Controllable Complex Freezing Dynamics Simulation on Thin Films - Supplement

YIJIE LIU*, TMCC, College of Computer Science, Nankai University, China
TAIYUAN ZHANG*, Dartmouth College, USA and TMCC, College of Computer Science, Nankai University, China
XIAOXIAO YAN, TMCC, College of Computer Science, Nankai University, China
NUOMING LIU, TMCC, College of Computer Science, Nankai University, China
BO REN†, TMCC, College of Computer Science, Nankai University, China

ACM Reference Format:
Yijie Liu, Taiyuan Zhang, Xiaoxiao Yan, Nuoming Liu, and Bo Ren. 2025. Controllable Complex Freezing Dynamics Simulation on Thin Films - Supplement. *ACM Trans. Graph.* 44, 4 (August 2025), 2 pages. https://doi.org/10.1145/3731170

This supplement provides more technical details of our freezing thin-film simulation framework. We hope this supplementary material to be self-contained and helpful for implementation.

1 COMPUTATION DETAILS OF TANGENTIAL DYNAMICS ON FREEZING THIN FILMS

In this section, we introduce our derivation of the implicit equation of the surfactant concentration Γ. We start with the freezing dynamics model:

$$\frac{\boldsymbol{u}^\top - \boldsymbol{u}^{\top *}}{\Delta t} = -\frac{2\bar{R}}{\rho\eta}(T^*\nabla_s\Gamma + \Gamma^*\nabla_s T^*) + \frac{1}{\rho}\boldsymbol{f}_{\text{ext}}^\top, \tag{1}$$

$$\frac{\Gamma - \Gamma^*}{\Delta t} = -\Gamma^*\nabla_s\cdot\boldsymbol{u}^\top, \tag{2}$$

$$\frac{\eta - \eta^*}{\Delta t} = -\eta^*\nabla_s\cdot\boldsymbol{u}^\top, \tag{3}$$

$$\frac{T - T^*}{\Delta t} = \frac{\partial T}{\partial t} - T^*\nabla_s\cdot\boldsymbol{u}^\top, \tag{4}$$

$$\frac{\zeta - \zeta^*}{\Delta t} = \frac{\partial \zeta}{\partial t} - \zeta^*\nabla_s\cdot\boldsymbol{u}^\top, \tag{5}$$

*Both authors contributed equally to the paper.
†The corresponding author.

Authors' addresses: Yijie Liu, liuyijie@mail.nankai.edu.cn, TMCC, College of Computer Science, Nankai University, Tianjin, China; Taiyuan Zhang, Taiyuan.zhang.gr@dartmouth.edu, Dartmouth College, Hanover, USA and TMCC, College of Computer Science, Nankai University, Tianjin, China; Xiaoxiao Yan, 2120230703@mail.nankai.edu.cn, TMCC, College of Computer Science, Nankai University, Tianjin, China; Nuoming Liu, liu_nm@qq.com, TMCC, College of Computer Science, Nankai University, Tianjin, China; Bo Ren, rb@nankai.edu.cn, TMCC, College of Computer Science, Nankai University, Tianjin, China.

ACM 0730-0301/2025/8-ART
https://doi.org/10.1145/3731170

The Eq.(1) can be written as:

$$\begin{aligned}
\boldsymbol{u}^\top - \boldsymbol{u}^{\top *} &= -\Delta t\frac{2\bar{R}}{\rho\eta}(T^*\nabla_s\Gamma + \Gamma^*\nabla_s T^*) + \Delta t\frac{1}{\rho}\boldsymbol{f}_{\text{ext}}^\top \\
\nabla_s\cdot\boldsymbol{u}^\top - \nabla_s\cdot\boldsymbol{u}^{\top *} &= -\Delta t\frac{2\bar{R}}{\rho}\nabla_s\cdot\frac{1}{\eta^*}(T^*\nabla_s\Gamma + \Gamma^*\nabla_s T^*) \\
&\quad + \Delta t\left(\nabla_s\frac{1}{\rho}\cdot\boldsymbol{f}_{\text{ext}}^\top + \frac{1}{\rho}\nabla_s\cdot\boldsymbol{f}_{\text{ext}}^\top\right) \\
\nabla_s\cdot\boldsymbol{u}^\top &= \nabla_s\cdot\boldsymbol{u}^{\top *} - \Delta t\frac{2\bar{R}}{\rho}\nabla_s\frac{1}{\rho}\cdot(T^*\nabla_s\Gamma + \Gamma^*\nabla_s T^*) \\
&\quad - \Delta t\frac{2\bar{R}}{\rho\eta^*}(2\nabla_s T^*\cdot\nabla_s\Gamma + T^*\nabla_s^2\Gamma + \Gamma\nabla_s^2 T^*) \\
&\quad + \Delta t\left(\nabla_s\frac{1}{\rho}\cdot\boldsymbol{f}_{\text{ext}}^\top + \frac{1}{\rho}\nabla_s\cdot\boldsymbol{f}_{\text{ext}}^\top\right).
\end{aligned} \tag{6}$$

From Eq. (2), we obtain

$$\nabla_s\cdot\boldsymbol{u}^\top = -\frac{1}{\Delta t\Gamma^*} + \frac{1}{\Delta t}. \tag{7}$$

Then we substitute Eq. (7) into Eq. (6):

$$\begin{aligned}
-\frac{1}{\Delta t\Gamma^*} + \frac{1}{\Delta t} &= \nabla_s\cdot\boldsymbol{u}^{\top *} - \Delta t\frac{2\bar{R}}{\rho}\nabla_s\frac{1}{\rho}\cdot(T^*\nabla_s\Gamma + \Gamma^*\nabla_s T^*) \\
&\quad - \Delta t\frac{2\bar{R}}{\rho\eta^*}(2\nabla_s T^*\cdot\nabla_s\Gamma + T^*\nabla_s^2\Gamma + \Gamma\nabla_s^2 T^*) \\
&\quad + \Delta t(\nabla_s\frac{1}{\rho}\cdot\boldsymbol{f}_{\text{ext}}^\top + \frac{1}{\rho}\nabla_s\cdot\boldsymbol{f}_{\text{ext}}^\top).
\end{aligned} \tag{8}$$

Finally,

$$\begin{aligned}
&(\Delta t\frac{2\bar{R}}{\rho}(\nabla_s\frac{1}{\eta^*}\cdot\nabla_s T^* + \frac{1}{\eta^*}\nabla_s^2 T^*) - \frac{1}{\Delta t\Gamma^*})\Gamma \\
&+ \Delta t\frac{2\bar{R}}{\rho}(\nabla_s\frac{1}{\eta^*}T^* + \frac{1}{\eta^*}2\nabla_s T^*)\cdot\nabla_s\Gamma + \Delta t\frac{2\bar{R}}{\rho}\frac{1}{\eta^*}T^*\nabla_s^2\Gamma \\
&= \nabla_s\cdot\boldsymbol{u}^{\top *} - \frac{1}{\Delta t} + \Delta t(\nabla_s\frac{1}{\rho}\cdot\boldsymbol{f}_{\text{ext}}^\top + \frac{1}{\rho}\nabla_s\cdot\boldsymbol{f}_{\text{ext}}^\top).
\end{aligned} \tag{9}$$

2 COMPUTATION DETAILS OF DENDRITIC CRYSTAL SIMULATION

Following Ren et al. [2018], the free energy function is given as:

$$\begin{aligned}
F = \int \frac{1}{2}\varepsilon^2|\nabla\zeta|^2 + g(\zeta) + p(\zeta)(f_s(T) + f_{ori}(|\nabla\boldsymbol{\theta}_{ori}|)) \\
+(1 - p(\zeta))f_l(T)d\mathbf{r},
\end{aligned} \tag{10}$$

Table 1. Parameters of examples.

Example	j	M_ζ	K	a	M_{ori}
Floating Bubble (Sphere Case)	6	40	3.7	0.0012	10000
Floating Bubble (Ellipsoid Case)	8	40	3.7	0.0012	10000
Swaying Dome-like Bubble	4	40	3.7	0.0012	10000
Catenoid	5	40	4.35	0.0012	10000
Symmetry Breaking	5	40	3.7	0.0012	2500
Spiral Dendritic Crystals	6	40	3.7	0.0012	10000
5-point Star-shaped Thin Film	6	80	3.0	0.0006	10000

where $f_s(T)$ and $f_l(T)$ represent free energy densities related to the temperature T. The anisotropy function ε is given as:

$$\varepsilon(\boldsymbol{\theta}) = \bar{\varepsilon}(1 + \delta cos(j(\boldsymbol{\theta}_{ori} - \boldsymbol{\theta}))), \tag{11}$$

where $\bar{\varepsilon}, \delta, j$ are constants that control the thickness of the solid-liquid interface, the strength of anisotropy, and the number of the main branches, respectively. Moreover, we use we use $g(\zeta) = \frac{1}{4}\zeta^2(\zeta-1)^2$, $p(\zeta) = \zeta^2(3-2\zeta)$, $f_l(T) = 0$, $f_s(T) = -\frac{\xi(T)}{6}$ and $f_{ori}(|\nabla\boldsymbol{\theta}_{ori}|) = J|\nabla\boldsymbol{\theta}_{ori}|$, where $\xi(T) = \frac{\alpha}{\pi}arctan(\gamma(T_e - T))$ represents the degree of temperature supercooling that drives the solidification of the liquid, and α, γ, T_e, J are constants representing the supercooling coefficient on $(0, 1)$, the thermal scaling factor and the melting temperature respectively.

The main shape control parameters and their value ranges used in our experiments are detailed below: 1) the anisotropic modulus j controls the number of main branches. In our experiments, it is set to an integer between 4 and 8. 2) M_ζ controls the growth speed, and a higher value results in faster growth, yielding smoother dendrite shapes. We set it up in the range $[40, 80]$ for our experiments. 3) K controls the branching effects. With higher K, there will be more branches in the final shape, but it also slows the growth speed. It changes from 3.0 to 3.2 in our practice. 4) a controls the thermal conduction, and higher a yields a smoother shape with fewer branch details. The appropriate value is in the range $[0.0006, 0.0012]$ in our experiments. 5) M_{ori} controls the symmetry. A higher M_{ori} results in more symmetrical patterns, while a lower one leads to stronger symmetry breaking effects. We set it between 2500 and 10000. The exact values of these parameters in our example are listed in Table 1.

3 IMPLEMENTATION DETAILS OF DISCRETIZAION

We implement the surface differential operators as:

$$\begin{cases} \nabla_s q = \sum\limits_{B\in\mathcal{N}(A)} a_B(q_B - q_A)\nabla_s\overline{W}(A,B), \\ \nabla_s\cdot\boldsymbol{w} = \sum\limits_{B\in\mathcal{N}(A)} a_B\top_B(\boldsymbol{w}_B - \boldsymbol{w}_A)\cdot\nabla_s\overline{W}(A,B), \\ \nabla_s^2 q = \sum\limits_{B\in\mathcal{N}(A)} a_B(q_B - q_A)\dfrac{2\left|\nabla_s\overline{W}(A,B)\right|}{|\boldsymbol{x}_A - \boldsymbol{x}_B|}, \end{cases} \tag{12}$$

where the indicator of particle sets is omitted since we compute the derivatives on all $\mathcal{E}, \mathcal{L}$, and $\mathcal{M}$, $W(A,B)$ is 3D kernel function of $\boldsymbol{x}_A - \boldsymbol{x}_B$, $\overline{W}(A,B)$ is 2D kernel function of $\top_A(\boldsymbol{x}_A - \boldsymbol{x}_B)$. Similar to MELP [Deng et al. 2022] , we approximate the $\nabla_s\overline{W}$ as $\mathbf{I}_{2\times2}\nabla_s\overline{W}$ to simplify the calculation and compute $\perp_A(\boldsymbol{w}) = (\boldsymbol{w}\cdot\mathbf{n}_A)\mathbf{n}_A$ and $\top_A(\boldsymbol{w}) = \boldsymbol{w} - \perp_A(\boldsymbol{w})$ for particle $A \in \mathcal{L}\cup\mathcal{E}\cup\mathcal{M}$. We use the Quintic spline kernel for both $\overline{W}$ and W, with the radius r set to $\Delta x_{\mathcal{E}}$ for discretization on $\mathcal{M}$, $\Delta x_{\mathcal{L}}$ for transfet between $\mathcal{L}$ and $\mathcal{M}$, and $4\cdot\Delta x_{\mathcal{E}}$ for $\mathcal{E}$, where $\Delta x_{\mathcal{L}}$ and $\Delta x_{\mathcal{E}}$ reflect the separation of $\mathcal{L}$ and $\mathcal{E}$. It is noteworthy that the Laplacian of the surfactant concentration Γ is calculated through $\nabla_s\cdot\nabla_s\Gamma$ rather than $\nabla_s^2\Gamma$, which is mentioned in IISPH [Ihmsen et al. 2013].

4 IMPLEMENTATION DETAILS OF FLUID-SOLID COUPLING

In the Fluid-solid Coupling step, we first employ the XSPH method (with viscosity parameter 0.1) on all $\mathcal{L}$ particles to smooth the velocity of fluid and ice crystals and enhance the stability of the coupling. Then, we use the atomic addition (the function atomicAdd provided by CUDA) to compute the total mass, momentum, centroid, and moment of inertia of each dendritic crystal in parallel. In the above process, we use connectivity calculations to determine which rigid body each solid particle belongs to. We assume that two solid particles are connected, if their distance is smaller than $1.5\cdot\Delta x_{\mathcal{L}}$. Besides, if the two solid particles belonged to different rigid bodies in the previous frame and the modulus of their relative velocity is less than a certain threshold (e.g., 0.1 times the mean value of the modal length of both velocities), we consider them to be not connected. Finally, we perform several XSPH steps (typically 5 in our examples) with the viscosity parameter of 0.999 for the tangential velocity of the fluid particles.

REFERENCES

Y. Deng, M. Wang, X. Kong, S. Xiong, Z. Xian, and B. Zhu. 2022. A Moving Eulerian-Lagrangian Particle Method for Thin Film and Foam Simulation. *ACM Trans. Graph.* 41, 4 (2022), 17 pages.

Markus Ihmsen, Jens Cornelis, Barbara Solenthaler, Christopher Horvath, and Matthias Teschner. 2013. Implicit incompressible SPH. *IEEE transactions on visualization and computer graphics* 20, 3 (2013), 426–435.

Bo Ren, Jiahui Huang, Ming C. Lin, and Shi-Min Hu. 2018. Controllable Dendritic Crystal Simulation Using Orientation Field. *Computer Graphics Forum* 37, 2 (2018), 485–495.