Graphical Models 94 (2017) 14–24

Contents lists available at ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

An efficient heat-based model for solid-liquid-gas phase transition and dynamic interaction

Yang Gao[a], Shuai Li[a,b,*], Lipeng Yang[c], Hong Qin[d], Aimin Hao[a]

[a] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, China
[b] Beihang University Qingdao Research Institute, China
[c] Amazon China Investment Corporation, China
[d] Department of Computer Science, Stony Brook University, USA

ARTICLE INFO

Article history:
Received 1 May 2017
Revised 17 August 2017
Accepted 1 September 2017
Available online 23 September 2017

Keywords:
FLIP
SPH
Multi-phase fluid
Phase transition and interaction
Simplified heat transfer model
Bubble simulation

ABSTRACT

For melting simulation, solid-liquid coupling, liquid-gas interaction, bubble/foam generation, etc., many new methods have been emerging in recent years in computer graphics. To further push advance the technical frontier of the aforementioned phenomena, our novel solution is to focus on an efficient heat-based method towards faithful simulation of physical procedures pertinent to phase transitions and their dynamic interactions. On the methodology aspect, this paper details a simplified temperature-based model to animate the phase transitions and their dynamic interactions, including melting, freezing, and vaporization, by integrating the latent heat model with relevant governing physical laws. On the numerical aspect, our framework supports a new algorithm aiming at tight coupling of heat transfer and multiphase FLIP-based fluids. Specifically for liquid-gas phase transition, we take into account the dissolved gas involved in liquid which further enhances the bubble generation effects. Besides the unique feature of heat transfer, we also devise a SPH-FLIP coupled model to simulate sub-grid bubbles, which enables three-phase dynamic interactions among solid, liquid, and gas. The extensive experiments show that our hybrid approach can simultaneously handle multi-phase transition driven by physics-based heat conditions, as well as the multi-phase dynamic interactions with high fidelity and visual appeal.

1. Introduction and motivation

The natural phenomena relevant to phase transition and the interactions among them are ubiquitous in our everyday life. Heat-based phenomena which involves several fundamental transition processes in physics, such as ice melting, water boiling and water freezing, are visually important in computer graphics applications. With rapid technical advancements in the past two decades, macro-level fluid-dynamics simulation has been well studied and become more mature.

The state-of-the-art method for visually-appealing multi-phase fluid animation is an attractive topic in computer graphics. The high-fidelity multi-phase fluid animation with realistic details need the support of physics-based model and efficient algorithm. Therefore, it is significant to build an efficient temperature governed multi-phase transition model by simplifying the physical-based heat condition.

Despite the recent successes in restricted phase-change simulation (such as liquid-gas [1–3] and solid-liquid animation [4,5]), certain difficulties still prevail and need to be resolved for high-fidelity multi-phase fluid animation. Specific challenges can be summarized as follows. First, solid, liquid, and gas have different geometrical morphologies and apparent characteristics of their own. Although the combination of existing techniques can provide a basic transition framework, the simplification of the heat-based systems for improving solution efficiency is an important goal to meet the needs of fluid animation. Second, the multi-phase transitions of solid, liquid, and gas involve many complex elements and closely relate to heat transfer, flow field, and multiple physical procedures. Considering the tightly coupling of complex physical phenomena, it is very important to formulate the integrated models involving different materials and multiple phases to express convincing details in a robust way.

Strongly motivated by the need for heat-based multiple phase fluid animation, this paper concentrates on simulating liquid-gas

* Corresponding author.
E-mail address: lishuaiouc@126.com (S. Li).

https://doi.org/10.1016/j.gmod.2017.09.001
1524-0703/

and solid-liquid transitions as well as their flexible interactions. We develop a unified heat-based method to handle the transitions among solid, liquid, and gas, and employ MultiFLIP method [6] as our basic framework, which is convenient to incorporate our heat transfer model. In contrast to the existing works, the novelty of our work is that, we introduce a simplified heat transfer model which is coupled with MultiFLIP solver to simulate phase transition in an efficient manner. Moreover, to simulate the complex multi-phase fluid details, we use a robust boundary condition and a penetration prevention method for liquid-solid stable interactions, as well as a dissolved gas model and a multiscale bubble model to express realistic visual effects. The salient contributions of this paper are summarized as:

- We propose to couple an efficient temperature field and a latent heat model with MultiFLIP model. With the proposed heat-based method, we can greatly simplify the physical-based system and improve the solution efficiency to simulate the consistent phase transition process (such as freezing, melting, and boiling) at the same time.
- We introduce a novel physics-based method to mimic the dissolved gas that is involved in liquid, which further enhances the bubble generation effects.
- We propose a SPH-FLIP coupled method to model different types of bubbles and to provide vivid interactions among solid, liquid, and gas, wherein SPH-based sub-grid bubbles and grid-based large bubbles can be conveniently converted to each other.

2. Related works

In recent years, there have been many methods developed to simulate fluids, including SPH [7–10], the position based methods [11,12], Lattice Boltzmann Method (LBM) [13,14], the data-driven methods [15,16], the FLuid Implicit Particle (FLIP) and Particle in Cell(PIC) methods [17–19], etc. The FLIP method was introduced in computer graphics by Zhu et al. [20], and then was extended to simulate splashing water [21], conduct fluid-solid coupling [22], combine with particle method [23], model multi-scale droplet/spray [24], etc. Ando [25] and Selino et al. [23] respectively proposed methods to improve the particle distribution of FLIP method. Particularly, since MultiFLIP can model two-phase flow by extending the FLIP method, it has inspired us to further extend MultiFLIP to simulate the mixtures of solid, liquid and gas with a heat driven model. The simulation of heat-based fluid mainly involves two subfields: multi-phase fluid interaction and phase transition. We briefly summarize the most relevant previous works below.

Phase transition. The common simulations of phase transition can be roughly categorized into two types: liquid-gas and solid-liquid, which are usually driven by temperature and heat transfer.

One of the most ubiquitous liquid-gas transition phenomena is the water boiling. Mihalef et al. [26] explored the visual effects of boiling based on the physical laws of heat transfer and mass transition of vaporization, but ignored sub-grid bubbles. Later on, Kim et al. [27] extended Yanagita model to simulate boiling. Another classic liquid-gas phenomenon is the release of water-dissolved gas (which generates small bubbles in liquid), which has been studied by Cleary [3] and Patkar et al. [28]. Patkar et al. [28] specifically focused on designing a sub-grid bubble model along with a Poisson formulation which coupled grid-based liquids with particle bubbles.

Melting and freezing are the main topics related to solid-liquid transition. Zhao et al. [29] modified the Lattice Boltzmann method (LBM) to simulate melting and the liquid flow by taking into account heat transfer. Losasso et al. [30] employed triangle mesh and particle level set to represent solid, liquid and gas, covering the transitions from solid to both liquid and gas. To reduce the complexity of generating liquid from solid, Chang et al. [31] extended SPH by adding an elastic stress. Iwasaki et al. [4] also integrated heat transfer into SPH to simulate the dynamics of droplets and water flows on ice surface. Ice formation is the main focus in current freezing simulation. Maréchal et al. [32] employed the finite volume method to simulate the conductive, convective and radiative heat transfer, and utilized voxel representation to animate the phase changes among snow, water and ice. Rather than using physics-based models, Gagnon et al. [33] decomposed the icicle formation into four stages, and obtained an efficient and parameter-controlled procedural model. Moreover, Im et al. [5] focused on how to release the dissolved air in water. And Stomakhin et al. [34] extended material point method to handle heat transport, melting and solidifying materials.

In this paper, we propose to simulate multi-phase transitions with a simplified temperature field. All the materials are represented by unified FLIP particles, and the heat condition is solved by coupling them with the grid-based solver of FLIP. A latent heat model is introduced to mimic the dissolved gas within liquid, which can further enhance bubble generation effects.

Multi-phase interaction. Multi-phase fluid interaction is important for state-of-the-art fluid simulation. The generation of bubbles and their movements are the most common phenomena in liquid-gas interaction. To model bubble geometry, Zheng et al. [35] developed a regional level set method for multi-manifold interface tracking. Boyd et al. [6] extended the hybrid particle-grid method to handle the liquid-gas coupling. Since level set is suitable for large well-resolved bubble and particle is capable of simulating sub-grid details, Hong [36], Patkar [37], Mihalef [38] and Kim et al. [28] respectively combined their advantages to simulate the rich details of multiple scale bubbles. Meanwhile, Busaryev et al. [39] treated bubble particles as the sites of a weighted Voronoi diagram to model the interaction of small bubbles in dense foam. Kim et al. [40] modeled numerous dispersed bubbles as a continuum, capable of animating complex phenomena with millions of bubbles. Ren et al. [41] modeled bubbles in fluid by combining the volume fraction data (obtained from fluid simulation) with a simple secondary bubble simulation.

For the coupling of solid and fluid, Solenthaler and Akinci et al. [6,42] proposed two-way coupling approaches, wherein the interaction between solid and liquid was modeled by way of forces. Allard et al. [43] and Yang et al. [44] extended the force-based interaction method to couple mesh based solid and particle-based fluid. On the other hand, Losasso, Mosher et al. [45–47] employed full Eulerian method for the coupling, in which both the velocities of liquid and solid were projected onto the grid and solved by a grid based solver. For the complex scenario having solid, liquid and gas simultaneously, Wang et al. [13] employed SPH to simulate bubble generation and its coupling with solid, while Macklin et al. [12] extended position-based dynamics to model their two-way coupling.

Inspired by the related works above, we aim to build an efficient framework involving three types of gas-liquid-solid interactions. We design specific collision-handling algorithm to guarantee the numerical stability and non-penetration condition of solid. We have found that very few existing techniques can handle the complicated gas-related interactions in a physical-based manner in terms of both heat and dynamics aspects. Therefore we particularly focus on the simulation of realistic bubble-liquid-solid interaction, and the interaction will be further enriched by introducing the force of an implicit incompressible SPH model into our FLIP solver. Moreover, we enable the flexible transformation of different bubble types, which makes our simulation more vivid and closer to the real world.

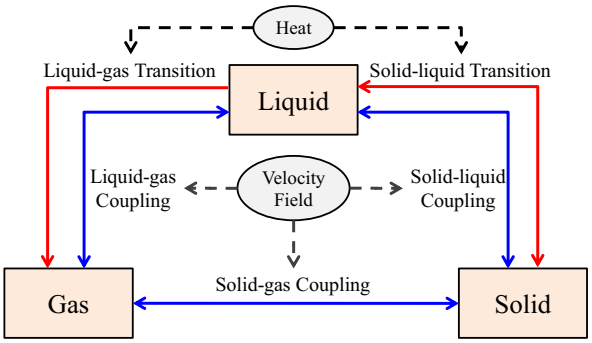

Fig. 1. The architecture of our framework.

Table 1
Symbol list of our equations.

Symbols	Meaning	Solver
$\mathbf{u}$	Velocity	Grid
p	Pressure	Grid
$\mathbf{P}$	Position	Grid
Δt	Timestep	Grid
T	Temperature	Grid
L	Latent heat	Grid
h	Grid size	Grid
ω	Angular velocity	Particle
m_i	Solid/liquid mass	Particle
m_g	Gas mass	Particle
$\mathbf{v}$	Velocity	Particle
$\mathbf{f}$	Force	Particle

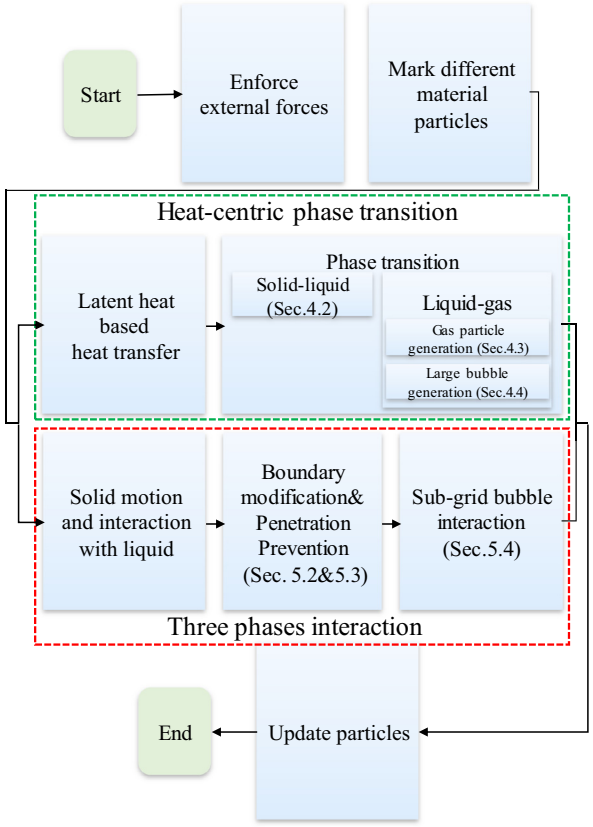

Fig. 2. The pipeline of our approach.

3. Method overview

To simulate the phase changes among solid, liquid and gas, as well as their seamless interactions, our framework mainly consists of two parts: heat-based phase transition and velocity field based coupling, as shown in Fig. 1. First, we model the heat transfer procedure in a grid-based way, which can tackle anisotropic diffusivity of heterogeneous materials. The solid-liquid and liquid-gas phase transitions are modeled based on temperature and latent heat, which are marked with red arrows in Fig. 1. Second, we extend MultiFLIP model to handle the interactions of multiple materials, which are marked with blue arrows in Fig. 1. To animate the sub-grid bubbles and their interactions with other materials, we turn to customized SPH with the grid solver of FLIP. Specifically, all of the involved techniques are integrated in a unified model within a heat-based framework. On that basis, we bridge the physical meanings and visual realism to automatically handle the phase transitions among liquid, solid and gas.

The pipeline of our hybrid framework is shown in Fig. 2, which can support seamless unification of phase transition and interaction. We use different colors dotted boxes to distinguish transition and interaction techniques. However, these two processes are coupled in our framework and happen simultaneously. For better understanding of our symbol system, we list the most important symbols used in the equations in Table 1 and describe which solver they belong to.

3.1. MultiFLIP model

In our framework, our hybrid model is built on MultiFLIP [6], which can seamlessly blend grid-based solver towards the natural yet strong coupling between FLIP and heat transfer model. We now briefly review the basic idea of fluid simulation, MultiFLIP model and its possible improvements. Fluid dynamics is essentially based on Navier–Stokes equations (N-S equations), which conserves both mass and momentum:

$$\frac{\partial \rho}{\partial t} + \rho \nabla \cdot \mathbf{u} = 0, \tag{1}$$

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}, \tag{2}$$

where ρ is the density, $\mathbf{u}$ is the velocity, p is the pressure, and $\mathbf{f}$ is the external force. In the MultiFLIP model, fluid is discretized as particles, and the traditional Eulerian method is employed to solve the N-S equations. The velocity change on grid (rather than the velocity itself) is interpolated over particles to avoid the numerical dissipation problem. To suppress the arising high frequency noise, FLIP velocity usually blends with Particle-in-Cell (PIC) velocity as that used in [20]:

$$\mathbf{v} = \alpha \mathbf{v}_{FLIP} + (1 - \alpha)\mathbf{v}_{PIC}, \tag{3}$$

where the blend factor α is set to be 0.95 in all of our experiments. To handle two-phase flows, the MultiFLIP model constructs a velocity field for each phase, wherein it employs level set to indicate the interface and then synthesizes a unified velocity field by taking into account the velocity fields of all phases.

In our framework, we employ a temperature field based on heat transfer to control the phase changes. Different phases accomplish inter-conversion via heat exchange. The latent heat based unified model handles the transitions between different types of particles, and marks them with corresponding phase flags. The velocity field of each phase is then updated separately, and the particle velocities are consequently updated. Besides, to prevent the different-phase particles from mixing and guarantee the stability of the simulation, it is critical for MultiFLIP model to maintain a smooth interface between different materials, which requires correcting positions in each time step [6].

4. Phase-transition model

In this section, we first introduce the heat transfer simulation method used in our framework, then describe how to model the solid-liquid and liquid-gas transitions respectively, and at last we will detail how to integrate the handling of the dissolved gas in liquid with a latent heat model. Since the realistic phase transition is generally represented by phase diagrams which are very complex and hard to be recovered by a simple simulation framework, we ignore these complexity and instead to introduce an simplified temperature field and latent heat model to imitate the heat-based phenomena in a physically reasonable way as much as possible.

4.1. Heat transfer

Compared to existing methods, using FLIP for phase changes brings the benefit that we can universally use particles to represent all the materials and their physical properties (such as temperature), and the heat convection can be solved conveniently by the grid scheme.

Meanwhile, we ignore the heat radiation for the purpose of simplicity, so we only need to model heat diffusion in each time step, which is governed by the heat equation:

$$\frac{\partial T}{\partial t} = a, \tag{4}$$

where a is the thermal diffusivity (depending on the material), and δT is the Laplacian operator.

To model the heat diffusion among different materials, we prefer to solve Eq. (4) on grid rather than on particle, because it is more robust to solve Eq. (4) on grid considering the possible uneven particle distribution in FLIP. Accordingly in each time step, we first map the temperature to grid cells before we solve the heat equation on the grid, and afterwards map the temperature back to particles. The temperature of particles is mapped to the center of each grid cell with a weighting function, and the heat exchanging among the particles driven by the grid-based solver is:

$$T_g = \frac{\sum_{i \in S_g} T_i m_i W(d_i, r)}{\sum_{i \in S_g} m_i W(d_i, r)}, \tag{5}$$

where T_g, T_i are respectively temperatures of grid cell and particle p_i, S_g is the surrounding particle set, m_i is the particle's mass, d_i is the distance between the cell center and the particle, r is the kernel radius, and $W(d, r) = max(0, 1 - d^2/r^2)$ is a weighting function.

When we simulate the heat transfer, the density difference between water/solid particle and gas particle is quite large, which means the diffusivity difference of them will be very large, too. Consequently, a direct summation over diffusivity values of all the particles (gas, water, solid) may cause significant errors. We take two independent measures to handle the heat transfer processes related to air phase. When heat transfers from free air to liquid/solid particle, we assign a default temperature T_{air} which can be used as a given temperature of the surface liquid/solid particle for the empty cell containing no particle which represents the free air; when heat transfers within a cell involving both gas and liquid particles, we consider that the gas particles make no contributions to the temperature of a liquid gird since the mass(m_i of Eq. (5)) and number of gas particles are just too small.

To solve the heat equation and obtain the temperature field on grid, we first discretize it as follows:

$$\frac{T^{n+1} - T^n}{\Delta t} = a\left(\frac{\partial^2 T^{n+1}}{\partial x^2} + \frac{\partial^2 T^{n+1}}{\partial y^2} + \frac{\partial^2 T^{n+1}}{\partial z^2}\right), \tag{6}$$

where T^n is the given temperature field obtained in the last time step, and T^{n+1} is the current one we need to update. We solve this equation using a standard Conjugate Gradient (CG) solver, and employ a central difference method to discretize the temperature field. In our simulation, we assign the corresponding thermal diffusivity parameter for each grid cell by synthesizing the particles' materials using Eq. (5), wherein it replaces T with a as a variable.

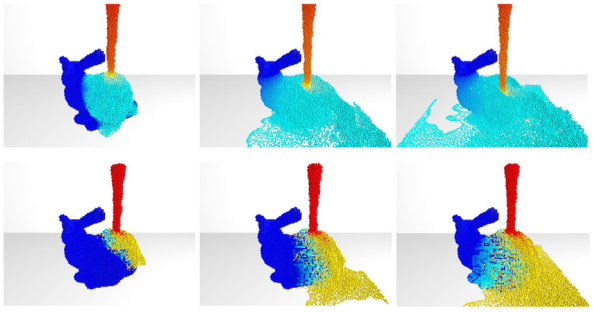

Fig. 3. Particle views of heat transfer simulations. The particles are colored according to their temperatures (blue means low temperature and red means high temperature). The top row shows a melting procedure, the bottom row shows a freezing procedure, wherein the initial temperatures for solid, liquid and free air are respectively $-10°C$, $30°C$, and $0°C$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

After updating the temperature on grid T_g^{n+1}, we need to map temperature back to particles. However, directly mapping it back tends to suffer from excessive numerical dissipation, which is similar to the ill-posed PIC method [20]. So, instead we employ the FLIP scheme for the particles' temperature updating, map the temperature changes of the grid to particles (obtaining δT_p for each particle), and then update the particle temperature by $T_p^{n+1} = T_p^n + \delta T_p$. Fig. 3 shows the heat transfer procedure among heterogeneous materials. Heat diffuses more rapidly in the materials with higher thermal diffusivity coefficients.

4.2. Latent heat model for melting and freezing

The phase transition model that incorporates latent heat has been borrowed from Stomakhin et al. [34]. As a solid is heated, temperature increases until reaching the melting point. Then the supplied heat is transferred into latent heat to break the chemical bonds, which causes the changes from solid to liquid. On the contrary, as heat is drained out of the liquid, its temperature drops to the freezing point, and then it loses latent heat and changes into solid. So we take both temperature T and latent heat L into consideration.

In our latent heat model, every particle is marked by a flag (F_{fluid} or F_{solid}) to identify which material it belongs to, the solid particle will be involved in a global constraint that controls the shape and movement of the solid, and the particle with F_{fluid} represents a free fluid particle. We update its temperature and latent heat in each time step, and then determine whether its phase flag needs to be changed or not. So melting and freezing procedures would be in charge of the flag updating.

After obtaining the temperature of each particle documented in Section 4.1. This temperature updating is only depending upon the heat transfer while ignoring the possible phase changes, which may lead to physical incorrectness. For example, a solid particle's temperature is higher than a fluid particle's temperature. To tackle this problem, we correct the temperature by transferring excessive temperature to the change of latent heat as follows:

$$\Delta L = m_i c(T - T_c), \tag{7}$$

where ΔL is the change of latent heat, T_c is the melting or freezing point temperature. For melting, c denotes specific heat capacity of water, and when simulate freezing, c is specific heat capacity of ice. When a particle's temperature reaches to melting/freezing

point, we do not change its phase flag eagerly, instead we store the latent heat to imitate the critical state (heat transfer happens without temperature change). When the absolved/released heat of a particle oversteps the heat that a particle can store L, we then change this particle's phase by changing flag: $F_{fuild/solid}$:

$$\begin{cases} F_{fluid} \leftarrow F_{solid}, \Delta L - L_{THmax} \geq 0 \\ F_{solid} \leftarrow F_{fluid}, L_{THmin} + \Delta L \leq 0. \end{cases} \tag{8}$$

Through this scheme, we can simulate the sustained melting procedure by increasing the solid particle's latent heat while keeping its temperature at the melting point. When the latent heat is high enough to break the chemical bonds, the phase change happens. Therefore, we set a threshold L_{THmax} for the latent heat. On the other hand, freezing happens when a liquid particle's latent heat is dropped below a threshold L_{THmin}, with temperature at the freezing point. It should be noted that, as for our unified framework, L_{THmin} equals to L_{THmax} and is denoted as $m_i L_{unit}$, where L_{unit} is the specific latent heat required for unit mass. For water, it is 334 kJ/kg.

In this model, we unify melting and freezing as two inverse physical processes, wherein temperature and the latent heat determine the transitions. As shown in Fig. 3, melting and freezing are simulated uniformly by our approach when the initial conditions are different. Since our heat-based model ignore the complex physical system and instead to introduce the simplified temperature field and latent heat model decoupled from the fluid dynamics, it greatly simplifies the physical system and improves solution efficiency.

4.3. Heat-based generation of gas particles

Similar to melting and freezing phenomena, the transition from liquid to gas is also dominated by the latent heat model. As the liquid absorbs heat, its temperature increases up to boiling point T_{boil}, and then the latent heat of vaporization is accumulated. So we update the liquid particle's latent heat ΔL_p according to $\Delta L_p = m_i c(T - T_{boil})$, which is similar to Eq. (7).

However, which is different from melting process, the transition from liquid to gas can not be completed by simply changing the flag of liquid particle due to the large density difference between liquid and gas. Actually, liquid particle generates gases and bubbles are formed gradually. To simulate this process, we first compute the fraction of gas generated from liquid particle by checking the change of heat in each time step:

$$c_{trans} = \begin{cases} 0, & \Delta L_p - L_{THmax} < 0 \\ \frac{\Delta L_p - L_{THmax}}{L_{gas}}, & L_{gas} > \Delta L_p - L_{THmax} \geq 0 \\ 1, & \Delta L_p - L_{THmax} \geq L_{gas} \end{cases}, \tag{9}$$

where L_{gas} is the specific latent heat required for unit-mass gas generation. $c_{trans} = 0$ means there is no dissolved gas released from liquid, and $c_{trans} = 1$ means all the gas dissolved in liquid will be released. Eq. (9) controls what percentages of gas dissolved in liquid will be released as the temperature increases, when the liquid is reaching to boiling point.

Then we need to know the solubility to decide the generated exact quantity of gas. Given the temperature and pressure, the gas solubility measures the maximum dissolved gas in liquid. The solubility of gas in liquid can be computed by the following equation, called Henry's law:

$$c_{aq} = k_{H,cp} p, \tag{10}$$

where c_{aq} is the concentration of gas in the solution, $k_{H,\,cp}$ is the Henry's law constant, p is partial pressure of gas above the solution. Meanwhile, the relation between Henry's law constant and temperature is defined as:

$$k_{H,cp}(T) = k_{H,cp}(T^{\Phi}) exp[C(\frac{1}{T} - \frac{1}{T^{\Phi}})], \tag{11}$$

where T is the temperature, T^{Φ} is the referred standard temperature. In our method, we assume the parameters p and C to be constant, so that the solubility of gas only depends on the temperature. As the temperature increases, the Henry's law constant $k_{H,\,cp}(T)$ decreases, meaning the solubility of gas c_{aq} decreases. When the gas contained in liquid particle is larger than its solubility, the excess part will be released. In each time step, we compute the solubility of liquid particle based on the given temperature (Eq. (10)), and then determine the gas quantity Δm_g that should be released by

$$\Delta m_g = \begin{cases} 0, & c_{trans} < c_{aq} \\ m_i \cdot (c_{trans} - c_{aq}), & c_{trans} \geq c_{aq} \end{cases}. \tag{12}$$

Through the above process, we construct a dynamic relationship between the latent heat model and the phase transition model. Then we store the gas concentration for each liquid particle, which is initialized at the beginning of simulation.

After quantifying the gas that should be generated from each liquid particle, we determine the nucleation site to seed the gas particles. There are two types of nucleation sites in our approach, the random preset ones on the solid boundary (similar to [26]), and the existing gas particles. We utilize each grid cell as a separate computational unit, in which we sum up the gases contained in liquid particles. We generate a gas particle when there is at least a nucleation site in the cell and the accumulated gas quantity ($\Sigma \Delta m_g$) is equal to a gas particle's mass m_g.

4.4. Generation of large bubbles

Fig. 4 illustrates the grid-based solver. The shapes of solids, liquids and large bubbles are handled by grid-based solver, while SPH based sub-grid bubbles are employed to model the interaction phenomena, which will be detailed in Section 5.3. To model the transition from liquid to gas, we use the gas particles for grid-based bubble generation, the multi-phase mixture is based on MultiFLIP [6].

As that in MultiFLIP method, given the velocities and positions of liquid and gas particles, we map their velocities respectively on grid to construct two separate velocity fields. Then we reconstruct a level set to represent the interface between the liquid and gas, on top of this, we synthesize a new velocity field according to the velocity fields of gas and liquid. N-S equations (Eqs. (1) and (2)) are solved on grid to get a divergence-free velocity field, and we update the velocity of liquid and gas particles respectively by FLIP method.

Algorithm 1 shows the details of our phase transition model. Each operation within the scope of "for each... do" is accomplished by invoking threads on CUDA, and then they are executed in parallel. Our program strictly follows the flow listed in the algorithm, thus, some operations are not in the same execution sequence, as they are introduced in the aforementioned sections.

5. Dynamic model of three-phase coupling

In this section, we introduce the method used for multi-phase interactions of solid, liquid and gas. We first introduce our grid-based solver, then describe how to handle the boundary condition and how to prevent penetration for liquid-solid interaction, finally, we introduce a novel SPH-based model to simulate sub-grid bubbles, which will significantly strengthen the reality of the three phase liquid-gas-solid interactions.

5.1. Grid-based solver

We use FLIP particles to represent all materials and mark different materials with different flags (e.g., F_{solid}, F_{fluid} and F_{gas}).

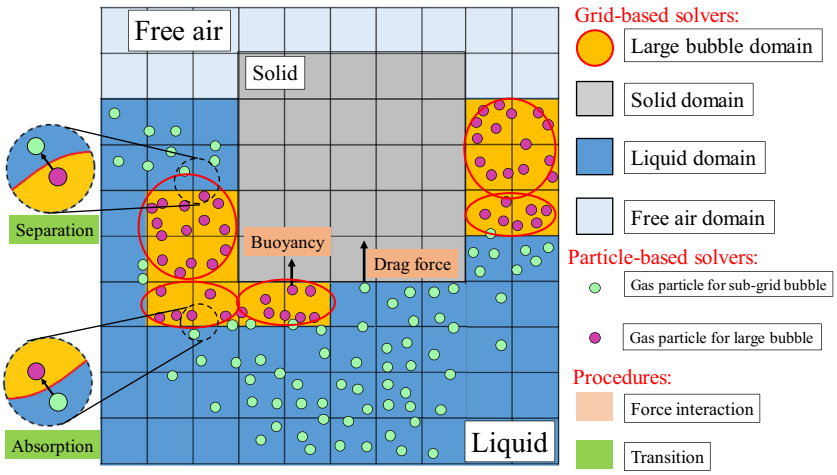

Fig. 4. Illustration of our unified solver. The simulation domains and their material attributes are highlighted by different colors. We do not draw the solid and liquid particles in the interest of both space and clarity. The gas particles are classified into two types, Type-one gas particles are employed for SPH-based sub-grid bubbles (colored in green), and Type-two particles (colored in purple) are clustered to form grid-based large bubbles (highlighted in red). As a result, they apply different forces to liquid and solid, and the two types of particles can transform to each other at the interface of large bubbles (through the absorption and separation procedures). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Algorithm 1 Phase transition model.

Require: Particles' positions, temperatures, material types
Ensure: Particles' new temperatures, material types
  for all grid cell **do**
    Interpolate a temperature field T_g^n from particle.
    Set thermal diffusivity a according to material type.
    Compute T_g^{n+1} by solving Eq. 6.
  for all particle **do**
    Map temperature change from ΔT_g.
    Update latent heat according to Eq. 7.
  for all solid particle **do**
    Check whether it is melted to a liquid particle (Section 4.2).
  for all liquid particle **do**
    Update solubility by Eq. 10.
    Update the generated vapor quality (Eq. 12).
    Check whether it is freezed to a solid particle or not (Section 4.2).
  for all grid cell **do**
    Summarize the mass of generating gas.
    Seed gas particles according to Section 4.3.

Inspired by [48,49], we sum up solid particles' velocities to obtain the global linear and angular velocities for rigid constraint, and confine its movement to be translation and rotation only. External forces on solids are enforced before conducting grid-based solving. In such way, the velocity interaction between solids and other materials can be solved.

In sharp contrast to the MultiFLIP model, our model does not discretize free air into particles, wherein bubbles will burst when they reach the liquid surface (i.e., foam is not involved). We detect the bubble's status at each time step, and delete the internal gas particles when it is exposed to free air. To achieve this goal, we mark each grid cell according to the occupying-particles' materials, and a cell has no particles will be marked as free air. Then we use the flood-fill algorithm to expand the free-air cell. By repeating this process until there is no mark change on the grid, we can get all the bubbles that may encounter free air on the grid, and delete the gas particles inside them at each time step.

5.2. Solid boundary and penetration prevention measurement

When simulate fluid, fluid boundaries are marked as static grids with no physical attributes, and the fluid interaction with boundary grid will rebound in an inverse direction. By this way, fluid particles can ensure the conservation of momentum, because they are discrete and independent of each other along the directions of their velocities. However, for fluid-solid interaction, the solid particles are clustered together, simply take the same boundary conditions will lead to unrealistic global deformation, and affect the simulation stability. For a set of solid particles, we allow transitory penetrating into a virtual boundary grid to keep the global shape unchanged. Thus, we use a virtual boundary condition for solid.

And when handle the interface of fluid and solid particles, how to prevent different particles penetrate into each other need to be considered. Inspired by the position correcting idea used for smooth interface [6], we use an additional algorithm to prevent particle penetration. The algorithm of virtual boundary and penetration prevention are the same as our previous work, researchers who have interests can find more technique details in [50].

5.3. SPH-based sub-grid bubbles

In Section 4, we have detailed the transition process from gas particles to grid-based bubbles, the generated large bubbles can be naively treated as FLIP particles. However, for a lifelike fluid phenomenon enriched by bubbles, large bubble makes limited contributions towards force-driven interaction. On the one hand, bubbles form and break all the time, besides grid-based large bubbles, sub-grid bubbles play a more important role for realistic visual effects embracing details. On the other hand, while interacting with other materials, larger bubbles have only buoyancy and surface tension, which are not enough to perform the effective dynamic interaction among multiphases.

To realize vivid liquid-gas-solid interactions, we propose an implicit incompressible SPH (IISPH) model [23] to imitate sub-grid bubbles. Compared with multi-phase models, IISPH preserves the particle's volume well and does not suffer from compression, which is conducive to simulating the solo bubble's motion in liquid. It is possible to select a part of gas particles as sub-grid bubbles (as shown in Fig. 4), which play a more important role in interacting with solid, and retain the other gas particles for large

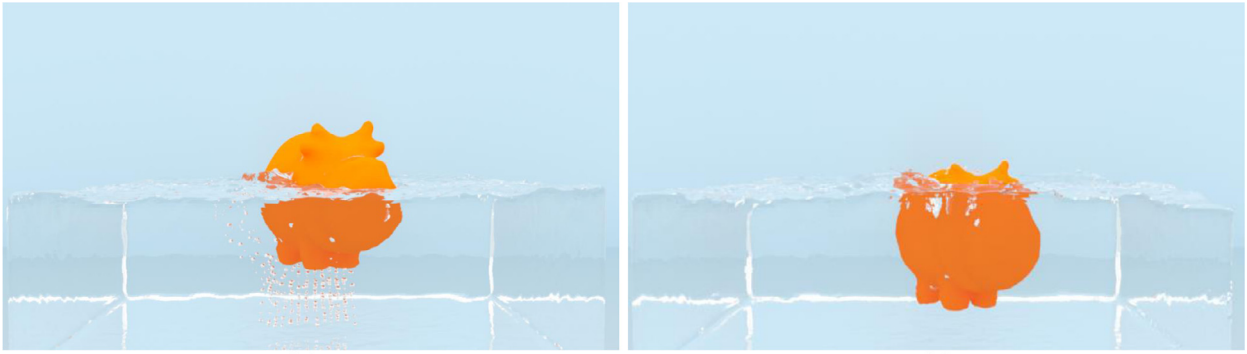

Fig. 5. Sub-grid bubbles simulation. We artificially pour gas particles into the bottom, sub-grid bubbles are generated from bottom and interact with solid and liquid, and then break in the liquid surface. Fig.(a) shows the sub-grid bubbles interact with solid and liquid and generate large bubbles; Fig.(b) shows the liquid-solid interaction without any bubbles' effect.

bubble generation. At the algorithmic level, we make use of an IISPH model to permit the chosen gas particles to dynamically interact with solid and liquid. And at the topological level, we represent sub-grid bubbles by modeling these SPH-based gas particles with small spheres, whose radii are randomly generated from 0.01 to 0.05 h to represent different bubbles' scale.

In a time step t, if a gas particle is chosen as a sub-grid bubble, we need to re-initialize it with the mass of sub-grid bubble m_p, interpolate its position from grid, and perform the external force (buoyancy force) by

$$\mathbf{f}_b = \frac{4\pi R^3}{3}\rho\mathbf{g}, \tag{13}$$

where $\mathbf{g}$ is gravity, ρ is the density of liquid, R is the random radius. Thus, sub-grid bubbles with different radius will perform uneven buoyancy. Then, we use IISPH for the sub-grid bubble to get the SPH velocity v_s [23], the pressure and viscosity forces are computed by solving a Poisson equation. And in the next time step $t + \Delta t$, the sub-grid bubble's velocity will be interpolated back to FLIP particles by blending FLIP and PIC [20,21,25]:

$$\mathbf{v}_{i_F} = \mathbf{v}_{i_F}(t) + \frac{1}{\sum_{j_S} W_{ij}} \sum_{j_S} \left((\mathbf{v}_{j_S}(t + \Delta t) - \mathbf{v}^*_{j_S}) \right) W_{ij}, \tag{14}$$

$$\mathbf{v}^{PIC}_{i_F} = \frac{1}{\sum_{j_S} W_{ij}} \sum_{j_S} \mathbf{v}_{j_S}(t + \Delta t) W_{ij}, \tag{15}$$

where $\mathbf{v}_F$ is the velocity of FLIP particle, $W_{ij} = W(||x_i - x_j||, R)$ is the SPH kernel function with compact support, which is different to [23], to correctly match with the geometrical model, the spacing of SPH particles is equal to the geometric radius. i and j are the positions of FLIP or SPH particles. The blended velocity is finally computed by Eq. (3).

Since the sub-grid bubbles are simulated by a separate particle-based model rather than the grid-based solver, we need to model their interactions with other materials, wherein their interactions with grid-based liquid are modeled by way of a drag force, which is computed as follows

$$\mathbf{f}_{drag} = \sigma m_i (\mathbf{v}_p - \mathbf{v}_g)|\mathbf{v}_p - \mathbf{v}_g|, \tag{16}$$

where σ is a constant coefficient, m_i is the particle's mass. This force is enforced to both sub-grid bubble and liquid in order to conserve momentum. When a sub-grid bubble reaches to the liquid surface, we directly delete it. Fig. 5(a) demonstrates the sub-grid bubbles interacts with solid and liquid, to exhibit the effect of SPH-based solver, we prevent the generation of large bubbles, and all gas particles convert to sub-grid bubbles.

In our method, both types of bubbles are represented by the gas particles, and the difference between them is whether the gas particles should be clustered to form large bubbles. We can freely choose some of the gas particles to generate large bubbles for phase transition, and retain the others to simulate sub-grid bubbles' dynamic interactions with liquid and solid. Moreover, we propose an effective approach to handle the transition between large well-resolved bubbles and small invisible sub-grid bubbles. When a bubble is large enough, the level set value interpolated at its position will indicate that it belongs to a large bubble [6]. On the other hand, when a bubble is small, liquid will dominate the level set value, and it indicates the gas particle is a sub-grid bubble in liquid and needs not to be rendered as a sphere. Thus, by setting a threshold value for the level set, we can dynamically determine whether a gas bubble is in a grid-based bubble or not, and then change the simulation model of gas particles accordingly (i.e., grid-based model for large bubbles and SPH model for sub-grid bubbles). With this strategy, sub-grid bubbles are absorbed naturally when they encounter large bubbles, and is transitioned into a large bubble.

Since we only need to check the level set value at each gas particle's position, our method is quite efficient, and the entire simulation procedure is similar to the case in real world. Fig. 10 shows the whole phase transition and interaction procedure, wherein a part of gas particles generate large bubbles, and others turn to sub-grid bubbles. This phenomenon displays both the phase transition and interaction effects. Algorithm 2 shows the details of our dynamic model of three-phase coupling.

6. Implementation and experimental results

6.1. CUDA-based numerical computation

We have implemented our entire method on CUDA for the efficiency purpose. To accelerate memory access, all physical properties are stored in the global GPU memory, such as positions, velocities, forces, temperatures, latent heat, etc. For each particle, we invoke a CUDA thread to calculate which grid cell it belongs to, and then use a CUDA thread for each grid cell to interpolate its velocity from particles. The multiFLIP solver's acceleration follows [24,51] for efficient storage management. For each particle with SPH constraint, we store the particle flag to accelerate the interpolation.

To couple the phase transition and interaction model, grid-based solver is computed first, followed by particle-based model for sub-grid bubbles. This is accelerated with GPU by the widely-used hashing and sorting method in SPH [52], which stores the indices of all the involved particles for each grid cell. The penetration prevention is enforced when the velocities and

Algorithm 2 Dynamic model of three-phase coupling.

Require: Particles' positions, velocities, material types
Ensure: Particles' new positions, velocities
for all particle **do**
Enforce external forces.
for all grid cell **do**
Compute a level set field *LS*.
for all gas particle **do**
Determine whether it belongs to a large bubble or represents a sub-grid bubble (Sec.5.3).
for all grid cell **do**
Construct velocity fields for each material.
Synthesize a unified velocity field $\mathbf{u}_g^n$.
Solve N-S equations, obtaining $\mathbf{u}_g^{n+1}$.
for all particle **do**
Interpolate particle velocity from $\mathbf{u}_g^{n+1}$.
for all solid object **do**
Compute a global linear and angular velocity.
Update the particles' positions and velocities.
for all liquid and gas particle **do**
Update the velocity and position while preventing penetration into solid (Sec.5.2).
for all sub-grid bubble particle **do**
Compute the density and pressure in IISPH (Sec.5.3).
Compute the pressure force, viscosity force, and drag force.
Update the velocity and position with the mechanism of penetration prevention.

Table 2
Parameter settings in our experiments.

Parameters	Values	Unit
a for solid (Eq. (4))	0.2	m^2/s
a for water	0.14	m^2/s
a for air	19	m^2/s
c for water (Eq. (7))	4.2	$kJ/(g\cdot^\circ C)$
c for ice	2.1	$kJ/(g\cdot^\circ C)$
$T_{melt/frezz}$	0	$^\circ$C
T_{boil}	100	$^\circ$C
L_{gas}	2260	kJ/kg
L_{THmin}, L_{THmax}	0, 0.2	kJ
p (Eq. (10))	1	atm
T^Φ (Eq. (11))	273	K
C (Eq. (11))	1500	K
σ (Eq. (16))	0.03	–

positions of particles get updated rather than being a standalone process.

6.2. Experimental results and discussions

Our framework is implemented with C++ and CUDA, and all the examples are run on a PC with Nvidia Geforce GTX 1080 GPU and Intel Core i7 CPU. Table 2 shows the default parameters used in our experiments, and Table 3 lists the experimental performance of CUDA-based implementation. It should be noted that the computation time for liquid and droplets depends on both particle number and grid resolution. Fig. 3 use little spheres to render particles, and all the other models are represented as triangular meshes. Our experiment results involves 2 million particles at most.

Fig. 6 shows the water boiling procedure in a pot, wherein heat is continuously imported from the bottom of the pot. As temperature rises, the gas dissolved in water is released and forms bubbles around the seeding positions. As a bubble gathers more gas and becomes large enough, it departs from the bottom and rises while its volume continues to grow. As the water temperature reaches the boiling point and heat is persistently imported from the bottom, bubbles are generated rapidly due to vaporization, and the water vaporization becomes intense with many rising bubbles, especially when large bubbles burst around the water surface.

Fig. 7 demonstrates the realistic rendering result of melting phenomenon, wherein the ice melts due to the heat absorbed from free air or liquid. On the other hand, Fig. 8 shows the freezing phenomenon. As the water flows across the surface of the ice bunny, it goes slower and loses heat, and new ice forms on the edges of the ice block. The only difference between these two scenarios is the initial temperature, and it proves that our approach is capable of modeling melting and freezing in a unified heat-based model.

In Fig. 9, an ice block is dropped into the boiling water. The ice melts rapidly due to the high temperature of boiling water. As the ice melts, the water temperature is lowered, and then the water stops boiling. Re-boiling happens when the water is heated back to the boiling point again. This scenario is designed to involve multi-phase transitions, wherein the coupling of solid, liquid and gas is also handled in a physical manner.

Fig. 5(a) shows the interactions among solid, liquid and gas. In order to concentrate on the SPH-based sub-grid bubbles, here the phase transitions are disabled. Gas particles are injected from the bottom of a water tank and interacts with a toy model dropped from the top. Fig. 10 illustrates the interactions and phase transitions, as small bubbles generate and rise, they can gather and transit to large bubbles. In these two scenarios, since the gas particles have initial velocities, the water is pushed upwards, and the solid bunny is lifted when it collides with bubbles. Moreover, bubbles rise around the solid bunny, which will not penetrate into the solid surfaces. As a comparison, Fig. 5(b) demonstrates a scenario with the same parameters except for the gas particles. Without bubble interaction, solid has less buoyancy and is submerged deeper in water.

In contrast to the prior works that only consider the generation of bubbles and air-fluid coupling, we model the heat transfer to drive the multi-phase exchanges, the simplified temperature field has greatly improves solution efficiency. Besides, our model takes the sub-grid bubble generations and interactions into consideration, which can well accommodate two-way transitions between SPH-based bubbles and FLIP-based gas particles.

Moreover, our method comprises heat transfer, phase change, gas generation, particle-gird hybrid model, etc. In our method, we employ the position-correcting approach [25] to maintain the uniform distribution of particles. However, we need to prevent oscillation during the transition between large grid-based bubbles and small sub-grid bubbles. Since they are simulated with different computational models, even when the simulation parameters are well tuned, converting them too frequently may cause instability problem.

Limitation. Since our approach mainly focuses on the flexible heat-based fluid animations and their interactions occurred in various natural phenomena, grid and particle-based solvers must be tightly coupled, which increases the computation time significantly. Meanwhile, our multi-phase transition does not involve condensation, sublimation and deposition, because they are relatively less visually-important but require complex water-cycle modeling. Besides, we also ignore the heat radiation for simplification, even though it may induce common melting phenomenon.

7. Conclusion and future work

We have detailed a hybrid heat-based framework to simulate multi-phase transitions. It involves heterogeneous material transitions and interactions governed by a simplified heat transfer model, which greatly simplifies the system and improves solution efficiency. All the phenomena are handled in a simplified temperature field and a latent heat model, and we can achieve multi-phase

Table 3
Time performance (in milliseconds) of our experiments.

Scene	Grid resolution	#Particles	Dynamics	Phase transition
Sub-grid bubbles (Fig. 5(a))	$64 \times 64 \times 64$	645.4k	155.8	0
Boiling (Fig. 6)	$96 \times 96 \times 64$	2060.4k	276.3	75.3
Melting (Fig. 7)	$64 \times 64 \times 64$	88.8k	81.8	37.6
Freezing (Fig. 8)	$64 \times 64 \times 64$	82.5k	80.9	38.8
Melting-boiling (Fig. 9)	$96 \times 96 \times 96$	1578.3k	222.3	72.0
Interaction (Fig. 10)	$64 \times 64 \times 64$	689.4k	155.6	40.2

Fig. 6. Boiling simulation. Dissolved gases are released as the temperature rises, and massive bubbles are generated when the water is boiling.

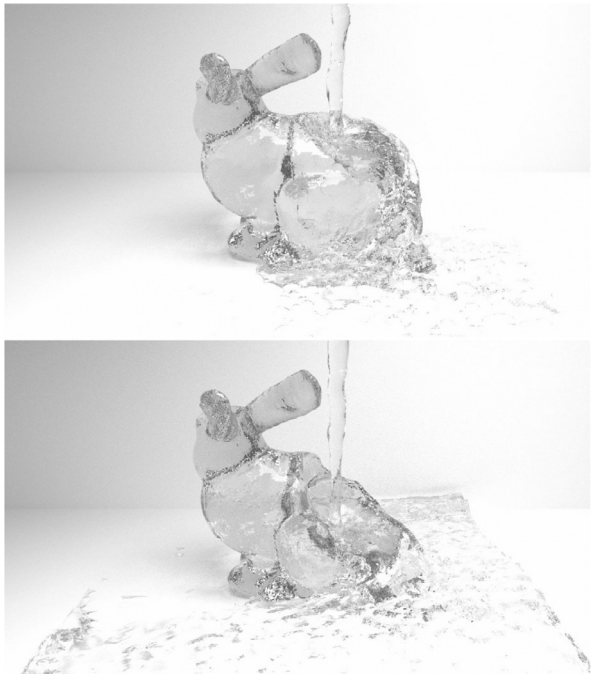

Fig. 7. Melting simulation. Hot water is poured onto a fixed ice bunny and makes it melt gradually.

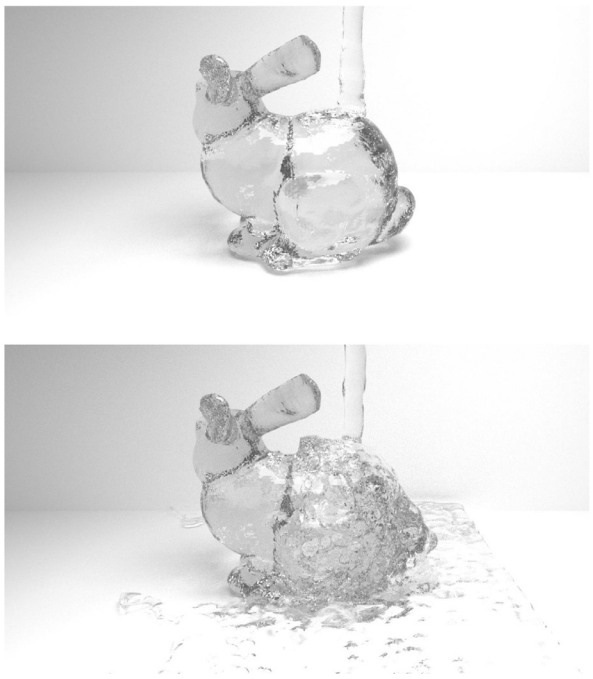

Fig. 8. Freezing simulation. Cold water is poured onto a fixed ice bunny and freezes on it, and the ice block is enlarged gradually.

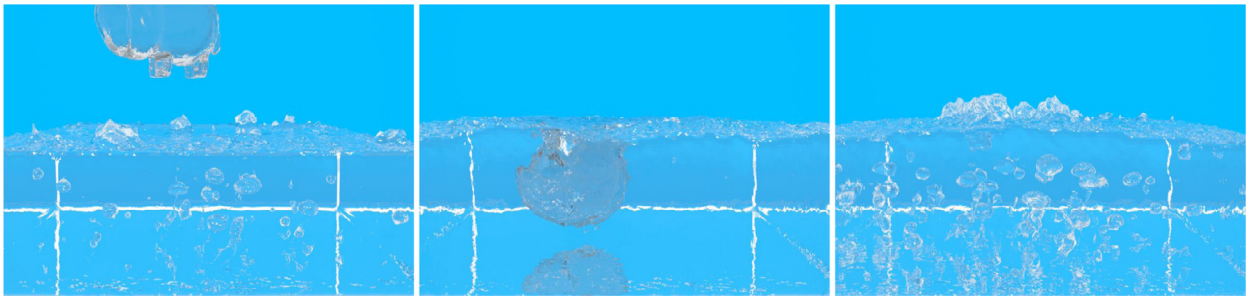

Fig. 9. Melting and boiling simulation. Heat the container bottom, water begins to boil, and an ice block is dropped into the boiling water, which melts gradually, and water begins to re-boil.

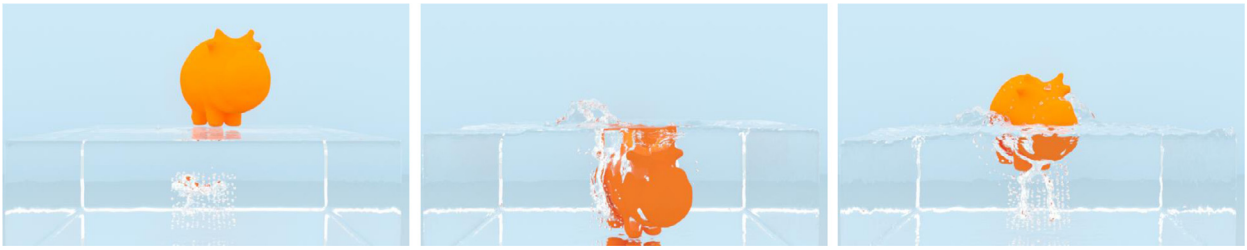

Fig. 10. Three-phase interactions and transitions. Gas particles are released from the bottom, large bubbles and sub-grid bubbles are generated, which interact with solid and liquid, and then break on the liquid surface.

transitions of solid, liquid and gas by changing the scenario temperature. Also, the gas dissolved in liquid is considered for bubble generation before boiling, and we also propose a novel method to model different types of bubbles. Grid-based large bubble is used for heat-driven phase transition and SPH-based sub-grid bubble is designed for the effective force-based interaction, while natural transition between sub-grid bubbles and large bubbles is achievable. Directly benefited from our framework, the solid, liquid, and bubbles are coupled together effectively in a seamless fashion, and such concerted multi-phase phenomena can be simulated and controlled under a fast and efficient solver. Our experimental results work well with demonstrated visual efforts and graphics appeal.

Currently we ignore the compressibility of the fluid, because it would significantly increase our method's computation complexity and deteriorate the overall time performance. We plan to integrate it into our framework by further improving our model. Meanwhile, in order to make our model accommodate the all-way water cycle transition with full functionalities among solid, liquid and gas, some relatively less visually-important solid-gas or gas-fluid transitions, such as sublimation and liquidation, also deserve investigating. Besides, to achieve interactive performance for more complex scenarios while guaranteeing high-fidelity simulation, we will further explore algorithmic improvements and their GPU implementation.

Acknowledgments

This research is supported in part by National Natural Science Foundation of China (No. 61190120, 61190121, 61190125, 61300067, and 61532002), Applied Basic Research Program of Qingdao (No. 161013xx) and National Science Foundation of USA (No. IIS-0949467, IIS-1047715, IIS-1049448 and IIS-1715985). We thank Yinghao Xu, Zhong Zheng and Xiao Zhai for assistances with the experiments and rendering.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.gmod.2017.09.001

References

[1] J.-M. Hong, C.-H. Kim, Discontinuous fluids, ACM Trans. Graph 24 (3) (2005) 915–920.
[2] B. Kim, Multi-phase fluid simulations using regional level sets, ACM Trans. Graph 29 (6) (2010) 175:1–175:8.
[3] P.W. Cleary, S.H. Pyo, M. Prakash, B.K. Koo, Bubbling and frothing liquids, ACM Trans. Graph 26 (3) (2007) 97.
[4] K. Iwasaki, H. Uchida, Y. Dobashi, T. Nishita, Fast particle-based visual simulation of ice melting, Comput. Graph. Forum 29 (7) (2010) 2215–2223.
[5] J. Im, H. Park, J.H. Kim, C.H. Kim, A particle-grid method for opaque ice formation, Comput. Graph. Forum 32 (2) (2013) 371–377.
[6] L. Boyd, R. Bridson, MultiFLIP for energetic two-phase fluid simulation, ACM Trans. Graph. 31 (2) (2012) 1–12.
[7] B. Solenthaler, M. Gross, Two-scale particle simulation, ACM Trans. Graph. 30 (4) (2011) 1–8.
[8] A. Peer, M. Teschner, Prescribed velocity gradients for highly viscous sph fluids with vorticity diffusion, IEEE Trans. Visual. Comput. Graph. 99 (2016).
[9] X. Yan, Y. Jiang, C. Li, R.R. Martin, S. Hu, Multiphase sph simulation for interactive fluids and solids, ACM Trans. Graph. 35 (4) (2016) 79.
[10] R. Winchenbach, H. Hochstetter, A. Kolb, Proc. acm siggraph/eurographics symp. computer animation, 2016, 10.2312/sca.20161222.
[11] M. Macklin, M. Müller, Position based fluids, ACM Trans. Graph. 32 (4) (2013) 1–12.
[12] M. Macklin, M. Müller, N. Chentanez, T. Kim, Unified particle physics for real–time applications, ACM Trans. Graph. 33 (4) (2014) 1–12.
[13] C. Wang, Q. Zhang, H. Xiao, Q. Shen, Simulation of multiple fluids with solid–liquid phase transition, Comput. Animat. Virtual Worlds 23 (3–4) (2012) 279–289.
[14] Y. Guo, X. Liu, X. Xu, A unified detail-preserving liquid simulation by two-phase lattice boltzmann modeling, IEEE Trans. Visual. Comput. Graph. 23 (5) (2016) 1479–1491.
[15] K. Raveendran, C. Wojtan, N. Thuerey, G. Turk, Blending liquids, ACM Trans. Graph. 33 (4) (2014) 137.
[16] S. Jeong, B. Solenthaler, M. Pollefeys, M. Gross, et al., Data-driven fluid simulations using regression forests, ACM Trans. Graph. 34 (6) (2015) 199.
[17] R. Ando, N. Thuerey, C. Wojtan, A stream function solver for liquid simulations, ACM Trans. Graph. 34 (2) (2015) 8.
[18] C. Jiang, C. Schroeder, A. Selle, J. Teran, A. Stomakhin, The affine particle-in-cell method, ACM Trans. Graph 34 (4) (2015) 51.
[19] F. Ferstl, R. Ando, C. Wojtan, R. Westermann, N. Thuerey, Narrow band flip for liquid simulations, Comput. Graph. Forum 35 (2) (2016) 225–232.
[20] Y. Zhu, R. Bridson, Animating sand as a fluid, ACM Trans. Graph. 24 (3) (2005) 965–972.
[21] D. Gerszewski, A.W. Bargteil, Physics-based animation of large-scale splashing liquids, ACM Trans. Graph. 32 (6) (2013) 1–6.
[22] A. Selino, M. Jones, Large and small eddies matter: animating trees in wind using coarse fluid simulation and synthetic turbulence, Comput. Graph. Forum 32 (1) (2013) 75–84.
[23] J. Cornelis, M. Ihmsen, A. Peer, M. Teschner, IISPH-FLIP for incompressible fluids, Comput. Graph. Forum 33 (2) (2014) 255–262.
[24] L. Yang, S. Li, A. Hao, H. Qin, Hybrid particle-grid modeling for multi-scale droplet/spray simulation, Comput. Graph. Forum 33 (7) (2014) 199–208.
[25] R. Ando, N. Thurey, R. Tsuruno, Preserving fluid sheets with adaptively sampled anisotropic particles, IEEE Trans. Visual. Comput. Graph. 18 (8) (2012) 1202–1214.
[26] V. Mihalef, B. Unlusu, D. Metaxas, M. Sussman, M.Y. Hussaini, Physics based boiling simulation, in: Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation, 2006, pp. 317–324.
[27] T. Kim, M. Carlson, A simple boiling module, in: Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation, 2007, pp. 27–34.
[28] S. Patkar, M. Aanjaneya, D. Karpman, R. Fedkiw, A hybrid lagrangian-eulerian formulation for bubble generation and dynamics, in: Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation, 2013, pp. 105–114.
[29] Y. Zhao, L. Wang, F. Qiu, A. Kaufman, K. Mueller, Melting and flowing in multiphase environment, Comput. Graph. 30 (4) (2006) 519–528.
[30] F. Losasso, G. Irving, E. Guendelman, R. Fedkiw, Melting and burning solids into liquids and gases, IEEE Trans. Visual. Comput. Graph. 12 (3) (2006) 343–352.
[31] Y. Chang, K. Bao, Y. Liu, J. Zhu, E. Wu, A particle-based method for viscoelastic fluids animation, in: Proc. ACM 16th ACM Symposium on Virtual Reality Software and Technology, 2009, pp. 111–117.
[32] N. Maréchal, E. Guérin, E. Galin, S. Mérillou, N. Mérillou, Heat transfer simulation for modeling realistic winter sceneries, Comput. Graph. Forum 29 (2) (2010) 449–458.
[33] J. Gagnon, E. Paquette, Procedural and interactive icicle modeling, Vis. Comput. 27 (6–8) (2011) 451–461.
[34] A. Stomakhin, C. Schroeder, C. Jiang, L. Chai, J. Teran, A. Selle, Augmented mpm for phase-change and varied materials, ACM Trans. Graph. 33 (4) (2014) 138.
[35] W. Zheng, J.-H. Yong, J.-C. Paul, Simulation of bubbles, Graph. Models 71 (6) (2009) 229–239.
[36] J. Hong, H. Lee, J. Yoon, C. Kim, Bubbles alive, ACM Trans. Graph (2008) 1–4.
[37] V. Mihalef, D. Metaxas, M. Sussman, Simulation of two-phase flow with sub-scale droplet and bubble effects, Comput. Graph. Forum 28 (2) (2009) 229–238.
[38] P. Kim, H. Lee, J. Kim, C. Kim, Controlling shapes of air bubbles in a multi-phase fluid simulation, Vis. Comput. 28 (6) (2012) 597–602.

[39] O. Busaryev, T.K. Dey, H. Wang, Z. Ren, Animating bubble interactions in a liquid foam, ACM Trans. Graph. 31 (4) (2012) 63.
[40] D. Kim, O. Song, H. Ko, A practical simulation of dispersed bubble flow, ACM Trans. Graph. 29 (4) (2010) 70.
[41] B. Ren, Y. Jiang, C. Li, M.C. Lin, A simple approach for bubble modelling from multiphase fluid simulation, Comput. Visual Media 1 (2) (2015) 171–181.
[42] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, M. Teschner, Versatile rigid-fluid coupling for incompressible sph, ACM Trans. Graph. 31 (4) (2012) 62.
[43] J. Allard, H. Courtecuisse, F. Faure, Implicit FEM and fluid coupling on gpu for interactive multiphysics simulation, in: ACM SIGGRAPH 2011 Talks, 2011, p. 1.
[44] L. Yang, S. Li, A. Hao, H. Qin, Realtime two-way coupling of meshless fluids and nonlinear fem, Comput. Graph. Forum 31 (7) (2012) 2037–2046.
[45] F. Losasso, T. Shinar, A. Selle, R. Fedkiw, Multiple interacting liquids, ACM Trans. Graph. 25 (3) (2006) 812–819.
[46] A. Robinson-Mosher, T. Shinar, J. Gretarsson, J. Su, R. Fedkiw, Two-way coupling of fluids to rigid and deformable solids and shells, ACM Trans. Graph. 27 (3) (2008) 46.
[47] A. Robinson-Mosher, R.E. English, R. Fedkiw, Accurate tangential velocities for solid fluid coupling, in: Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation, 2009, pp. 227–236.
[48] C. Batty, F. Bertails, R. Bridson, A fast variational framework for accurate solid-fluid coupling, ACM Trans. Graph. 26 (3) (2007) 100.
[49] B. Solenthaler, J. Schläfli, R. Pajarola, A unified particle model for fluid–solid interactions, Comput. Animat. Virtual Worlds 18 (1) (2007) 69–82.
[50] Y. Gao, S. Li, H. Qin, A. Hao, A novel fluid-solid coupling framework integrating flip and shape matching methods, in: Proceedings of the Computer Graphics International Conference, 2017, p. 11.
[51] D. Weber, J. Bender, M. Schnoes, A. Stork, D. Fellner, Efficient gpu data structures and methods to solve sparse linear systems in dynamics applications, Comput. Graph. Forum 32 (1) (2013) 16–26.
[52] M. Ihmsen, J. Orthmann, B. Solenthaler, A. Kolb, M. Teschner, SPH fluids in computer graphics, Eurograph. Assoc. (2014) 21–42.