute offers no guarantees of the density of the distribution of boundary particles, which can cause fluid particles to compromise and penetrate the solute boundary. It follows that a homogeneous solute favours evenly distributed samples with equal density. We will also demonstrate that this improves the smoothness of the solute shape change, benefits the the control of the dissolving process and ensures that our dissolution process is stable and predictable.

While simple grid sampling of solute distribution plausibly mimics the mass transfer of the solute, it introduces serrated artefacts on the solute surface, affecting the rendered results in high resolution, shown in Fig. 7. It also causes popping artefacts due to the lack of randomness during the particle phase transition, shown in Fig. 18.

A blue noise distribution of solute particles will generate points that are evenly yet stochastic dispersed. In this paper we use SPH sampling method with a Capacity Constrained Voronoi Tessellation (CCVT) profile [8] for solute particle distribution. This approach avoids boundary penetration thanks to the even density distribution, eliminates serrated artefacts arising from regular grid sampling, and also improves the rendering appearance, as shown in Fig. 7.

The comparison between SPH sampling and simple grid sampling can be demonstrated using Fourier frequency transform, shown in Fig. 8. Grid sampling has a considerable high oscillation which means high structured aliasing due to the excessive regularity. The anisotropy of grid sampling is also significantly high while the SPH sampling is almost perfectly isotropic.

The importance of the isotropic property can be easily explained using an example: a fluid particle moves along the boundary of a solute (Fig. 9), and the density contribution from the solute particles within the neighbourhood of the fluid particle is calculated. With grid sampling, the density is oscillating while the particle is rounding the spherical solute, shown as the red curve in the right image of Fig. 9. In contrast, while the fluid particle is rounding the SPH sampled solute, the density difference varies gently, shown as the blue curve in the right image of Fig. 9. The difference enlarges when the smooth radius is smaller. The error of density estimation will increase the inaccuracy of the energy calculation during dissolution, leading to discontinuous behaviour. Therefore, an isotropic sampling method is required which tightly fits a given solute boundary. SPH sampling maintains the density stability even when the smoothing length is reasonably small which allows us to improve simulation performance by stabilising the density calculation.

Fig. 7 compares three different sampling methods — random, grid and SPH sampling for solute particle distribution in dissolution simulation. It is easily noticed that random sampling results in a faster dissolution process because the irregular sampling cannot guarantee the boundary condition, leading to more particles colliding with the boundary particles. Thanks to our sampling independent dissolution model, the simple grid sampling and SPH sampling roughly have the same dissolution rate in this experiment.

It should be noted that the SPH sampling method produces rougher surfaces during the dissolution

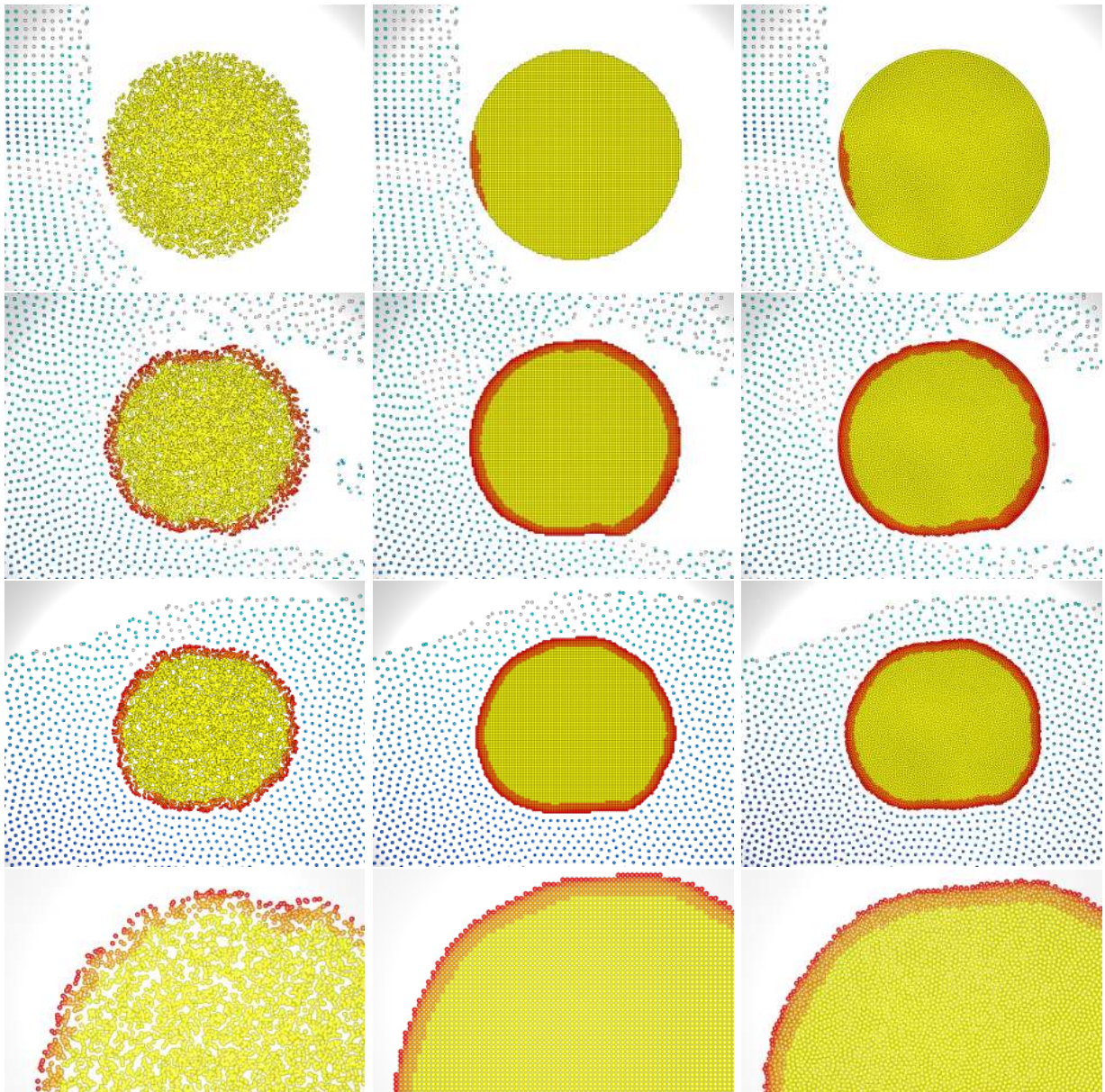


Figure 7: Comparison of different sampling methods for a solute in dissolution simulation at frame 100, 1000 and 10000 (top 3 rows). The bottom row shows a zoom in to the solute sampling. The left images are random sampling; the middle images are simple grid sampling; and the right images are the SPH sampling with CCVT profile. The solute spheres are all initially sampled with 8000 particles.

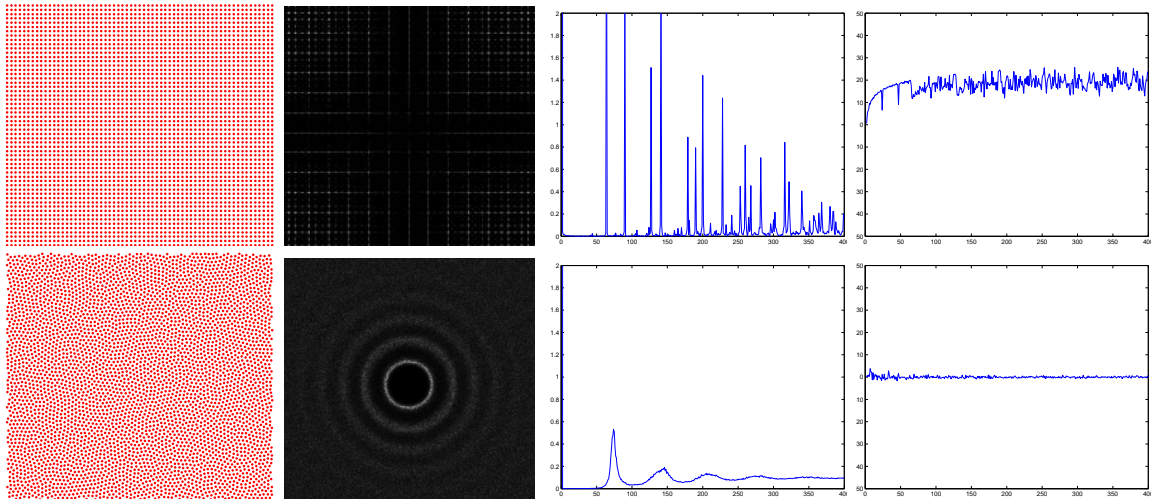


Figure 8: Fourier spectrum analysis of uniform grid sampling (top row) and SPH sampling (bottom row). The images from left to right are respectively sample points, spectrum, radial mean and anisotropy.

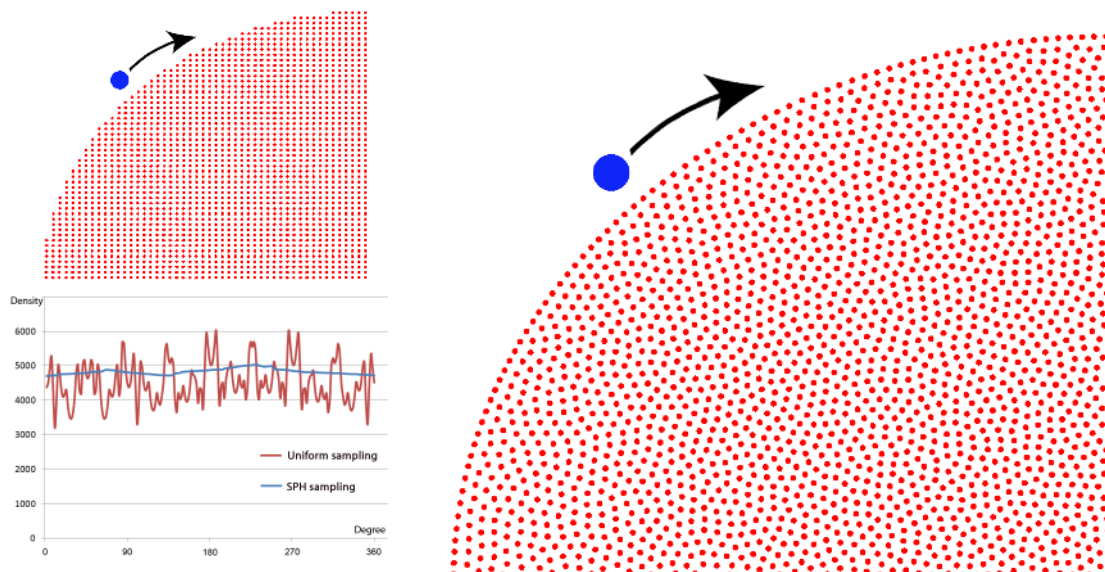


Figure 9: A fluid particle (blue) moves along the boundary of a solid solute (red). The graph shows the density changes while the fluid particle is moving along the spherical solute using uniform sampling and SPH sampling.

process — this mimics the behaviour in nature as molecular collision is a random process (as described in Sec. 3.1). This roughness is considerably more noticeable when rendered at high resolution.

5.2 A Comparison of Blue Noise Solute Sampling

We have experimented with alternative isotropic sampling techniques. Compared with the Poisson sampling, used in [6], SPH sampling minimizes the internal pressure of the solid solute and achieves a more even distribution (Fig. 10). [6] sampled the surface and enclosed volume separately using Poisson disk patterns similar to [34]. In contrast, the SPH sampling method in [8] samples both the surface and the volume simultaneously and in parallel, reducing computational cost.

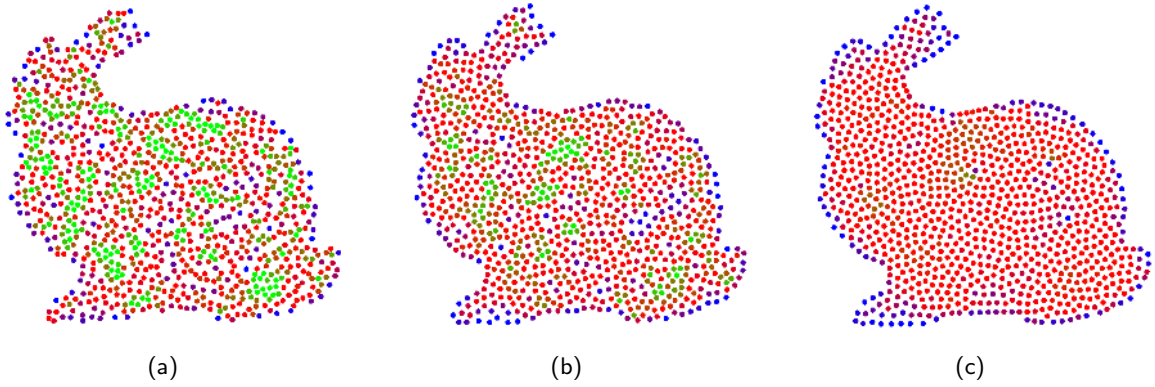


Figure 10: A comparison of solute sampling methods using (a) Poisson disk sampling, (b) Poisson disk sampling with relaxation [6] and (c) SPH sampling. 30 iterations for all the methods. The colour represents the density of each particle varying from green (high density) to blue (low density).

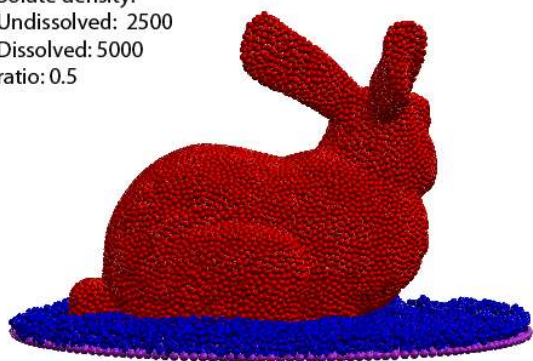
Compared with other blue noise sampling methods, the use of SPH sampling in our SPH dissolution model gives precise density control of the undissolved and dissolved solute particles. Take ice melting to water as an example: we can control the density of the undissolved solute by setting a corresponding sampling number to produce ice density ($0.9340g/cm$); and we can choose the density for the dissolved solute to match water density ($1,000kg/m$). Fig. 11 displays comparison examples of dissolving bunny at the same frame with different solute densities before or after dissolving. Users can choose the specific density ratio to control the volume change of the solute before and after dissolving, thus achieving desired visual effects.

5.3 Particle Distribution within a Boundary

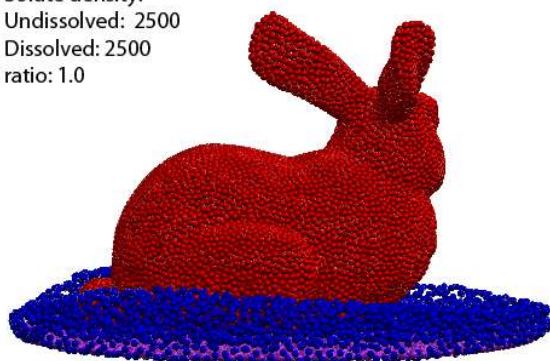
To ensure the particles are sampled within the solute object, Signed Distance Fields (SDF) [35] are used. SDF calculates the distance from a given point to the closet point on the object surface within a metric field [36] and the sign is used to indicate whether a particle is inside or outside the object. The particles inside the object will be kept and particles outside will be rejected until the desired sampling number of particles has been reached. We compute the sign using Angle Weighted Pseudo Normals [37], which requires the closest feature. There are numerous methods to find the closest feature on a mesh from a particle [36]. In this paper, we use AABB trees [38] for acceleration.

Once the solute particle distribution is initialized, there is no relative movement between these particles. Therefore, the sampling of the solute can be achieved off-line before the simulation.

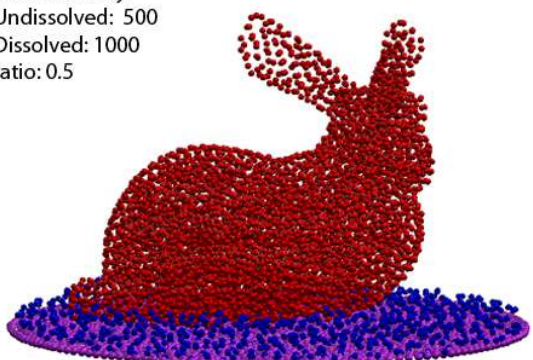
Solute density:
Undissolved: 2500
Dissolved: 5000
ratio: 0.5



Solute density:
Undissolved: 2500
Dissolved: 2500
ratio: 1.0



Solute density:
Undissolved: 500
Dissolved: 1000
ratio: 0.5



Solute density:
Undissolved: 500
Dissolved: 500
ratio: 1.0

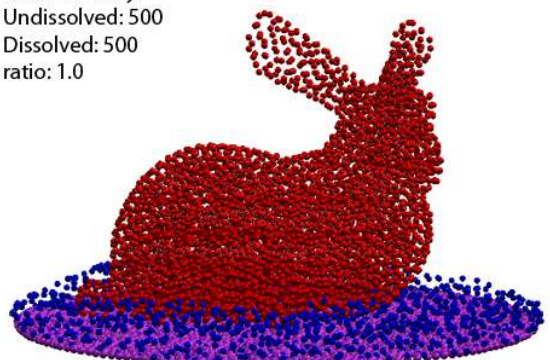


Figure 11: We demonstrate the user control of the volume change between the undissolved solute and dissolved solute by varying the density ratio between them.

6 Solute Update

During dissolution, parts of the object may break off and form separate shapes. Each disconnected component is treated as a rigid body. In this section the method to identify and simulate object separation is described.

6.1 Object Segmentation

In order to track the topology changes of the shape, we use the region growing algorithm to detect the object separation during dissolution, a method commonly used in image segmentation [39, 40]. We apply the technique to identify connected regions amongst the volumetric particles. Particles close enough are grouped into clusters, each representing a separate rigid body.

The most costly component of this algorithm is testing the proximity of particles, but as we already have neighbourhood information as part of the fluid simulation, this becomes trivial. We update the neighbourhood information before evaluating the region growing algorithm.

Sequential region growing quickly becomes the bottleneck in our implementation (time complexity $O(N)$) as the remainder of the algorithm is implemented in parallel. We propose a new parallel region growing algorithm on the GPU which integrates effectively into our dissolution framework. It is derived from [41] and improves the time complexity to an average of $O(\log(N))$.

Initially, all the particles are considered as seeds and are labelled with the particle index representing a class, shown in Fig. 12. Each parallel thread assigned to individual particle merges adjacent particles iteratively to minimize the number of distinct classes. To avoid race conditions, every time-step we update the particle's class as the *lowest* label of the neighbouring particles (including itself). The algorithm stops when there is no change in the classes (labels).

In our implementation, our parallel region growing method avoids the data transfer between CPU and GPU during the separation check every frame. Compared with CPU region growing, the performance on the GPU improves with the number of the separate parts. The best case scenario is if each particle represents an isolated object, in which case the time complexity reaches $O(1)$. The performance is also improved if each separated part consists of fewer particles. The overall computational cost is constrained by the largest separated part.

We compare the performance of the sequential and parallel region growing methods, implemented on the CPU and GPU respectively, using a simple test in which there are two objects, each with 70000 particles. Given the same neighbouring information, sequential region growing costs 31ms to converge while parallel region growing takes 15ms, demonstrating a 2X speed-up. In a test with four separate parts, each with 35000 particles, our parallel algorithm maintains the speed with 16ms while the sequential implementation takes 53ms. Further discussion and performance improvement of the algorithm is an interesting area of future research.

6.2 Rigid Body Dynamics

As each object represented by particles is assumed not to be deforming, no relative motion is expected among the particles in one rigid body. To maintain the rigidity of the solute, it is helpful to think that all the forces are exerted on the mass centre of the rigid bodies, which simplifies the motion of solute particles. For the acceleration we consider gravity, forces applied by the fluid, and the supportive force when the object hits the boundary. The forces from fluid such as pressure are equal to the opposite of the corresponding forces from the object to fluid. We assume that the viscosity is not large enough to affect the movement of the solids. The rules updating the solid obey the standard rigid body dynamics [42].

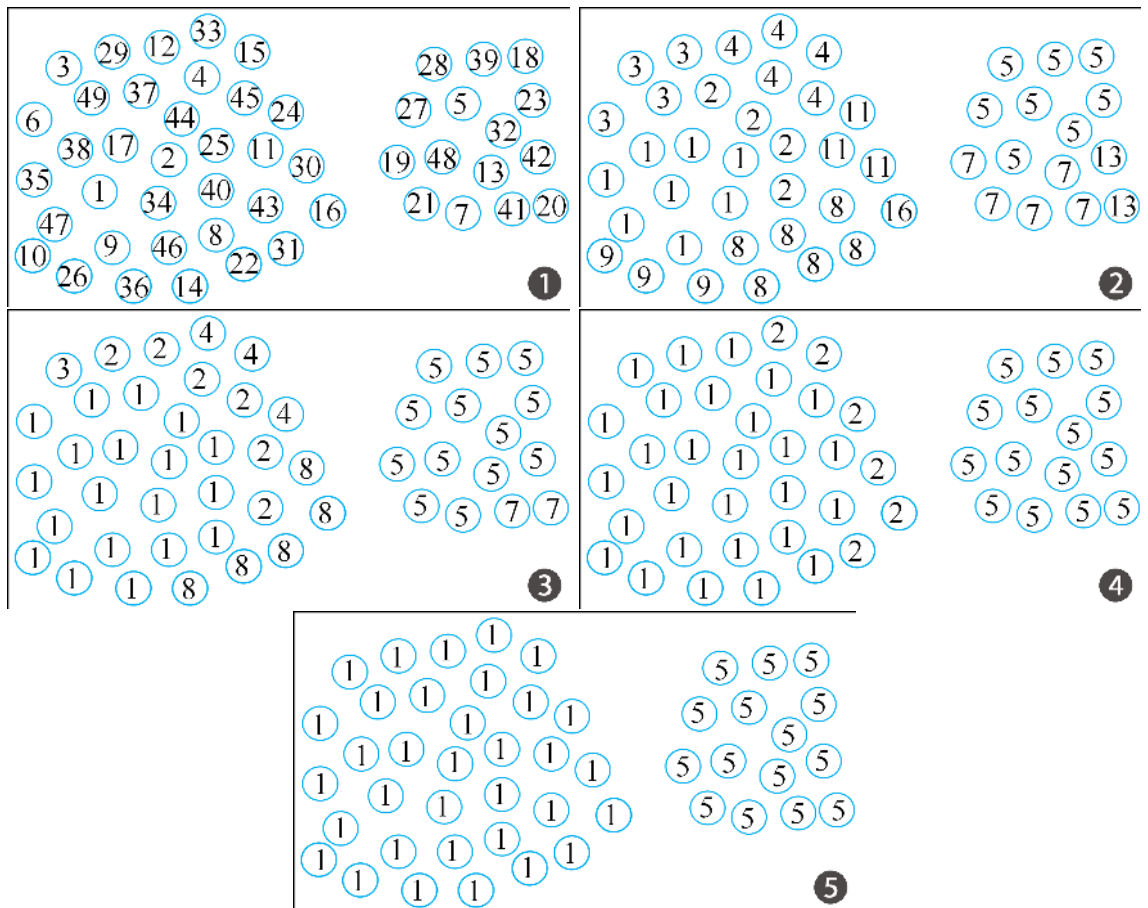


Figure 12: Parallel region growing process. Each particle is initially labelled with one class. Afterwards, they gradually merge class with their neighbours and are assigned the lowest label value. The results show that there are two isolated objects with label 1 and 5.

7 Activation Energy

Based on collision theory, the dissolution rate of individual particle can be controlled by the activation energy. However, this offers no control over the time taken for the object to be completely or partially dissolved: a property that is particularly important to animators seeking to control the simulation without having to resort to trial and error.

We provide the bridge to connect the global dissolution time with the local particle excitation. Dissolution depends on the nature of the solvent and solute, the temperature of both, the inter-facial surface area between them, and the presence of mixing [24]. In our method the presence of mixing is implicitly solved by the particle excitation (e.g., Fig 4). For simplicity, we assume temperature to be a constant parameter θ (although this could be varied within more complex simulations).

We determine the relationship between the total dissolution time T , activation energy E_0 and the inter-facial surface area S based on our simulation experiments. Fig. 13 shows the results of dissolving objects with different inter-facial surface areas and different activation energies. The three objects (sphere, cube, spiky cube) have the same volume of unit 1. The surface areas are respectively 4.83, 6.0, and 8.20.

Each curve describes the dissolution times of certain object with different activation energies. The shape of the curve can be best approximated with a general exponential function ($\alpha = 0.25$ in our examples): $T \propto \exp(\alpha E_0)$.

By observing the dissolution time of the different objects with the same activation energy, the relationship between dissolution time T and surface area S matches: $T \propto \frac{1}{S}$. This agrees with common sense: a larger surface area results in shorter dissolution time.

Thus the relationship between the total dissolution time, activation energy and the inter-facial surface area can be described as:

$$T = \frac{\exp(\alpha E_0)}{\theta S}, \quad (5)$$

Using this experimental methodology, artists are able to better estimate the total dissolution time for objects based on the surface area and the activation energy. An interactive tool could be developed which uses preceding simulation results of a particular fluid configuration to calculate the coefficients for better estimation. Once the coefficients are determined, the inverse of this function can be used to determine the activation energy for an artist specified target dissolution time. More complicated dissolution processes, such as a period of fast dissolution followed by a period of slow dissolution is also achievable by tuning the activation energy at the user specified time frame.

8 Applications and Results

Our simulations are implemented in CUDA. Real-time results are rendered in OpenGL and simulation results are rendered using *Houdini*. These results can be viewed in the accompanying video.

8.1 Solute Dissolves in Fluid

Antacid pills in water are a classic example of dissolution behaviour. Fig. 14 demonstrates the dissolution effect of two antacid pills dropped into a glass of water. Bubbles caused by the release of trapped air and chemical processes are recognisable clues that dissolution is taking place. We add bubbles at the dissolution site to increase the authenticity of the simulation. The bubble motion is modelled using basic Brownian motion [18].

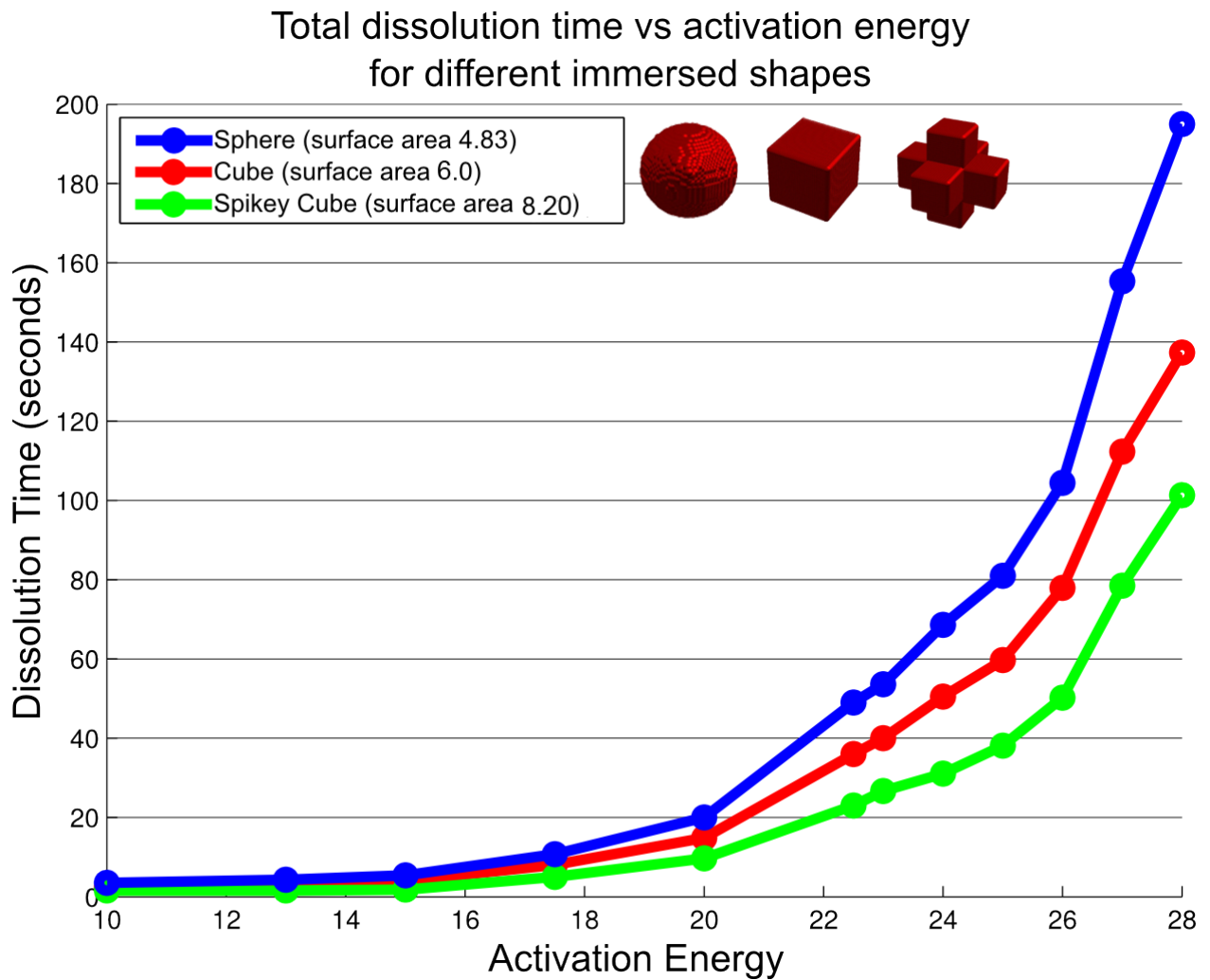


Figure 13: The relationship between the total dissolution time and activation energy for different solute shapes. Shapes with more surface area in contact with the solvent naturally dissolve faster. The observed dissolution rates lead us to use the exponential function to model the total dissolution time as a function of activation energy [9].



Figure 14: Two antacid pills dissolving in water. The bubbles are generated at dissolution sites and animated using basic Brownian motion. The buoyancy of the objects is not considered in the examples [9].

8.2 Erosion

Our dissolution method can also be applied to simulate hydraulic erosion phenomena. While we do not use the actual physical properties of different terrain types, these examples demonstrate the viability of using our model for practical simulations. Fig. 15 shows the hydraulic erosion acting on the floor of a river bed. As the fluid runs over the terrain, certain soil particles become dissolved in the water. Note that more complex properties of erosion, such as fluid carrying and relocating sediment, are ignored in this simulation.

One advantage of our energy-based model is that we can simulate different terrain types by simply modifying the activation energy of individual layers of particles. This allows us to simulate erosion of rocks consisting of different layers each with different material properties, giving rise to natural formations similar to those which can be found in mountainous or coastal regions.

This simulation is demonstrated in Fig. 16, where a three layered terrain is immersed in turbulent fluid to simulate coastal erosion. Different activation energies are assigned to particles in each layer, with the lowest activation energy assigned to the middle layer — leading to a natural arch formation.

8.3 Melting

Our dissolution model can be easily transformed to mimic a simple melting effect. Fig. 17 shows a bunny melting with different temperatures. The model is simplified by only considering the heat transfer from the circular pan to the bunny and ignoring the heat transfer elsewhere. We transformed the energy calculation of Eq. 2 by replacing the kinetic energy with temperature θ as below:

$$E_s = m_{vs} \sum_h \theta W(\mathbf{x}_s - \mathbf{x}_h, h), \quad (6)$$

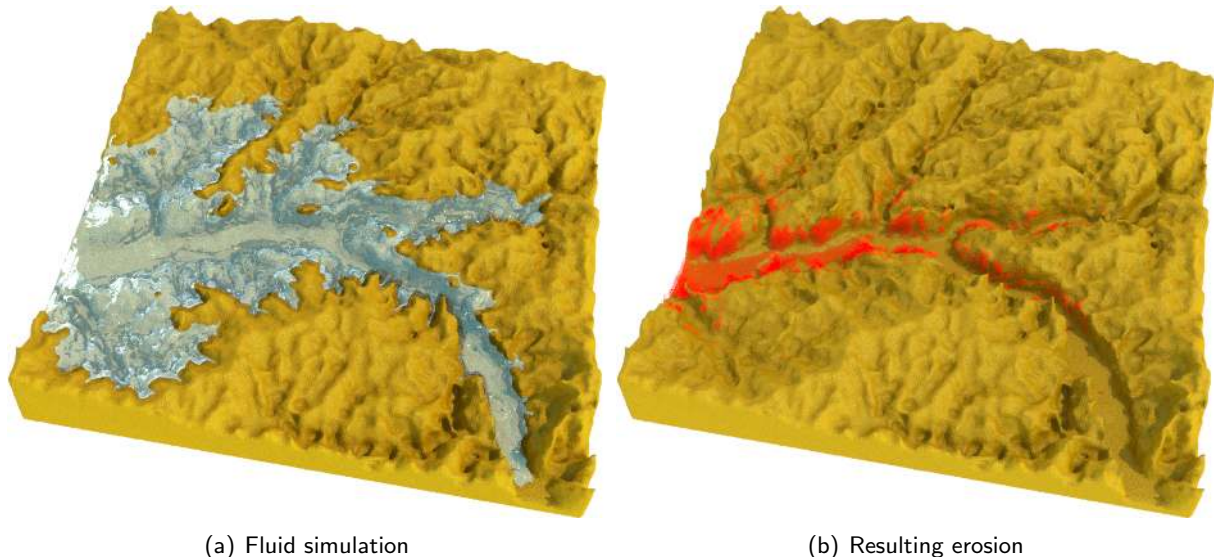


Figure 15: Hydraulic erosion using our dissolution model. In (a) we show the dissolution process, while in (b) the removed soil particles are shown in red [9].

where m_{vs} is the volume mass, \mathbf{x}_s and \mathbf{x}_h are the position of the solute particle and the heated particle respectively, which are the particles representing the pan. When the heat energy of the solid particle exceeds a user defined threshold, the particle will detach from the solid object and become fluid particle. The solute particles are distributed using SPH sampling.

We also test this example using simple grid sampling. Fig. 18 shows a bunny with grid sampling melting in a heated pan. Since one layer of particles will have the same energy (no randomness), they will gain the same heat and melt at the same time, leading to obvious popping artefacts during the melting process as all the particles in each layer melts simultaneously.

9 Conclusion and Future work

In this paper we introduce a novel energy-based dissolution model based on the Collision Theory. Both solute and solvent are simulated using Smoothed Particle Hydrodynamics. We demonstrate by way of examples that this method achieves physically and chemically plausible dissolution, erosion and melting behaviours.

The key to this model is a new expression for particle excitation based on kinetic energy. By using the volume mass instead of mass to update local particle energy, we are able to ensure the consistent dissolution results independent of the sampling density, which is beneficial for pre-visualization.

We also establish a mathematical relationship between the activation energy and the total dissolution time, allowing animators to easily adjust the global properties of the dissolution simulation by modifying a single parameter.

The particle representation naturally lends itself to a parallel implementation. We introduce a parallel region growing method which integrates into our dissolution framework and greatly improves performance.

During fluid-solid interaction the representation of the solute becomes vital particularly when the solid shape is changing. The choice of the sampling methods not only affects the rendering results, but also influences the correctness of the simulation. By using SPH sampling for the solute particle

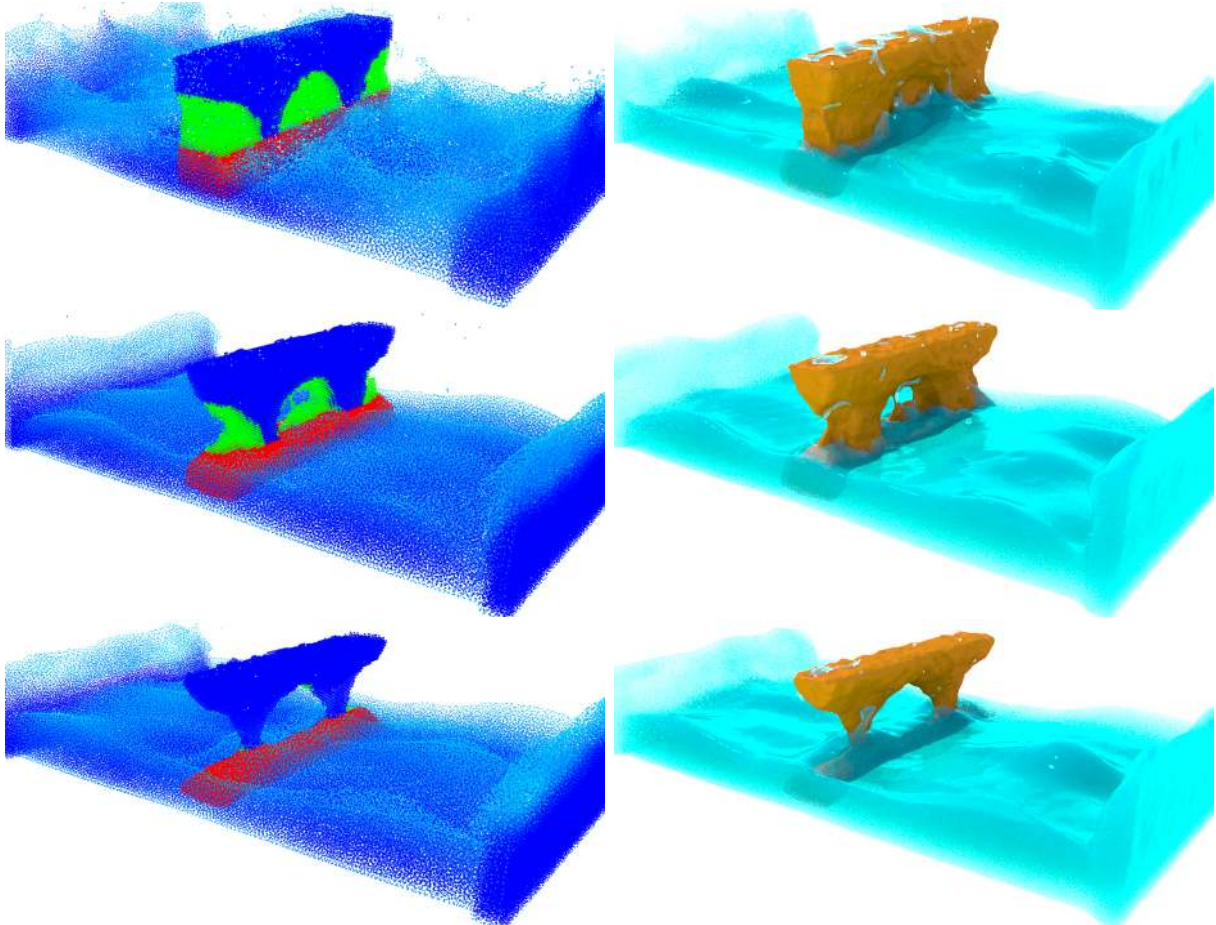


Figure 16: A simulation of layered terrain erosion using our dissolution model. The solute consists of three different material layers, each with a different activation energy. The green layer has the lowest activation energy which leading to a faster dissolving. Forces are added to the fluid to keep the fluid active. The rendered results are shown in the right corresponding to the particle representations [9].

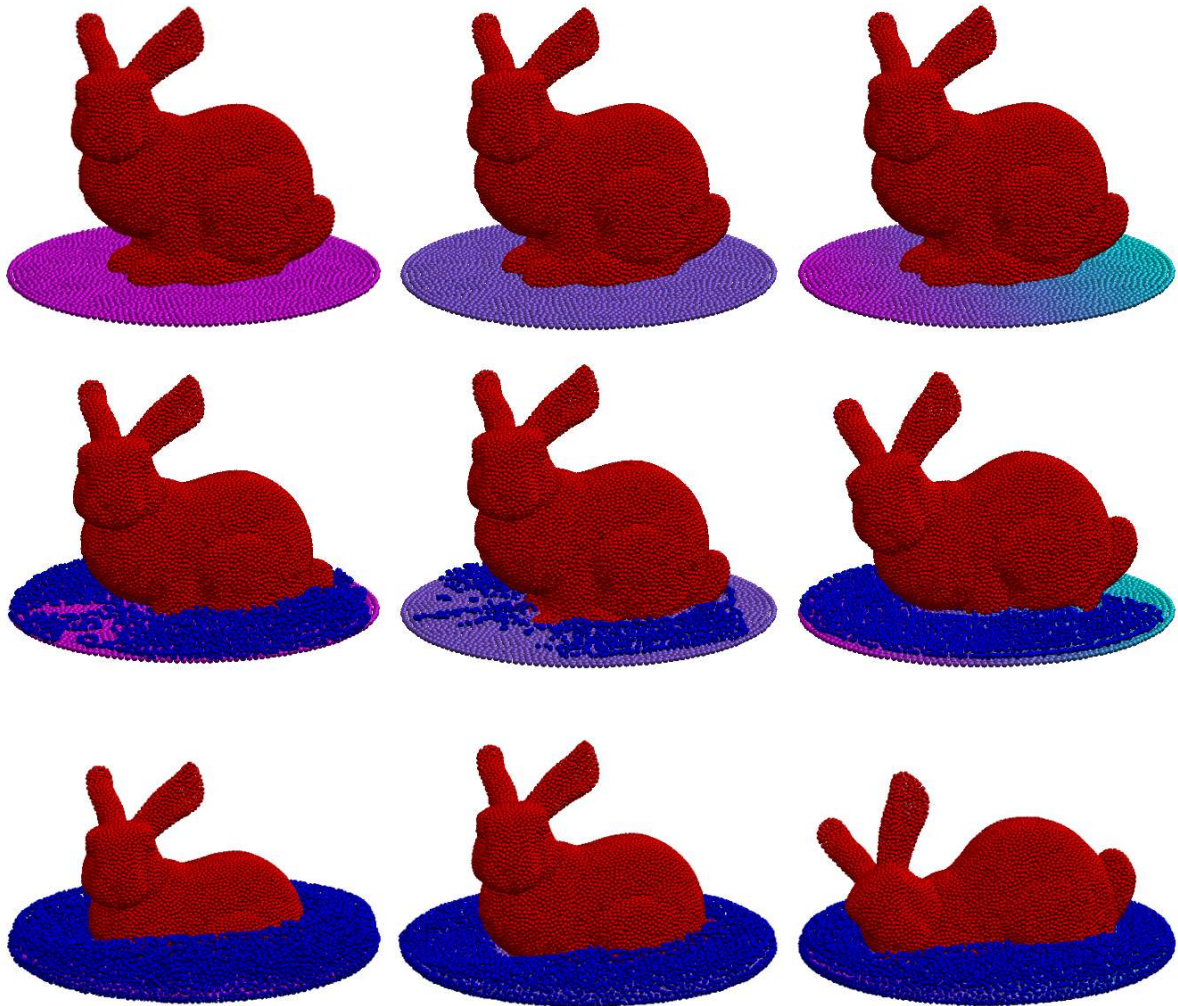


Figure 17: The left column shows a bunny melting in a high temperature pan, while the middle one melts in a low temperature pan, and the right one melts in a pan whose temperature varies from one side to the other. Purple particles have high temperature and green ones have low temperature.

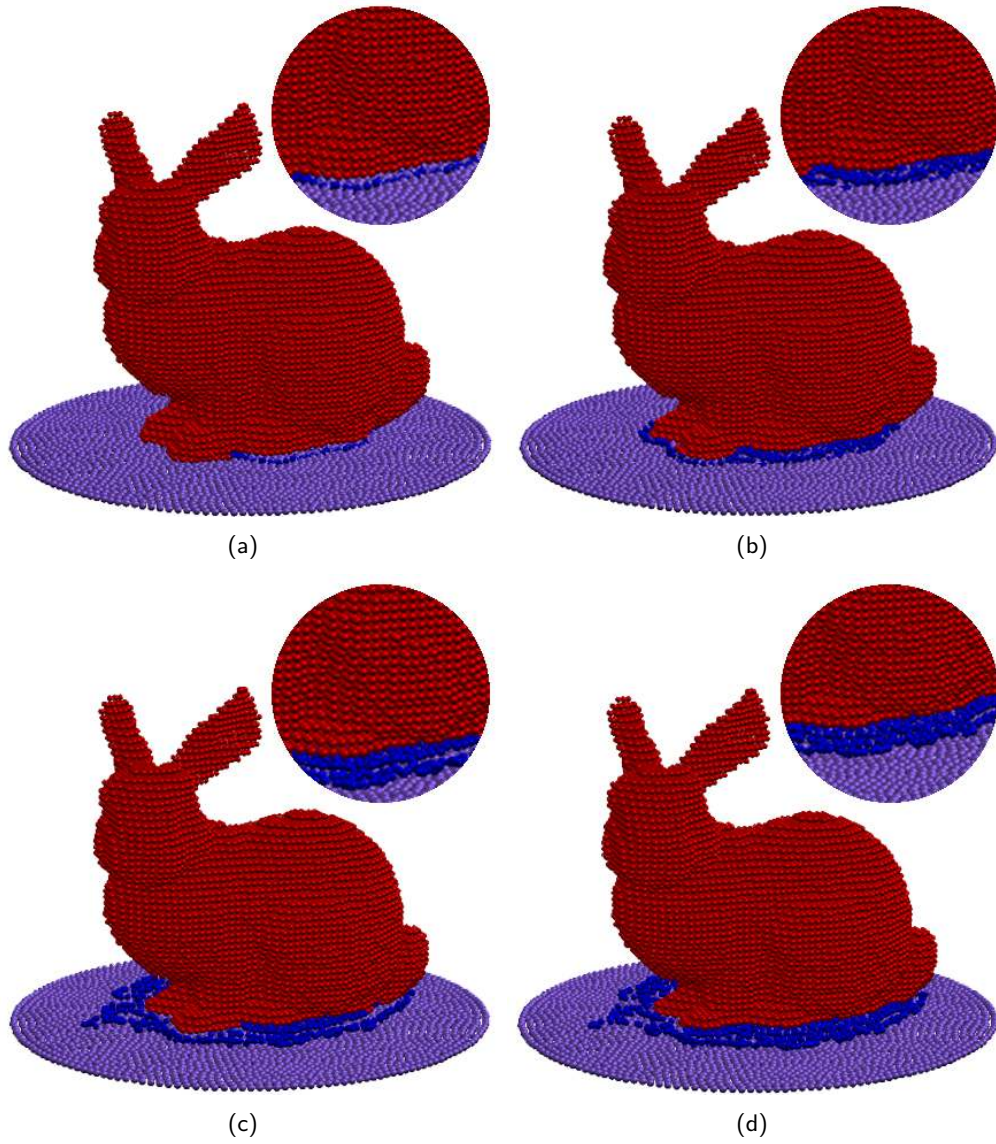


Figure 18: In (a), (b), (c) and (d) heat is applied to the base of the bunny causing it to melt over time. Due to the regularity of the grid sampling particles on each layer accumulate enough energy to melt at the same time. This causes the object melts layer by layer over a constant time interval instead of melting gradually and smoothly, resulting in obvious popping artefacts during simulation.

distribution we avoid artefacts and create better visual results on the surface compared with simple grid sampling. Compared with other blue noise sampling methods, SPH sampling allows users to control the volume change of the dissolved and undissolved solute and achieve the desired visual effects.

There are several directions for future work. The interaction of our particle-based model with a gas model for wet and dry foam could potentially enhance realism. More significantly, this and recent published works hint at the development of a unifying particle-based representation for interacting media — allowing gas, solids and fluids to be simulated within a single framework, and leading to the development and simulation of other complex natural phenomena such as evaporation and boiling.

References

- [1] Markus Ihmsen, Nadir Akinci, Gizem Akinci, and Matthias Teschner. Unified spray, foam and air bubbles for particle-based fluids. *Vis. Comput.*, 28(6-8):669–677, June 2012.
- [2] Paul W Cleary, Soon Hyoung Pyo, Mahesh Prakash, and Bon Ki Koo. Bubbling and frothing liquids. *ACM Transactions on Graphics (TOG)*, 26(3):97, 2007.
- [3] Oleksiy Busaryev, Tamal K. Dey, Huamin Wang, and Zhong Ren. Animating bubble interactions in a liquid foam. *ACM Trans. Graph.*, 31(4):63–71, July 2012.
- [4] Christopher Batty, Florence Bertails, and Robert Bridson. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph.*, 26(3), July 2007.
- [5] Avi Robinson-Mosher, Tamar Shinar, Jon Gretarsson, Jonathan Su, and Ronald Fedkiw. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Trans. Graph.*, 27(3):46–55, August 2008.
- [6] Hagit Schechter and Robert Bridson. Ghost SPH for animating water. *ACM Trans. Graph.*, 31(4):61–69, July 2012.
- [7] Nadir Akinci, Gizem Akinci, and Matthias Teschner. Versatile surface tension and adhesion for SPH fluids. *ACM Trans. Graph.*, 32(6):182–190, November 2013.
- [8] Min Jiang, Yahan Zhou, Rui Wang, Richard Southern, and Jian Jun Zhang. Blue noise sampling using an SPH-based method. *ACM Transactions on Graphics (TOG)*, 34(6):211, 2015.
- [9] Min Jiang, Richard Southern, and Jian Jun Zhang. A particle-based dissolution model using chemical collision energy. In *Proceedings of the 10th International Conference on Computer Graphics Theory and Applications (VISIGRAPP 2015)*, pages 285–293, 2015.
- [10] Olivier Génévaux, Arash Habibi, and Jean-Michel Dischler. Simulating fluid-solid interaction. In *Graphics Interface*, volume 2003, pages 31–38. Citeseer, 2003.
- [11] Mark Carlson, Peter J. Mucha, and Greg Turk. Rigid fluid: Animating the interplay between rigid bodies and fluid. *ACM Trans. Graph.*, 23(3):377–384, August 2004.
- [12] Takashi Amada. Real-time particle-based fluid simulation with rigid body interaction. In *GPU Gems 3*, volume 6, pages 189–205. Pearson Education, 2006.
- [13] Takahiro Harada, Masayuki Tanaka, Seiichi Koshizuka, and Yoichiro Kawaguchi. Real-time coupling of fluids and rigid bodies. *Proc. of the APCOM*, 2007.

- [14] Markus Becker, Hendrik Tessenorf, and Matthias Teschner. Direct forcing for lagrangian rigid-fluid coupling. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):493–503, May 2009.
- [15] Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. Versatile rigid-fluid coupling for incompressible SPH. *ACM Trans. Graph.*, 31(4):62–70, July 2012.
- [16] Barbara Solenthaler, Jürg Schläfli, and Renato Pajarola. A unified particle model for fluid–solid interactions: Research articles. *Comput. Animat. Virtual Worlds*, 18(1):69–82, Feb 2007.
- [17] Alexey Stomakhin, Craig Schroeder, Chenfanfu Jiang, Lawrence Chai, Joseph Teran, and Andrew Selle. Augmented MPM for phase-change and varied materials. *ACM Trans. Graph.*, 33(4):138:1–138:11, July 2014.
- [18] Doyub Kim, Oh-young Song, and Hyeong-Seok Ko. A practical simulation of dispersed bubble flow. *ACM Trans. Graph.*, 29(4):70:1–70:5, July 2010.
- [19] Seung-Ho Shin, Hyeong Ryeol Kam, and Chang-Hun Kim. Hybrid simulation of miscible mixing with viscous fingering. *Computer Graphics Forum*, pages 675–683, 2010.
- [20] Chris Wojtan, Mark Carlson, Peter J. Mucha, and Greg Turk. Animating corrosion and erosion. In *Proceedings of the Third Eurographics conference on Natural Phenomena*, NPH’07, pages 15–22, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [21] B. Beneš, V. Těšínský, J. Hornyš, and S. Bhatia. Hydraulic erosion. *Computer Animation and Virtual Worlds*, 17(2):99 – 108, 2006.
- [22] P. Kritof, B. Bene, J. Kivnek, and O. t’ava. Hydraulic erosion using smoothed particle hydrodynamics. *Computer Graphics Forum*, 28(2):219–228, 2009.
- [23] T.P. Kravtchenko, J. Renoir, A. Parker, and G. Brigand. A novel method for determining the dissolution kinetics of hydrocolloid powders. *Food Hydrocolloids*, 13(3):219 – 225, 1999.
- [24] Alan D. McNaught and Andrew Wilkinson. *Collision theory*. Oxford: Blackwell Scientific Publications, 1997.
- [25] Max Trautz. Das gesetz der reaktionsgeschwindigkeit und der gleichgewichte in gasen. bestätigung der additivität von $cv-3/2r$. neue bestimmung der integrationskonstanten und der moleküldurchmesser. *Zeitschrift für anorganische und allgemeine Chemie*, 96(1):1–28, 1916.
- [26] J. Clark. *The Essential Dictionary of Science*. New York: Barnes & Noble Book, 2004.
- [27] Nick Foster and Dimitri Metaxas. Realistic animation of liquids. *Graphical models and image processing*, 58(5):471–483, 1996.
- [28] Robert A Gingold and Joseph J Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977.
- [29] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astron. J.*, 82:1013–1024, December 1977.

- [30] Matthias Müller, David Charypar, and Markus 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.
- [31] Markus Ihmsen, Jens Orthmann, Barbara Solenthaler, Andreas Kolb, and Matthias Teschner. SPH Fluids in Computer Graphics. In Sylvain Lefebvre and Michela Spagnuolo, editors, *Eurographics 2014 - State of the Art Reports*. The Eurographics Association, 2014.
- [32] Miles Macklin and Matthias Müller. Position based fluids. *ACM Trans. Graph.*, 32(4):104–116, July 2013.
- [33] Rama C. Hoetzlein. Fast fixed-radius nearest neighbors: Interactive million-particle fluids. *GPU Technology Conference (GTC)*, 2014.
- [34] Robert Bridson. Fast Poisson disk sampling in arbitrary dimensions. In *ACM SIGGRAPH 2007 sketches*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.
- [35] Bradley A. Payne and Arthur W. Toga. Distance field manipulation of surface models. *IEEE Computer Graphics and Applications*, 12(1):65–71, 1992.
- [36] Mark W Jones, Jakob Andreas Bærentzen, and Milos Sramek. 3D distance fields: A survey of techniques and applications. *Visualization and Computer Graphics, IEEE Transactions on*, 12(4):581–599, 2006.
- [37] J. Andreas Baerentzen and Henrik Aanaes. Signed distance computation using the angle weighted pseudonormal. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):243–253, May 2005.
- [38] Pierre Alliez, Stephane Tayeb, and Camille Wormser. CGAL: 3D fast intersection and distance computation (AABB tree). http://doc.cgal.org/latest/AABB_tree/, 2015. Accessed: 2016-02-17.
- [39] SA Hojjatoleslami and Josef Kittler. Region growing: a new approach. *IEEE Transactions on Image processing*, 7(7):1079–1084, 1998.
- [40] Martin Baatz and Arno Schäpe. Multiresolution segmentation: an optimization approach for high quality multi-scale image segmentation. *Angewandte Geographische Informationsverarbeitung XII*, 58:12–23, 2000.
- [41] P. N. Happ, R. Q. Feitosa, C. Bentes, and R. Farias. A region-growing segmentation algorithm for GPUs. *IEEE Geoscience and Remote Sensing Letters*, 10(6):1612–1616, Nov 2013.
- [42] Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.