The Application of Simplified Strassen Algorithm to Snow Simulation with Material Point Method

Mingrui Wang

St. Olaf College, Northfield, MN, US, 55057, sueraywang@gmail.com

Abstract. As a beautiful yet complicated material, snow and its simulations are significantly challenging in the computer graphics community. Since the introduction of the material point method, abbreviated as MPM, the performance of snow simulations has been augmented in many aspects, including accuracy, stability, and efficiency. Nowadays, researchers work on developing simpler mathematical models or incorporating better hardware design to accelerate the snow simulations, but the overall running speed remains a problem. Instead of improving on the simulation methods, this paper presents a novel approach to optimize the efficiency of snow simulations by applying the simplified Strassen algorithm, an abridgment of a famous fast matrix multiplication algorithm, to snow simulations implemented with MPM. Rather than straightforwardly showing the triumph of the simplified Strassen algorithm over the standard matrix multiplication algorithm in runtime, the results present the evidence that leads to the inference that for simulations with a certain level of disorder, the simplified Strassen algorithm will surpass the standard matrix multiplication algorithm as the simulations progress.

CCS CONCEPTS • Computing methodologies • Computer graphics • Animation • Physical simulation

Keywords: Computer Graphics, Snow Simulation, Physically-based Simulation, Material Point Method, Fast Matrix Multiplication Algorithm

1 Introduction

Simulating snow aesthetically and efficiently is challenging but significant in the computer graphics community. To achieve results with satisfying visual quality and physical accuracy, Stomakhin et al. treated snow as an elastoplastic material and proposed the material point method, abbreviated as MPM, to promote the simulations [1]. For the first time, snow simulations are implemented with one efficient method, and the pain of requiring multiple solvers, which makes snow simulations hard and slow, is alleviated. Since then, many researchers have kept working on augmenting the MPM or applying other methods like Smoothed Particle Hydrodynamics (SPH) to obtain more efficient results [2-3]. Nevertheless, due to the enormous amount of operations involved, the runtime of snow simulations remains too long to make real-time snow simulations visually appealing and physically satisfying. Even though many previous works have presented snow simulations in real-time applications with the help of GPUs or rendering techniques, the application of computational algorithms to snow simulations for optimizing running speed is rarely seen [4-5]. Therefore, this paper presents a main contribution by presenting evidence and inferring that lightly disordered snow simulations implemented with MPM using the simplified Strassen algorithm, a simplification of a well-known fast matrix multiplication algorithm, will be more efficient in runtime than those using the standard matrix multiplication algorithm as the simulations proceed. The paper is organized as follows: Section 2 discusses the related work. Section 3 delineates the methodology and the simplified Strassen algorithm. Section 4 provides detailed information on the setups of the experiments and snow simulations. Section 5 presents the collected data. Then, this paper closes with a discussion of the results and conclusion in sections 6 and 7, respectively.

2 Related Work and Problem Formulation

2.1 Fast Matrix Multiplication Algorithm

Matrices are essential in digital signal processing. Therefore, the design and application of faster matrix multiplication algorithms is significant. The time complexity of the standard matrix multiplication algorithm, as shown in equation (1), is $O(n^3)$.

$$M_{i,j} = \sum_{k=1}^{n} a_{i,k} * b_{k,j} \quad (1)$$

That is to say, given two n×n matrices, this algorithm eventually yields n3 scalar multiplications such that its runtime increases as the cube of the given matrices' size n. For a long time, this algorithm was considered the best result people could derive. Nonetheless, in 1969, Volker Strassen devised a clever method of dividing the input matrices into four quadrants and then recursively continuing this process until the sub-matrices are 2×2 [6]. Then, Strassen gave a set of relationships that replaces those eight scalar multiplications of the normal 2×2 matrix multiplication algorithm with seven scalar multiplications plus fourteen extra scalar additions. Finally, the products of all 2×2 matrix multiplications are reassembled into the final result of the original problem. By saving a single scalar multiplication of each sub-problem, Strassen improved the theoretical performance of matrix multiplication from n^3 to $n^{2.81}$ multiplicative steps.

2.2 Snow Simulation with Material Point Method

In 2013, Stomakhin et al. developed an elastoplastic model, which considers snow as an elastic material while the stress is below the threshold and lets it undergo plastic deformation otherwise [1]. Then, they apply MPM to run the simulations. As a hybrid method that takes advantage of Lagrangian material particles and Eulerian Cartesian grids, MPM helps to apply natural treatments of topological deformations and collisions in the simulations [7]. On the one hand, it avoids the need to model every snow grain when handling plasticity and fracture compared with the Lagrangian paradigm. On the other hand, it outperforms Eulerian methods by tracking conservative quantities through Lagrangian particles. Even though the result sacrifices some physical accuracy for engineering purposes due to their simple model with one constitutive relation, it works well for generating realistic snow dynamics with satisfactory performance.

2.3 Problem Formulation

Given the above references, this paper proposes applying the Strassen Algorithm to the snow simulations implemented with MPM to speed up the simulation. Runtime comparison experiments are designed and conducted to make a quantitative analysis of the effects of Strassen Algorithm on snow simulations implemented with MPM.

3 Methodology

To compare and contrast the efficiency of two-dimensional snow simulations implemented with MPM that uses different matrix multiplication algorithms, this paper conducts simulations that animate the process of two snowballs approaching one another and then smashing (Figure 2B) with the same setup but using the standard matrix multiplication algorithm and simplified Strassen algorithm, respectively. The same simulation runs ten times with each algorithm, and the average runtime (in seconds) is recorded. There are four distinct setups. They differ by varying the number of particles involved in the simulations, which have a linear correlation to the computational load and hence affect the runtime of simulations. Furthermore, simulations of two other scenes are carried out to examine the relationships between the scenes' complexity and the simulations' runtime.

3.1 Simplified Strassen Algorithm

The Strassen algorithm is redeemed as the optimal and practical solution to the general smaller matrix multiplications [8]. This research takes it as the starting point to accelerate the snow simulations. The simulations in the experiments are two-dimensional such that only 2×2 matrices are involved, wherefore this paper proposes the simplified Strassen algorithm. It removes the divide-and-conquer approach in the traditional Strassen algorithm and uses seven scalar multiplication and eighteen scalar additions when multiplying two matrices. The flowchart is shown in Figure 1. By doing so, the simplified Strassen algorithm replaces one scalar multiplicative operation with fourteen additive operations compared to the standard matrix multiplication algorithm. This replacement increases the runtime of single matrix multiplication. Analogous to the runtime optimization endowed by the Strassen algorithm using Master theorem, however, it is expected that the overall runtime of all matrix multiplications will decrease, provided that the number of matrix multiplications is substantial. Moreover, the simplified Strassen algorithm eliminates the need for nested loops while implementing the standard matrix multiplication algorithm, which should save some extra operations.

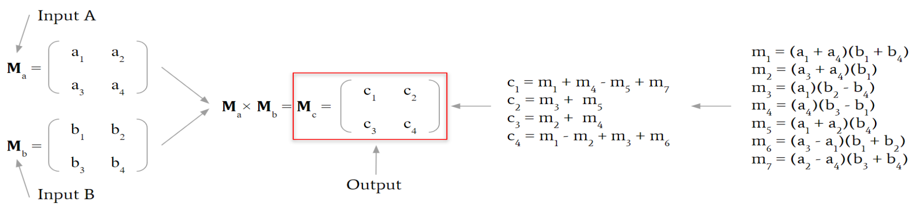

Figure 1: Simplified Strassen Algorithm

4 Experiment

4.1 Experiment Setup

As mentioned in Section 3, three scenes are to be simulated.

1. A single snowball falls with a downward initial speed = 20 m/s (Figure 2A),
2. Two snowballs approach one another with each's speed = 90 m/s then smash (Figure 2B),
3. Two snowballs crash with a still snowman with each's speed = 90 m/s (Figure 2C).

The number of particles in Scene 1 and Scene 3 are 6080 and 6180, respectively, which are about the same as setup 1 of Scene 2. For Scene 2 specifically, there are four setups:

1. Number of particles = 5540,
2. Number of particles = 27361, about 5 times as setup 1,
3. Number of particles = 55487, about 10 times as setup 1,
4. Number of particles = 138518, about 25 times as setup 1.

The setups 1 is determined by setting up the diameter of particles in the simulations to the default number suggested by Stomakhin. All other setups are determined by setting the particle diameter to finite numbers closest to those that reveal the linear and quadratic relationship between runtime and computational load.

Then, for every scene or setup, the simulation first runs for 100 frames, and the average runtime is recorded. Next, the simulation duration is extended with an increment of 100 frames. The exact process is repeated until the simulation's duration reaches 500 frames. The upper bound of 500 frames is determined because it is the point where the simulations of Scene 2 start to get very unstable such that taking any frames after that is regarded as meaningless. For more details concerning each scene or setup, please refer to the implementation [9]. The experiments were conducted on a computer with a 12th Gen i7-12700KF 3.60 GHz Intel processor, 16GB RAM, and Windows 11.

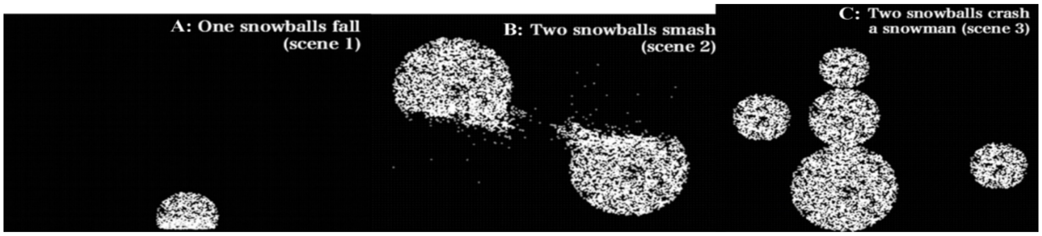

Figure 2: Experimental scenes.

2A: Scene 1: A single snowball falls; 2B: Scene 2: Two snowballs approach each other and smash; 2C: Scene 3: Two snowballs approach a still snowman and crash with it.

4.2 Snow Simulation

The snow simulation is two-dimensional, implemented with MPM, according to the paper by Stomakhin et al[1]. The original snow simulation is serially implemented on CPUs and thus could perform better [10]. Despite that, this implementation has poor stability when the scenes get more disordered due to unknown issues. Consequently, to successfully run the experiment, the final implementation is adapted from some open-source implementations of complete elastoplastic MPM snow simulations [9]. The configuration and parameters are slightly modified based on those recommended values from Stomakhin et al., which can be found in the code [1][9]. Especially noticeable is that the density of snowballs in the simulation was set lower than that suggested by Stomakhin et al. on purpose [1]. It makes the scene lightly disordered at the end of simulations.

5 Results

This research has conducted various simulations, demonstrating that the running speed of the snow simulations implemented by MPM with the simplified Strassen algorithm converges more quickly than those with the standard matrix multiplication algorithm. In particular, this paper conducts simulations of the same scene with different setups, aiming to illustrate the relationship between runtime and computational load. The results are shown in Figure 3 - 6.

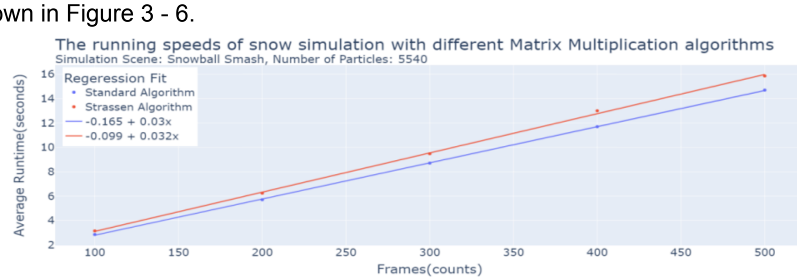

Figure 3: Runtimes of snow simulations of Scene 2 with 5540 particles

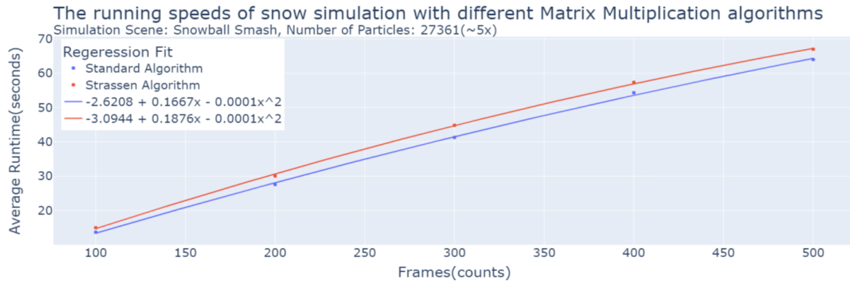

Figure 4: Runtimes of snow simulations of Scene 2 with 27361 particles

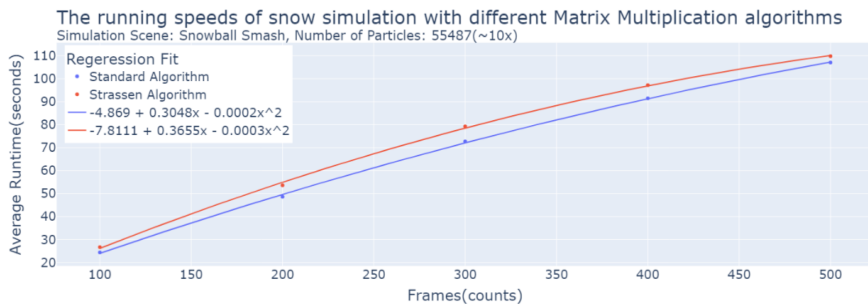

Figure 5: Runtimes of snow simulations of Scene 2 with 55487 particles

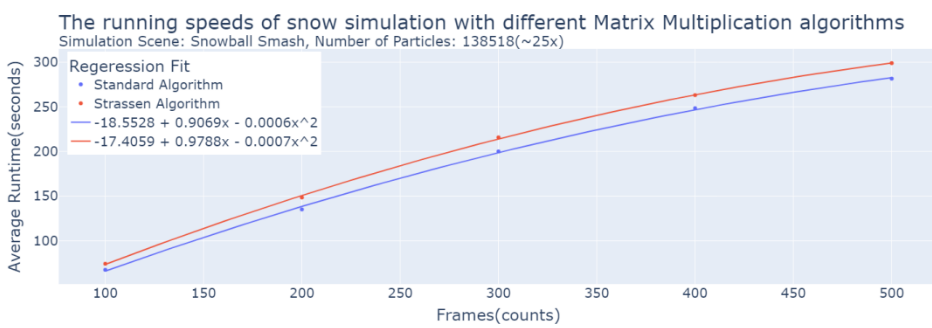

Figure 6: Runtimes of snow simulations of Scene 2 with 138518 particles

To better show the trend, this paper also plots the runtime difference between every increment of 100 frames, i.e., the rate of runtime change in Figure 7-10.

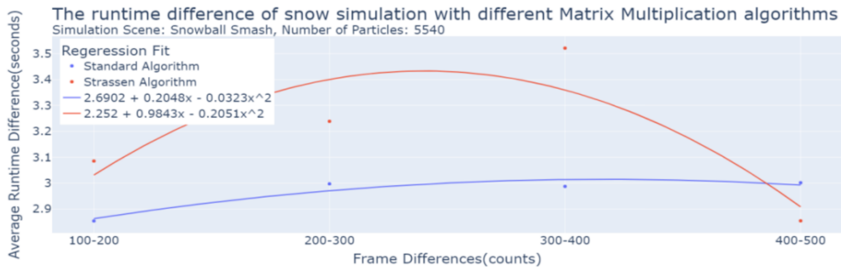

Figure 7: Runtime differences of snow simulations of Scene 2 with 5540 particles

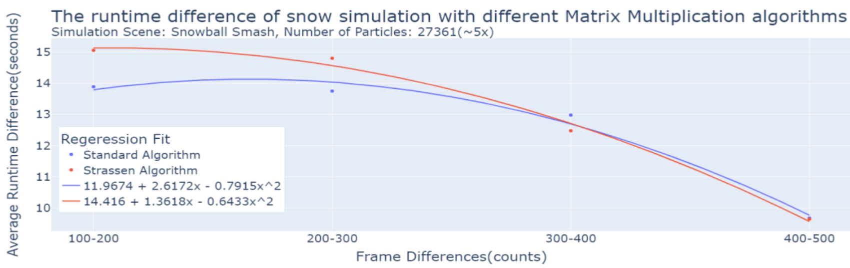

Figure 8: Runtime differences of snow simulations of Scene 2 with 27361 particles

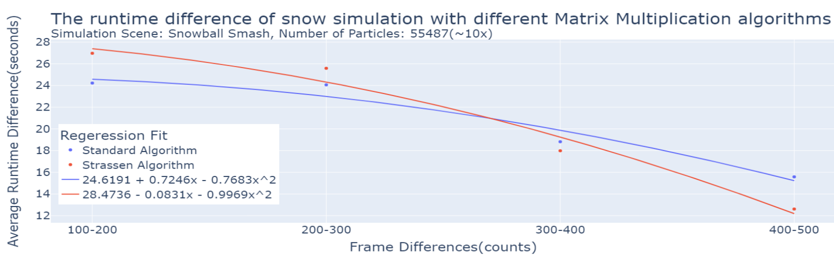

Figure 9: Runtime differences of snow simulations of Scene 2 with 55487 particles

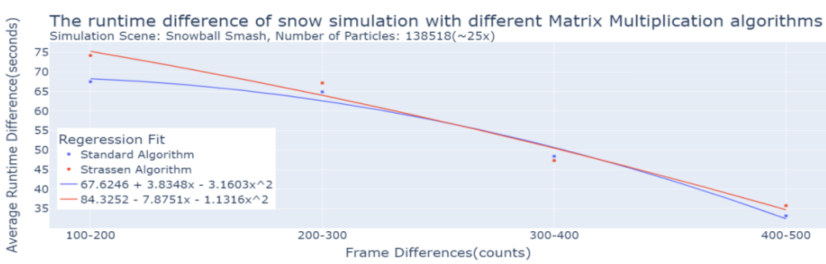

Figure 10: Runtime differences of snow simulations of Scene 2 with 138518 particles

Next, this research presents the runtimes/runtime differences of simulations of the other two scenes to illustrate the potential relationship between runtime/convergence and the level of disorder in Figure 11-14.

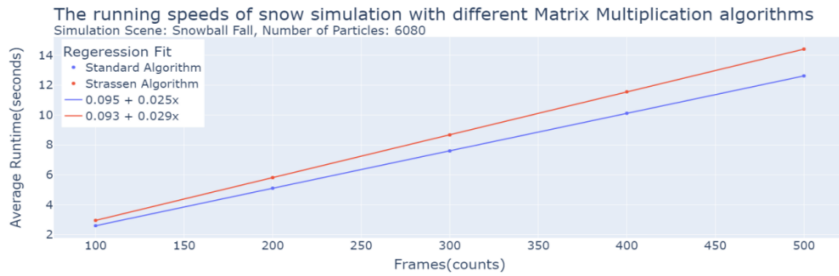

Figure 11: Runtimes of snow simulations of Scene 1 with 6080 particles

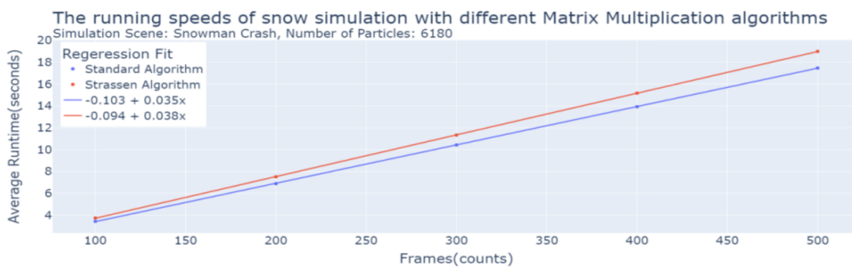

Figure 12: Runtimes of snow simulations of Scene 3 with 6180 particles

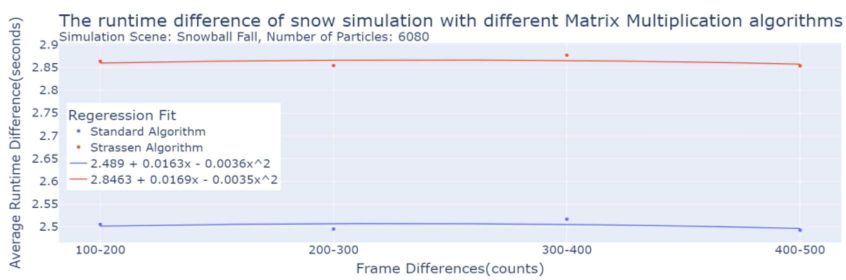

Figure 13: Runtimes differences of snow simulations of Scene 1 with 6080 particles

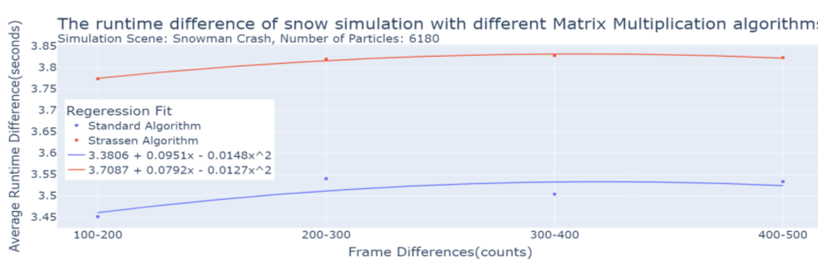

Figure 14: Runtimes differences of snow simulations of Scene 3 with 6180 particles

6 Discussion and Limitations

According to Figure 3, although the runtimes of the simulations implemented with the simplified Strassen algorithm are longer than those implemented with the standard matrix multiplication algorithm, the trend is not linear. As the number of particles increases, the runtimes of the simulations implemented with the simplified Strassen algorithm tend to converge, and the tendency of convergence becomes more and more significant. Figure 4 shows that the intersection of two derivatives is smaller as the number of particles increases. It means that the point where simulations implemented with the simplified Strassen algorithm converge faster than the other comes earlier. Therefore, it is reasonable to infer that if the simulations run longer, the runtime of those simulations driven by the simplified Strassen algorithm will eventually be smaller than those driven by the standard matrix multiplication algorithm. Simultaneously, the number of particles, or the computational load, determines how early the transcendence will occur. Yet, as mentioned in Section 4.1, the simulation must terminate at 500 frames. So, Scene 1 and Scene 3 are constructed, where the simulations stay stable even after 500 frames have elapsed. According to Figure 5, the pattern observed in Figures 3 and 4 disappears. The fitting line is linear, and the regression of the derivatives has no sign of intersection. For this reason, the research hypothesizes that a certain level of disorder is required for the power of the simplified Strassen algorithm to be evident. Only when the simulations are lightly disordered simplified Strassen algorithm will surpass the standard matrix multiplication algorithm as the simulations progress. The computational load positively correlates with the time when the transcendence occurs. The scene's complexity has no significant impact on the simulations' runtimes, given that the number of particles in the simulation is about the same for nonidentical scenes.

The data collected from this experiment have generated compelling evidence for inference, but much more work remains unfinished. First, the data is far from ample, so the regressions are biased. It might be better to increment 10 frames instead of 100 frames. Aside from that, the duration of this simulation could be extended for more comprehensive data to be collected, or a wider range of scenes could be constructed for data collection. Currently, the number of comparable scenes is limited, leading to a less convincing contrast. A clever choice of scenes that start with good stability and gradually become disordered is problematic, as many parameters must be tuned by hand. Developing a method that adjusts the parameters would be exciting.

7 Conclusion

This paper has presented the results of applying the simplified Strassen algorithm to the snow simulations implemented with MPM. The results imply that for the simulations with a certain level of disorder, the simplified Strassen algorithm will be more efficient in runtime than the standard matrix multiplication algorithm as the duration of the simulations gets longer, provided that the scene of the simulations is adequately chosen. The results also show some positive correlations between the number of particles involved in the same simulation and the time when the Strassen algorithm will surpass the standard one. In particular, the more particles required, the earlier transcendence will occur. As such, this paper contributes with a discussion of a novel approach to accelerate the snow simulation implemented by MPM by applying the simplified Strassen algorithm. Simultaneously, this paper needs more ample, comprehensive data to make the results less straightforward. The triumph of the simplified Strassen algorithm over the standard one in runtime is predictive. The potential extension of this paper could lie in the larger amount of data collection under different scenarios and the presentation of better quantitative analysis and contrast of the results.

ACKNOWLEDGMENTS

I, the author of this thesis, would like to show my deepest gratitude to Dr. Eitan Grinspun, professor at the University of Toronto, for providing me with valuable guidance in implementing the snow simulation with MPM. Further, I would like to thank all my professors, friends, and parents for their encouragement and support. Lastly, I also thank the anonymous reviewers for their suggestions.

REFERENCES

[1] Stomakhin A, Schroeder C, Chai L, Teran J, and Selle A. (2013) A Material Point Method for Snow Simulation. ACM Trans. Graph. 32.

[2] Stomakhin A, Schroeder C, Jiang C, Chai L, Teran J, and Selle A. (2014) Augmented MPM for Phase-Change and Varied Materials. ACM Trans. Graph. 33.

[3] Gissler C, Henne A, Band S, Peer A, and Teschner M. (2020) An Implicit Compressible SPH Solver for Snow Simulation. ACM Trans. Graph. 39.

[4] Junker A and Palamas G. (2020) Real-Time Interactive Snow Simulation Using Compute Shaders in Digital Environments. In: Proceedings of the 15th International Conference on the Foundations of Digital Games (FDG '20). New York. Article 70, pp. 1–4.

[5] Goswami P, Markowicz C, and Hassan A. (2019) Real-time Particle-based Snow Simulation on the GPU. In: Eurographics Symposium on Parallel Graphics and Visualization. The Eurographics Association. pp. 49–57.

[6] V. Strassen. (1981) The computational complexity of continued fractions. In: Proceedings of the fourth ACM symposium on Symbolic and algebraic computation. New York, pp. 51–67.

[7] Jiang C, Schroeder C, Teran J, Stomakhin A, and Selle A. (2016) The Material Point Method for Simulating Continuum Materials. In ACM SIGGRAPH 2016 Courses (SIGGRAPH'16). New York. Article 24, pp. 1–52

[8] Fawzi, A., Balog, M., Huang, A., Hubert, T., Romera-Paredes, B., Barekatain, M., Novikov, A., R. Ruiz, F. J., Schrittwieser, J., Swirszcz, G., Silver, D., Hassabis, D., & Kohli, P. (2022) Discovering faster matrix multiplication algorithms with reinforcement learning. Nature, 610(7930), pp. 47–53.

[9] Mingrui Wang. (2023) snow_modified, https://github.com/sueraywang/snow_modified.

[10] Kodai Takao. (2018) OpenGL_MPMSnowSimulation2D, https://github.com/kodai100/OpenGL_MPMSnowSimulation2D.