Snow Simulation Considering Pressure Sintering

Jiang Ye[1,,a)] Masanori Sugimoto[1,b)]

Abstract: In the real world, when snow is pressed, it becomes hard. And the higher the pressure, the harder it becomes. This phenomenon is caused by the pressure sintering of snow. Therefore, considering pressure sintering increases the reality of dynamic snow simulation. In this paper, we present a method to simulate dynamic snow considering pressure sintering. We integrate a pressure sintering model into a material point method solver. In the simulation, we first press the snow and apply pressure sintering. To observe its hardness, we then make it collide with an object and observe its collision behavior. We simulate the effect of different pressure, different snow shapes and different object shapes. Results show that, with the proposed method, the snow stays relatively soft when pressed with low pressure and more powders than lumps of snow appear after colliding.On the other hand, the snow becomes relatively hard when pressed with high pressure and more lumps of snow than powders appear after colliding.

1. Introduction

Physics-based CGs aim to simulate the behavior of natural phenomena, such as the behavior of water, and they are widely used in VFX. Snow is a common natural phenomenon. In the real world, when snow is pressed, it becomes hard. And the higher the pressure, the harder it becomes. This phenomenon is caused by the pressure sintering of snow. Therefore, considering pressure sintering increases the reality of dynamic snow simulation.

In this paper, we present a method to simulate dynamic snow considering pressure sintering. We integrate a pressure sintering model into a material point method (MPM) solver proposed by Stomakhin et al. [12]. According to Maeno [17], the pressure sintering of snow is similar to that of ceramic or metal powder, and it is driven by the internal force between snow particles and the external force exerted on them simultaneously. It causes the density and elasticity change of snow. Based on these facts, when computing density, we apply the model of Maeno and Ebinuma [7]. As Maeno and Ebinuma do not present a model to describe elasticity change, we apply the model of Sarbandi [11] when computing elasticity. We also develop an empirical model to update the plastic term in the MPM solver of Stomkahin et al. Details of the proposed method can be found in Section 3.

Simulations are conducted to verify our method. In each simulation, we first press the snow and apply pressure sintering. To observe its hardness, we then make it collide with an object and observe its collision behavior. We simulate the effect of different pressure, different snow shapes and different object shapes. Results show that, with the proposed method, the snow stays relatively soft when pressed with low pressure and more powders than lumps of snow appear after colliding. On the other hand, the snow becomes relatively hard when pressed with high pressure and more lumps of snow than powders appear after colliding. Details of the simulations and results can be found in Section 4.

We summarize our contributions as:

- An expanded MPM solver to simulate the behavior of snow considering pressure sintering.
- Combining a density model of snow with a elasticity model of ceramic to describe the density and elasticity change in the pressure sintering process.

2. Previous Work

In the physics-based simulation, we need an algorithm (also called a solver) to discretize the target material. The material point method (MPM) is a solver used in solid mechanics to simulate solid and it is introduced into Computer Graphics by Stomakhin et al. [12]. For example, Tampubolon et al. [15] develop a two-grid MPM-based framework to simulate the mixture of sand and water. They use one grid to discretize the sand and another grid to discretize the water. Other MPM-based simulation can be found in [1], [10] and [6]. In addition, [8] and [5] provides the details of MPM.

Probably the earliest work on snow simulation in Computer Graphics is by Nishita et al.[9]. They present a method to simulate static snow, for example snow-covered objects. Other work on static snow simulation, such as snow accumulation, can be found in [2] and [3]. On the other hand, several methods are proposed to simulate dynamic snow. For example, Stomakhin et al. combine a hyper-elastic model with a multiplicative plastic model to describe wet dense snow and discretize it with an MPM solver. Takahashi et al. [13] use a constant (called sintering coefficient) to describe the sintering effect of snow particles and integrate it into the smoothed particle hydrodynamics (SPH) solver. Takahashi et al. [14] define a variable (called durability) to present the air voids in a snow particle and integrate it into the fluid implicit particle (FLIP) solver.

[1] Laboratory of Intelligent Information Systems, Hokkaido University, Kita 14, Nishi 9, Kita-ku, Sapporo, 060-0814, Japan
[a)] iamyoukou@ist.hokudai.ac.jp
[b)] sugi@ist.hokudai.ac.jp

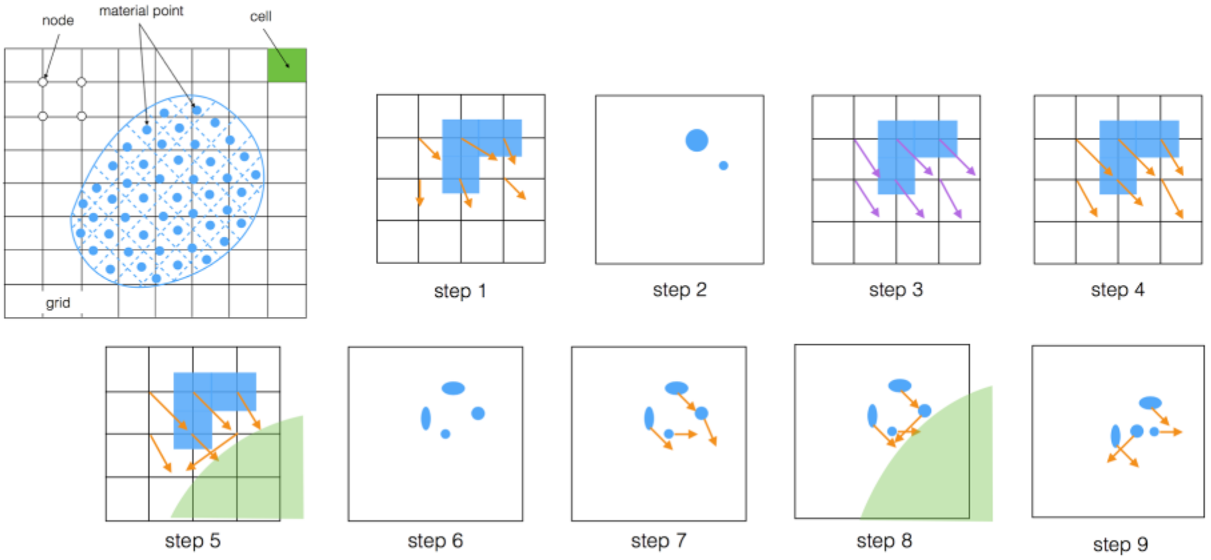

Fig. 1 The MPM solver proposed by Stomakhin et al. The orange arrow denotes the node velocity. The purple arrow denotes the node force. Blue squares denote nodes and blue ellipses denote particles. The green area in step 5 and 8 denotes an object. **step 1** computes the node velocity (orange arrow) using a weight function and the particle velocity. **step 2** computes the particle density and volume. **step 3** computes the node force. **step 4** computes the node velocity. **step 5** detects the collision between a node and and object. **step 6** computes the particle deformation. **step 7** computes the particle velocity using a weight function and the node velocity. **step 8** detects the collision between a particle and an object. **step 9** updates the particle position.

The method proposed by Stomakhin et al. can simulate a wide variety of dense and wet snow behaviors. However, as their method lacks a pressure sintering model, it can not simulate the behaviors caused by the pressure sintering of snow. Although Takahashi et al. consider the sintering effect of snow, our method differs theirs in several ways.

- We consider pressure sintering instead of sintering. While sintering merely considers the internal force between snow particles, pressure sintering also considers the external force from objects, for example the force exerted by hands when we press the snow using our hands.
- They use a constant to describe sintering, i.e. they treat sintering as a static process. However, we describe pressure sintering with the change of the density, the elasticity and the plasticity of snow, i.e. we treat pressure sintering as a dynamic process according to [17] and [11].
- We use the MPM solver instead of the SPH solver because generally, MPM is used in solid simulation but SPH is used in fluid simulation.

3. Proposed Method

Our method is based on the MPM solver proposed by Stomakhin et al. [12]. They treat snow as a hyper-elastic material and use the following constitutive relation to compute the position of a snow particle.

$$\frac{\partial \rho}{\partial t} = 0, \qquad \rho \frac{\partial \boldsymbol{v}}{\partial t} = \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g}, \qquad \boldsymbol{\sigma} = \frac{1}{J}\frac{\partial \Psi}{\partial \boldsymbol{F}_E}\boldsymbol{F}_E^T$$

where ρ is density, t is time, $\boldsymbol{v}$ is velocity, $\boldsymbol{\sigma}$ is the Cauchy stress which measures the internal force exerted on a particle, $\boldsymbol{g}$ is the gravity and Ψ is the strain energy density function. $\boldsymbol{F}_E$ is computed from $\boldsymbol{F} = \boldsymbol{F}_E\boldsymbol{F}_P$ according to the multiplicative plasticity theory. It separates deformation gradient $\boldsymbol{F}$ into an elastic part $\boldsymbol{F}_E$ and a plastic part $\boldsymbol{F}_P$. To simplify the computation of elasticity and plasticity, Stomakhin et al. develop an empirical model based on phenomenological observations. It combines elasticity (described by Lamé coefficients) with plasticity (described by hardening coefficient) and is as follows

$$\mu(\boldsymbol{F}_P) = \mu_0 \, \mathrm{e}^{\xi(1-J_P)}, \qquad \lambda(\boldsymbol{F}_P) = \lambda_0 \, \mathrm{e}^{\xi(1-J_P)} \tag{1}$$

where μ, λ and μ_0, λ_0 are the current and initial Lamé coefficients, respectively, ξ is hardening coefficient and $J_P = \det(\boldsymbol{F}_P)$ is the volume change of a particle.

According to Maeno [17], as the pressure sintering progresses, the density and elasticity of snow change. To describe the change of density, we use the model of Maeno et al. [7]. However, they do not present a model to compute the change of elasticity. To compensate this, we use the elastic model of Sarbandi [11]. Note that the model of Sarbandi is used to describe the sintering of ceramic. As Maeno [17] also states that one can treat the pressure sintering of snow as that of ceramic and pressure sintering as the sintering under a certain pressure, we consider that it is suitable to use this model. To better fit the MPM solver of Stomakhin et al., we also develop an empirical model to compute the change of plasticity. We then integrate the elastic and plastic model into equation 1. We will present the details in the following sections.

3.1 Density

Maeno et al. state that the pressure sintering process of snow consists of several stages and each stage is explained by one or more models. In total, they present six models to describe the change of density. As it is too complicated to use all the six models, we simplify them to one. Although this simplification lacks accuracy compared to its original, we found that it is effective for simulating the phenomena we desire. We use the following equations to compute density.

$$\rho^{n+1} = \rho^n + \Delta t \rho^n \left[C_1 \left(\frac{\sigma_m^n}{\rho_r^n} \right) + C_2 ||\boldsymbol{b}|| \right] \tag{2}$$

$$\sigma_m^{n+1} = \sqrt{VM_1 + VM_2} \tag{3}$$

$$VM_1 = \frac{1}{2} \left[(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{zz} - \sigma_{xx})^2 \right] \tag{4}$$

$$VM_2 = 3 \left(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2 \right) \tag{5}$$

$$\rho_r^{n+1} = \frac{\rho^{n+1}}{\rho_{max}} \tag{6}$$

where n is frame number and Δt is time step, ρ, ρ_r and ρ_{max} are current density, relative density and maximum density, respectively, σ_m is the von Mises stress computed from $\boldsymbol{\sigma}$ and σ_{ij}, $(i, j = x, y, z)$ is its component, $||\boldsymbol{b}||$ is the length of the external force vector $\boldsymbol{b}$ exerted on the particle, C_1 and C_2 are constants.

In equation 2, we use $\frac{\sigma_m}{\rho_r}$ and $||\boldsymbol{b}||$ to describe the effect of internal and external force, respectively. We found that C_1 and C_2 can make the simulation more stable. Maeno et al. use a scalar stress term in their model, but they do not provide its detail. Therefore, we use von Mises stress which convert a stress tensor into a scalar value. To compute relative density, Maeno et al. use the following equation

$$\rho_r = 1 - (d/r)^3$$

where d is the radius of pores inside snow and r is the radius of a snow particle. To avoid tracing the change of these radii, we develop equation 6. We found that this simplification works well.

3.2 Elasticity

To compute elasticity, we use the model of Sarbandi.

$$E^{n+1} = E_{max} \, e^{C_3 (1 - \rho_r^n)}, \quad \upsilon^{n+1} = \upsilon_{max} \sqrt{\frac{\rho_r^n}{3 - 2\rho_r^n}} \tag{7}$$

where E and E_{max} are the current Young's modulus and maximum Young's modulus, respectively, υ and υ_{max} are the current Poisson's ratio and maximum Poisson's ratio, respectively, and C_3 is a constant to make the simulation stable. Instead of manually specifying E_{max} and υ_{max}, we use the following strategy

$$E_{max} = \frac{E_0}{e^{C_3 (1 - \rho_r^0)}}, \quad \upsilon_{max} = \upsilon_0 \Bigg/ \sqrt{\frac{\rho_r^0}{3 - \rho_r^0}} \tag{8}$$

where E_0 and υ_0 are the initial Young's modulus and the initial Poisson's ratio, respectively. In order to use equation 1, we compute μ and λ using

$$\mu = \frac{E}{2(1 + \upsilon)}, \qquad \lambda = \frac{\upsilon E}{(1 + \upsilon)(1 - 2\upsilon)}$$

3.3 Plasticity

We develop an empirical model, based on experiments and observations, to compute the plasticity. We relate the hardening coefficient ξ in equation 1 with an internal force term σ_m and an external force term $\boldsymbol{b}$ as follows

$$\xi^{n+1} = \xi^n + C_4 \sigma_m^n + C_5 ||\boldsymbol{b}^n|| \tag{9}$$

where C_4 and C_5 are constants to make the simulation stable. Equation 9 means that the internal and external force affect the plasticity of snow in an irreversible way. As the plastic yield criteria are already defined using critical compression θ_c and critical stretch θ_s in the MPM solver of Stomakhin et al., we do not use any yield criteria here.

3.4 The expanded MPM solver

Figure 1 shows the overview of the MPM solver. To discretize the snow, it first sets a grid in the simulating space. The grid consists of cells and nodes. The snow is then discretized into material points (or particles) and the properties (e.g. position) of a particle are initialized with certain values. The proposed method is integrated between **step 3** and **step 4**. After computing the node force in **step 3**, we update density (using equations 2 to 6), elasticity (using equations 7 and 8) and plasticity (using equation 9).

4. Implementation

We use the MPM implementation provided by the open source computer graphics library *Taichi* [16]. The environment information is as follows.

- **OS:** OS X EI Captian 10.11.6.
- **Memory:** 16GB 1600MHz DDR3.
- **CPU:** 2.5 GHz Intel Core i7.

4.1 Simulation Setting

From frame 1 to frame 100, we exert a pressure (external force) on the snow and activate the pressure sintering (**Figure 4**). At this stage (pressure sintering stage), we set gravity to 0. From frame 101, we remove the external force and deactivate the pressure sintering. At this stage (falling stage), we set gravity to a certain value to let the snow fall on the floor. The snow collides with the floor and we observe its colliding behavior to verify our method. We simulate the effect of different pressure, different snow shapes and different floor shapes. **Table 1** shows the parameter setting we use and **Table 2** shows the number of particles and frame rates.

4.2 Results and Discussion

We show some of the results in **Figure 5** and **Figure 6**. In Figure 5, we use *ball* snow and *L* floor, and in Figure 6, we use *star* snow and *stair3* floor. In *row 1*, *row 2*, *row 3* and *row 4*, *row 5*, *row 6*, we use different pressure. In *row 1*, *row 2*, *row 3*, we show the results with pressure sintering. As a contrast, in *row 4*, *row 5*, *row 6*, we show the results without pressure sintering.

Comparing *row 1*, *row 2*, *row 3* with *row 4*, *row 5*, *row 6*, we can see the effect of pressure sintering. In the real world, if we press the snow, it becomes hard, and the higher the pressure, the harder it becomes. As a visual way to evaluate hardness, we can throw it against an object and observe its collision behavior. After collision, if lumps of snow appear more than powder, we consider its hardness is high. Otherwise, we consider its hardness is low. As shown in *row 1*, *row 2* and *row 3*, with pressure sintering, when the pressure increases, lumps of snow appear more than

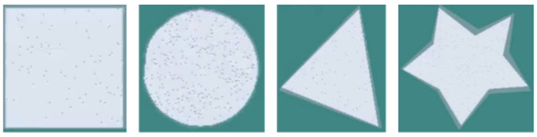

Fig. 2 Different snow shapes used in our simulations. From left to right, we name them *rectangle*, *ball*, *triangle* and *star*.

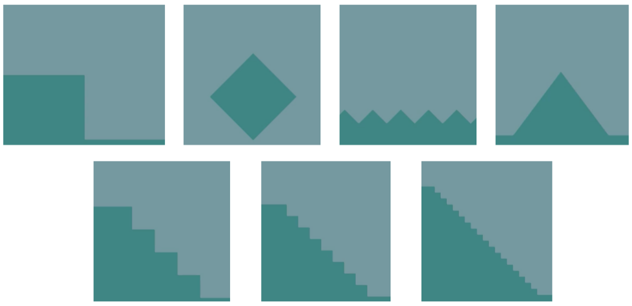

Fig. 3 Different floor shapes used in our simulations. In the first row, from left to right, we name them *L*, *square*, *zigzag* and *triangle*. In the second row, from left to right, we name them *stair1*, *stair2* and *stair3*.

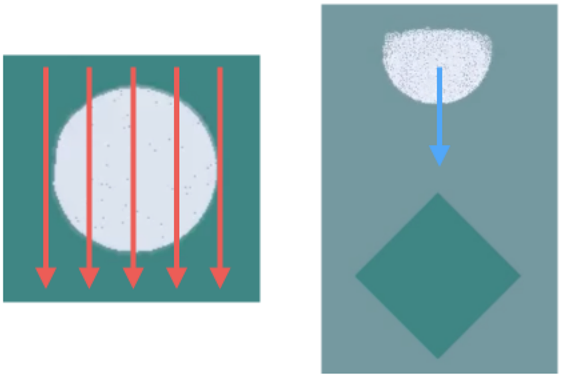

Fig. 4 Simulation setting (2D). **Left**: An external force (red arrows) is exerted on the snow and the pressure sintering is activated. **Right**: The external force is removed and the pressure sintering is deactivated. The blue arrow denotes the direction of the gravity. The dark green area denotes the floor. The white particles denote the snow.

Table 1 Parameter setting used in 2D examples. Note that examples in Figure 5 and Figure 6 use the same parameter setting.

$\rho[kg/m^3]$	$\rho_{max}[kg/m^3]$	E_0	υ_0	$m[kg]$
100	700	1.4×10^5	0.2	0.01
C_1	C_2	C_3	C_4	C_5
5×10^{-7}	10^{-3}	-9	10^{-3}	10^{-3}
θ_c	θ_s	ξ		
10^{-2}	5×10^{-3}	5		

Table 2 The number of particles and the frame rate of 2D examples.

Example	Number of particles	sec/frame
ball shape, *L* floor	8,151	0.25
star shape, *square* floor	22,076	0.51

powder after collision, This is similar to what we observe in the real world. On the other hand, as shown in *row 4*, *row 5* and *row 6*, without pressure sintering, even the pressure increases, lumps of snow do not appear and the snow always acts like powder after collision. Based on these results, we conclude that our method is able to demonstrate the hardening behavior of snow under pressure sintering.

5. Conclusion and Future Work

In this paper, we present a method to simulate the behavior of snow considering pressure sintering. We integrate a pressure sintering model into a material point method solver. Our method demonstrates the collision behavior of snow under pressure sintering. We simulate the effect of different pressure, different snow shapes and different object shapes. Results show that, with the proposed method, the snow stays relatively soft when pressed with low pressure and more powders than lumps of snow appear after colliding. On the other hand, the snow becomes relatively hard when pressed with high pressure and more lumps of snow than powders appear after colliding. This is similar to the phenomenon we observe in the real world.

As snow is porous, the hyper-elastic model of Stomakhin et al. may not be enough to describe its behavior. In the future we would like to examine a constitutive model for porous material such as sand. For simplification, we ignore several terms in the pressure sintering model of Maeno and Ebinuma. For example, we assume the temperature of the snow is constant and ignore the pores inside a snow particle. We would like to examine the effect when considering these terms. Also, in order to compare with Takahashi et al. [13], we would like to verify our method using an SPH solver.

References

[1] Daviet, G. and Bertails-Descoubes, F.: A semi-implicit material point method for the continuum simulation of granular materials, *ACM Transactions on Graphics (TOG)*, Vol. 35, No. 4, p. 102 (2016).

[2] Fearing, P.: Computer modelling of fallen snow, *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., pp. 37–46 (2000).

[3] Feldman, B. E. and O'Brien, J. F.: Modeling the accumulation of wind-driven snow, *ACM SIGGRAPH 2002 conference abstracts and applications*, ACM, pp. 218–218 (2002).

[4] Hashizume, H., Kaneko, A., Sugano, Y., Yatani, K. and Sugimoto, M.: Fast and accurate positioning technique using ultrasonic phase accordance method, *TENCON 2005 2005 IEEE Region 10*, IEEE, pp. 1–6 (2005).

[5] Jiang, C., Schroeder, C., Teran, J., Stomakhin, A. and Selle, A.: The material point method for simulating continuum materials, *ACM SIGGRAPH 2016 Courses*, ACM, p. 24 (2016).

[6] Klar, G., Gast, T., Pradhana, A., Fu, C., Schroeder, C., Jiang, C. and Teran, J.: Drucker-prager elastoplasticity for sand animation, *ACM Transactions on Graphics (TOG)*, Vol. 35, No. 4, p. 103 (2016).

[7] Maeno, N. and Ebinuma, T.: Pressure sintering of ice and its implication to the densification of snow at polar glaciers and ice sheets, *The Journal of Physical Chemistry*, Vol. 87, No. 21, pp. 4103–4110 (1983).

[8] Nguyen, V. P. and Unitversity, C.: Material point method: basics and applications (2014).

[9] Nishita, T., Iwasaki, H., Dobashi, Y. and Nakamae, E.: A modeling and rendering method for snow by using metaballs, *Computer Graphics Forum*, Vol. 16, No. 3, Wiley Online Library (1997).

[10] Ram, D., Gast, T., Jiang, C., Schroeder, C., Stomakhin, A., Teran, J. and Kavehpour, P.: A material point method for viscoelastic fluids, foams and sponges, *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, pp. 157–163 (2015).

[11] Sarbandi, B.: Finite element simulation of ceramic deformation during sintering, PhD Thesis, Ecole Nationale Supérieure des Mines de Paris (2011).

[12] Stomakhin, A., Schroeder, C., Chai, L., Teran, J. and Selle, A.: A material point method for snow simulation, *ACM Transactions on Graphics (TOG)*, Vol. 32, No. 4, p. 102 (2013).

[13] Takahashi, T. and Fujishiro, I.: Particle-based simulation of snow trampling taking sintering effect into account, *ACM SIGGRAPH 2012*

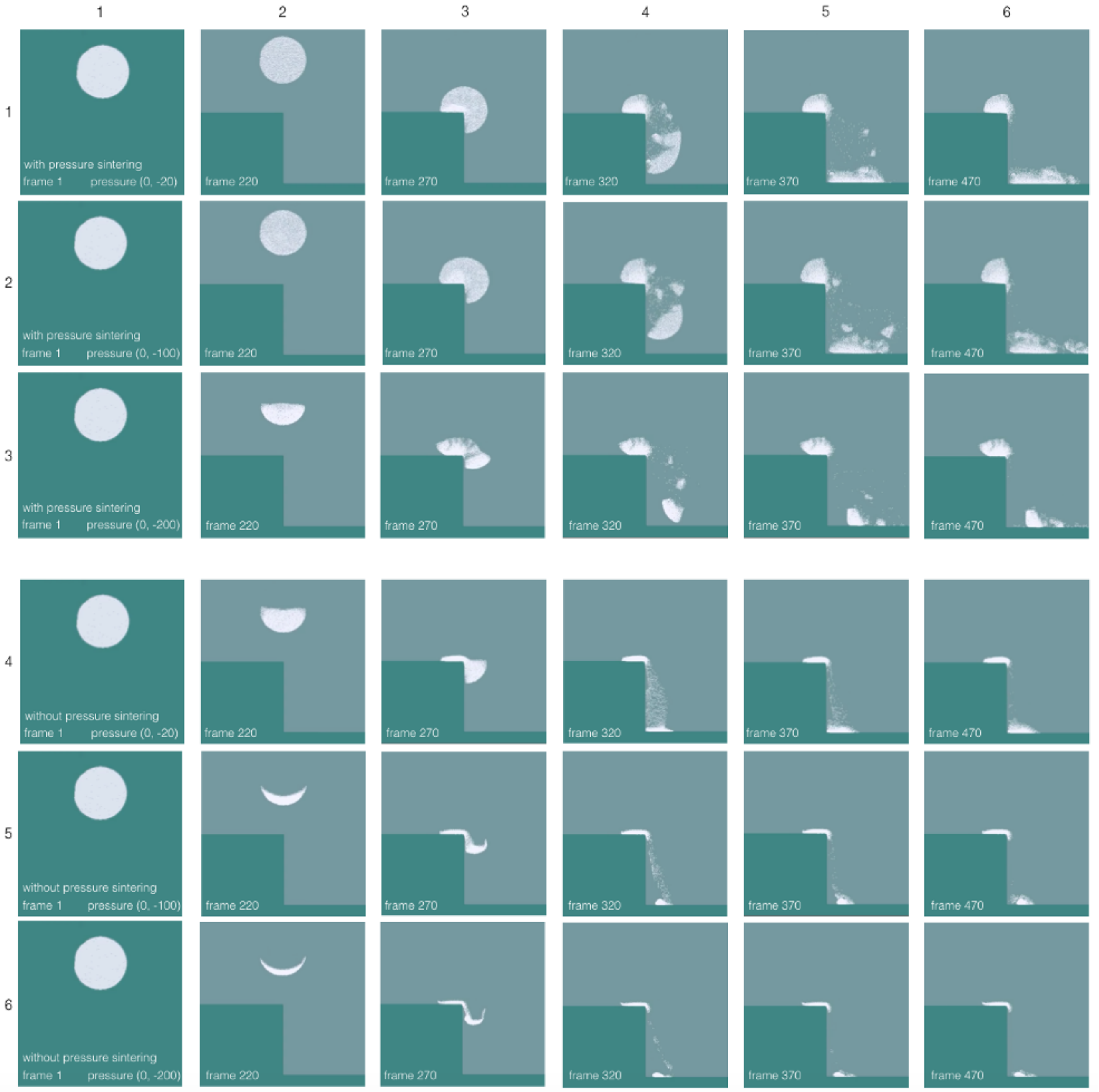

Fig. 5 2D example of *ball* snow and *L* floor.

Posters, ACM, p. 7 (2012).

[14] Takahashi, T., Fujishiro, I. and Nishita, T.: Visual Simulation of Compressible Snow with Friction and Cohesion.

[15] Tampubolon, A. P., Gast, T., Klár, G., Fu, C., Teran, J., Jiang, C. and Museth, K.: Multi-species simulation of porous sand and water mixtures, *ACM Transactions on Graphics (TOG)*, Vol. 36, No. 4, p. 105 (2017).

[16] Yuanming, H. and Yu, F.: Taichi Open Source Computer Graphics Library, `http://taichi.graphics/`.

[17] 前野紀一: 雪氷の構造と物性, **基礎雪氷学講座** (1986).

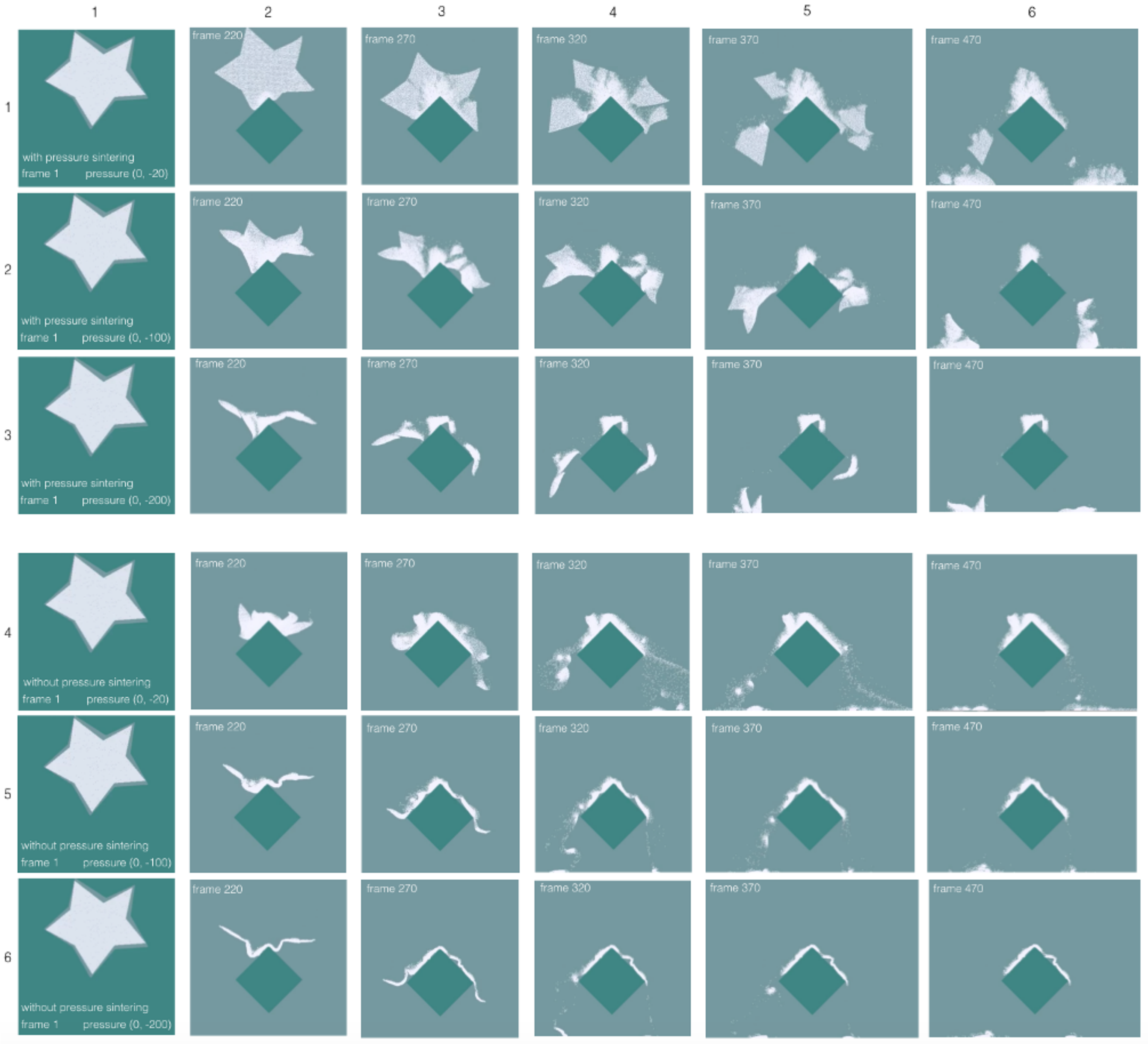
1
2
3
4
5
6
1
2
3
4
5
6
with pressure sintering
frame 1 pressure (0, -20)
with pressure sintering
frame 1 pressure (0, -100)
with pressure sintering
frame 1 pressure (0, -200)
without pressure sintering
frame 1 pressure (0, -20)
without pressure sintering
frame 1 pressure (0, -100)
without pressure sintering
frame 1 pressure (0, -200)
frame 220
frame 270
frame 320
frame 370
frame 470

Fig. 6 2D example of *star* snow and *stair3* floor.