ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2022
D. L. Michels and S. Pirk
(Guest Editors)

*Volume 41* (*2022*), *Number 8*

# CSC 473 Project - Dynamic Cloth Simulation: Utilizing Spring-Mass Systems

Churong Yu

Figure 1: Fabric Simulation Render with Boucle Bubbly Rows Texture.

**Abstract**

*This paper introduces a dynamic cloth simulation system developed using a spring-mass model based on the Blender 3D software. The focus of this project is on simulating basic cloth behaviors by manage stretch and shear springs to model the interactions between particles of the fabric. Each particle is considered a point mass, influenced by forces derived from a modified version of Hooke's law, which now includes damping and external forces such as gravity. To prevent the effect of superelasticity, the system has a mechanism to control spring stiffness depending on the rate of deformation. This method avoids unnecessary stretching of the cloth by enabling dynamic stiffness control.*

**CCS Concepts**

• ***Computing methodologies*** $\rightarrow$ ***Physical simulation;*** *Mass spring systems;*

## 1. Introduction

Simulating the behavior of fabric in a three-dimensional environment is a significant problem in computer graphics, particularly in physical modeling. The primary objective of this project is to simulate the behavior of lightweight textiles using a mass-spring system, a popular technique for simulating flexible materials in computer graphics. The primary goal of this study was to create realistic simulations of fabric hanging under gravity with defined places to reflect attachments or constraints typical in real-world scenarios, such as curtains or clothes on a hanger.

The biggest challenge in cloth simulation is the computational complexity and physical accuracy of modeling cloth dynamics using mass-spring systems. Such systems need to have their physical characteristics, such as mass distribution, spring stiffness, and damping factors, precisely adjusted to avoid unrealistic reactions. In Blender, a comprehensive 3D modeling and rendering software, implementing a custom mass-spring system posed significant technical challenges. These challenges were compounded by the need to seamlessly integrate this solution into Blender's existing architecture.

Controlling the springs' physical behavior was one of the difficulties. Compared to many fundamental spring models, which have high elasticity, typical textile materials have extremely low elasticity. This difference frequently results in simulations where fabric acts more like rubber elastic than actual fabric, constantly stretching and unable to stabilize into a stable shape. Early system trials produced simulations in which the mesh would erroneously "flow" or "explode," demonstrating how sensitive the simulation was to certain parameter values.

To address these challenges, this project adopted a sympathetic Euler method for updating particle states at each time step, dynamically adjusting spring stiffness in response to the fabric's deformation. This method was implemented based on established techniques and insights drawn from multiple academic papers, reflecting an understanding of existing knowledge.

The two main goals of this study were to improve one's comprehension of the mathematical and physical concepts that underlie cloth simulation and to acquire useful skills for putting these simulations into practice using a real-world program like Blender. The original intention was to imitate cloth as closely as possible; however, the difficulties faced during the process resulted in a less ambitious result. Nevertheless, the project succeeded in creating a basic simulation that could model the draping and hanging of cloth, providing a foundation for further exploration and refinement in future work.

## 2. Related Work

Within computer graphics, a lot of attention has been paid to the development of cloth simulation techniques, with the goal of achieving both computing efficiency and realism. Baraff et al. [BW23] describe in "Large Steps in Cloth Simulation" one of the fundamental methods for improving the realism of cloth simulations. Unlike conventional vertex and mass-spring systems, their approach incorporates a complicated system of particles and internal forces to approximate fabric structures. The approach mainly uses implicit integrals and offers strong collision management based on the application of scalar potentials. This technique has very high simulation accuracy, but it is too difficult and time-consuming to execute fully for this project; as a result, the more conventional mass-spring system is still used.

Among the limited studies focusing on Blender for physics simulations, the work by Orosz et al. stands out for its in-depth examination of tearable cloth [OU]. Orosz et al. thoroughly examine the challenges of simulating tearable cloth within the Blender platform. Their research overcame the difficulty of manipulating Blender's existing tools and cloth structure to create dynamic and realistic tearing effects. Orosz et al developed a visualization system to observe the forces acting on cloth springs and implemented a mechanism that allows the cloth to tear when stretched beyond a predefined force threshold. Their work significantly enhanced my understanding and capabilities of implementing cloth simulation in Blender. It also highlighted Blender's limitations for autonomously implementing simulation algorithms, suggesting that future research might better leverage open-source development platforms like Unity. However, for this project, I will continue using Blender. The insights and experiences shared by Orosz et al. in using Blender for cloth tearing were highly inspirational and have greatly assisted in setting up my framework for mass-spring construction and implementation within Blender.

The chapter "Mass-Spring Models" in Stuyck's book [Stu18] "Cloth Simulation for Computer Graphics" covers the detail of the use of mass-spring systems for simulating cloth dynamics in computer graphics. This chapter is invaluable for newcomers, as it outlines the fundamental concepts and mathematical foundations necessary for realistic cloth simulation. Stuyck introduces the standard structure of mass-spring models, where each mass is linked by three types of springs: structural, bend, and shear springs. Structural springs connect immediate neighbors, bend springs link alternate particles, and shear springs, which connect diagonally across particles, activate under compression. These springs play specific roles in defining the cloth's behavior, from maintaining shape to adding flexibility and resistance to bending. The chapter also dives deeper into the mathematical aspects of these models, applying Hooke's Law to define the forces within the springs and integrating damping to model energy dissipation. Stuyck's discussion on the importance of selecting appropriate stiffness constants for different types of springs is particularly crucial, emphasizing that the realism of simulations largely depends on these values. For instance, most materials typically have lower resistance to shearing, so shear and bend springs should generally have lower stiffness constants compared to stretch springs. This concept significantly influenced the parameter testing in my project.

Assunção [dFA20] presents "Cloth Simulation Framework," a cloth simulation system built inside the Unity engine with an emphasis on employing the mass-spring model to replicate realistic cloth dynamics in a dynamic setting. In order to solve the equations of motion for cloth particles, the core of the framework involves a thorough investigation of many numerical integration techniques with the goal of achieving the best possible balance between stability, accuracy, and processing overhead. This feature has provided me with a lot of valuable information for my project. His thorough analysis of various physical algorithms and their stability has had a significant impact. Assunço's meticulous technique in Unity motivated me to incorporate analogous numerical techniques in Blender, so augmenting the steadiness and precision of my system.

In "Study and Comparison Techniques in Fabric Simulation Using Mass Spring Model," Mozafary et al. explore various techniques for simulating fabric using the mass-spring model, emphasizing configurations and physical properties such as elasticity, damping, and collision [MP16]. They assessed different mesh types within the mass-spring model, discussing the impact of structured versus unstructured grids on simulation accuracy and computational demands. Their comparative analysis provides valuable insights into how different spring arrangements can influence cloth behavior and appearance. Given the complexity of implementing unstructured grids, this project will consider only rectangular meshes. Moreover, Mozafary et al. delved into the challenge of super elastic effects, where fabrics exhibit unrealistic stretchability under force. They discussed several strategies to address this issue, such as adjusting spring stiffness based on the forces applied and incorporating advanced damping mechanisms to better simulate the natural behavior of various fabric materials. The paper also compared numerical integration methods for solving the equations of motion in mass-spring systems, critically evaluating the trade-offs between stability, accuracy, and computational efficiency offered by explicit and implicit methods. Their thorough exploration of these topics significantly informed the development of my simulation framework, providing a foundation for selecting and implementing the most appropriate techniques for realistic cloth simulation. The insights from Mozafary et al. on managing super elastic effects have been particularly beneficial in addressing the challenges encountered in my project. This paper was super helpful in reducing the time spent searching for academic articles as it had already collected and compared statistics on different methods. It provided comprehensive information on super elastic effects and their resolution methods, making each method easy to understand and allowing me to easily access referenced articles, thus smoothly addressing the super elasticity issues encountered in my project.

In their paper "Practical and Realistic Animation of Cloth," Bayraktar et al. present an effective framework for cloth simulation that utilizes a mass-spring model enhanced by explicit time integration methods to solve motion equations [BGO07]. Their new approach involves dynamically updating the spring constants based on the net forces acting on the springs, which significantly enhances the realism and stability of cloth animations under varying physical conditions. This method is particularly effective in preventing the numerical instabilities commonly associated with large force interactions, thus avoiding the unrealistic stretching of fabric often seen in simpler simulation models. Their technique for adjusting spring constants in real-time ensures that cloth simulations remain stable and realistic, even when subjected to external forces like wind or interactions within virtual environments. This adaptability is crucial for maintaining the visual integrity of the cloth in dynamic scenes, which is essential for applications in video games and virtual reality, where real-time response and visual fidelity are paramount. This work aligns closely with the focus of my project on managing super elastic effects in cloth simulation. The methodologies discussed by Bayraktar et al. provided a foundation for the approaches I employed to tackle similar challenges in my framework. Specifically, the strategy of dynamically adjusting spring stiffness, as highlighted in their study, was instrumental in overcoming the issues of super elasticity that I encountered.

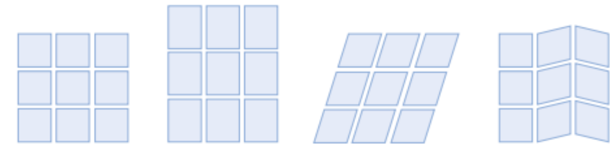



Figure 2: A simple visualization of stretching, shearing, and bending deformations of a square cloth patch [Stu18].

## 3. Overview

### 3.1. Overview of Blender Python API

Blender's Python API offers powerful control for cloth simulation, particularly through two core modules [Fou23]:`bpy.context` and `bmesh`.

#### 3.1.1. bpy.context

The `bpy.context` module provides access to Blender's current operational environment, enabling the manipulation and update of objects and properties within the scene. For instance, it allows for accessing currently selected objects, changing object positions, or dynamically adding and removing objects. This functionality is crucial in dynamic cloth simulation, as it necessitates the real-time updating of the cloth's state based on the results of physical calculations.

#### 3.1.2. bmesh

`bmesh` is a robust module for handling mesh data, offering advanced operations on basic elements such as vertices, edges, and faces. In the context of cloth simulation, `bmesh` is utilized to dynamically modify the positions of mesh vertices, thus achieving the dynamic behavior of the cloth. Each vertex is connected to others, simulating spring connections, and `bmesh` facilitates the adjustment of these vertices' positions based on the spring forces calculated at each timestep.

Using these tools, a system is established where each mesh vertex is treated as a particle. These particles are endowed with properties such as position and velocity and are assumed to be connected by springs. The positions and velocities of these particles are updated based on spring and damping forces at each timestep, simulating the natural behavior of cloth.

### 3.2. Mass-Spring Model

We know from wearing garments and handling textiles on a regular basis that under normal circumstances, materials usually show very little shearing and stretching. On the other hand, materials tend to bend out of the plane more easily, giving rise to the folds and wrinkles that give materials their drape and texture. Figure 2 illustrates these kinds of deformations, showing how cloth naturally prefers to flex rather than stretch or compress. To accurately simulate these resistance behaviors against various deformations, we implement a system of springs that connect each pair of adjacent particles within

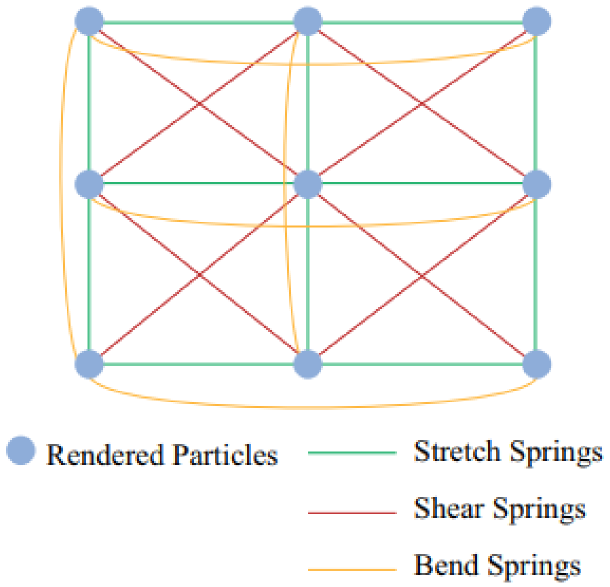



Figure 3: A simple mass-spring system consisting of nine particles [Stu18].

the cloth model. This network of springs is designed to replicate the fabric's inherent mechanical properties:

**Structural Springs:**

Structural springs connect each particle to its immediate neighbors, providing the primary resistance against the fabric stretching or compressing too much. They are crucial for maintaining the overall structure of the cloth and ensuring that it behaves in a realistic manner under tensile stress.

**Bend Springs:**

Linked to every alternate particle, bend springs are essential for simulating the cloth's ability to fold and bend without significant resistance. Although these springs facilitate the formation of folds and wrinkles, they will be omitted in this project since the focus is on simulating cloth that hangs freely, where such bending deformations are minimal.

**Shear Springs:**

Connecting diagonally across particles, shear springs activate under compression and are instrumental in maintaining the cloth's shear integrity. They help the fabric preserve its shape under lateral stresses and prevent it from shearing too easily.

For example, Figure 3, shows a simple mass-spring system consisting of nine particles organized in a grid. With this setup, we can watch and adjust how various kinds of spring affect the cloth's behavior dynamically. Because each of these spring types might have a distinct rest duration and stiffness constant, k, it is important to distinguish between them. By modifying the stiffness values of these springs, we can make substantial adjustments to the simulation and accurately replicate the mechanical behavior of different materials in the real world. Stretching springs typically have a stiffer constant than bending and shearing springs, which is consistent with most real-world materials where flexibility is preferred over elongation.

It is also important to note that although springs are called for the kind of deformation they affect mostly, other types of deformation resistance may be slightly impacted by their effects. For example, shear springs might marginally affect the amount of stretch that the fabric stretches even though its primary function is to prevent lateral deformation. The intricate interplay of several types of springs imbues the simulation with intricacy and authenticity, guaranteeing that the fabric responds authentically to a range of environmental circumstances.

#### 3.2.1. Spring Force

In the realm of cloth simulation, the potential energy stored within springs comes into play whenever there's a deviation from their rest length due to compression or stretching. This potential energy is derived from Hooke's Law, which suggests that the force applied by a spring is directly proportional to its displacement from the rest position. For a spring that bridges particles $i$ and $j$ with a rest length $L$, the exerted force can be articulated by a simple yet powerful formula:

$$F_{ij}^{sp} = k_{ij}^{s}(l_{ij} - \|\vec{x}_i - \vec{x}_j\|)\frac{\vec{x}_i - \vec{x}_j}{\|\vec{x}_i - \vec{x}_j\|}. \tag{1}$$

Here, $k_{ij}^{s}$ is the spring constant that captures the material's stiffness, and $l_{ij}$ denotes the rest length of the spring. The unit vector gives us the direction of the force applied. In this project, the formulas for the forces exerted by stretch and shear springs are essentially the same. The distinction lies in the stiffness constant.

#### 3.2.2. Damping Force

In cloth simulation, damping is a necessary component that accounts for the energy dissipation over time within the system. This dissipation typically arises from various forms of resistance encountered by the moving fabric, such as air drag or friction with other surfaces. When a spring within the cloth model connects two particles, $i$ and $j$, damping is represented as a force that acts in the opposite direction to the relative velocity of these particles. The damping force applied to particle $i$, connected to particle $j$, can be expressed as:

$$F_{ij}^{d} = -k_{ij}^{d}(v_i - v_j), \tag{2}$$

where $k^d$ is the damping coefficient, a parameter that determines the rate at which the system's energy is lost. Higher values of $k^d$ result in quicker energy loss, simulating a more 'damped' system, while lower values allow for more prolonged oscillations before the system settles.

It's important to recognize that while the simple linear damping model is computationally efficient and easy to implement, it may not always provide the most physically accurate results, especially when dealing with complex cloth materials or interactions. More complex damping models may consider factors like the velocity squared, which better approximates the non-linear behavior of certain damping forces, or even use different damping coefficients for different modes of vibration within the cloth.

#### 3.2.3. Simulation step: Symplectic Euler

The update of our simulation at each time step involves calculating the net force on each particle, considering the effects of both internal and external forces. For particle $i$, this net force is a sum of the spring forces from all connecting particles and the damping force due to the particle's velocity, along with gravity and any other external forces. The total force on particle $i$ can be expressed as:

$$\vec{F}_i = -k_d \vec{v}_i + m_i \vec{g} + \vec{F}^{\text{ext}} + \sum_j \left( \vec{F}^{\text{sp}}_{ij} + \vec{F}^{\text{d}}_{ij} \right), \tag{3}$$

where $\vec{F}^{\text{ext}}$ includes all external forces acting on the particle. The sum accumulates the forces from all particles $j$ connected to particle $i$ by springs.

In cloth simulation, each particle's state is updated at discrete intervals, a process controlled by the equations of motion. The symplectic Euler method is a numerical technique that allows us to approximate the solution of these equations over time, providing a means to predict the future state of the particles.

In our simulation, let's denote the position and velocity of a particle $i$ at time $t$ as $\vec{x}_i(t)$ and $\vec{v}_i(t)$, respectively. The symplectic Euler method updates the state of the particle using the force $\vec{F}$ acting on it in two stages:

**Velocity Update:** The new velocity $\vec{v}_i(t+\Delta t)$ is calculated based on the current velocity and the net force applied to the particle. It assumes that the force remains constant over the time step $\Delta t$. Mathematically, this can be expressed as:

$$\vec{v}_i(t+\Delta t) = \vec{v}_i(t) + \frac{\Delta t}{m_i} \vec{F}(\vec{x}_i(t), \vec{v}_i(t), t). \tag{4}$$

**Position Update:** Once the new velocity is obtained, the particle's position is updated using this new velocity. This step can be formulated as:

$$\vec{x}_i(t+\Delta t) = \vec{x}_i(t) + \Delta t \vec{v}_i(t+\Delta t). \tag{5}$$

Symplectic Euler is particularly effective in preventing the common issues of energy gain or loss seen in non-symplectic integrators like the standard Euler method, which can lead to unrealistic behavior over time. The choice of time step $\Delta t$ is crucial in the symplectic Euler method for cloth simulation. It dictates not only the stability and accuracy of the simulation but also its responsiveness to rapid changes in the physical state of the cloth. Too large a time step may lead to numerical instabilities, causing artifacts such as excessive stretching or even a total breakdown of the simulation.

#### 3.2.4. Pseudocode Implementation

The algorithm 1 below summarizes the algorithm for updating the cloth simulation in each time step of our actual simulation.

**Algorithm 1** Cloth Simulation Update using Mass-Spring Model

**Input:** List of particles with initial positions and velocities, list of springs with rest lengths and stiffness, timestep $\Delta t$, gravity $g$, external forces $F_{ext}$
**Output:** Updated positions and velocities of particles
**for** each timestep **do**
  **for** each particle $i$ not in pinning set **do**
    Initialize $F_i \leftarrow \vec{0}$
    Add gravity force $F_i \leftarrow F_i + m_i g$
    Add external forces $F_i \leftarrow F_i + F_{ext}$
    **for** each spring $s$ connected to particle $i$ **do**
      Calculate spring force $F_s$
      Calculate damper force $F_d$
      Add spring force to total force $F_i \leftarrow F_i + F_s + F_d$
    **end for**
    Update particle's velocity $v_i \leftarrow v_i + \Delta t \frac{F_i}{m_i}$
    Update particle's position $x_i \leftarrow x_i + \Delta t v_i$
  **end for**
  Apply constraints (e.g., pinning or collision response)
**end for**

#### 3.2.5. Super Elasticity Effect

While the mass-spring model offers a straightforward approach to cloth simulation, it presents a challenge commonly referred to as the super elasticity effect. In simple terms, a mesh simulated with a basic mass-spring model above tends to behave like rubber, exhibiting excessive elasticity, or flowing downward under its own weight if the particles are too heavy or densely packed.

In traditional mass-spring models for cloth simulation, the force-extension behavior is assumed to be linear. However, when a small element of the cloth is subjected to large concentrated forces, substantial spring deformations can lead to unnatural stretching and compressing of the cloth simulation. This phenomenon, known as the super elasticity effect, results in local deformations that appear less realistic than actual cloth, which lacks such properties, as discussed by Mozafary et al. [MP16]. Real cloth exhibits a nonlinear response to forces; stiffness increases dramatically as the deformation rate goes up. Under very high loads, the real fabric would tear before undergoing large deformations. Most researchers agree that the maximum deformation rate for most woven fabrics is around 10%, though this value may be lower for linens and calicos.

One straightforward method to avoid super elasticity is to increase the spring stiffness coefficient. Yet, simply increasing the spring stiffness might lead to model instability. To prevent excessive stretching, Bayraktar et al. [BGO07] introduced an elongation limit between the rest length and the extension to a spring's rest length. At each step, we measure the ratio between the distance between particles and the rest length, evaluating how far we are from the spring's predetermined elasticity. The spring coefficient is then scaled by the ratio of the distance to the allowed elongation to find the new coefficient. The spring stiffness is adjusted according to the following equation:

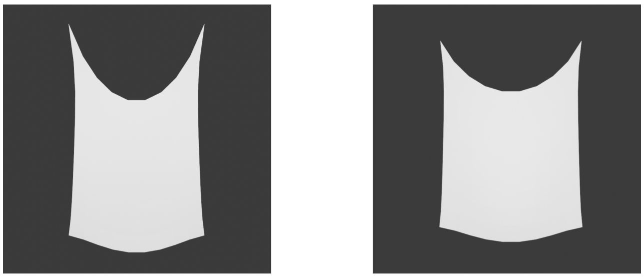
$$K_{\text{new}} = \begin{cases} \frac{K_{\text{original}} \cdot L}{L_0} & \text{if } L > L_{\text{allowed}} \\ K_{\text{original}} & \text{otherwise} \end{cases}, \tag{6}$$

where $K_{\text{new}}$ is the new stiffness coefficient, $K_{\text{original}}$ is the original stiffness coefficient, $L$ is the current spring length, $L_0$ is the spring's rest length, and $L_{\text{allowed}}$ is the maximum allowable spring length beyond which the stiffness coefficient is adjusted. In this project, the maximum allowable spring length is determined by multiplying the deformation rate by the rest length, providing a scalable boundary based on the cloth's expected mechanical behavior under strain.

By carefully managing these updates—stiffness adjustment, force calculation, and position updates, we ensure that the simulation progresses smoothly and that the cloth behaves in a manner consistent with its physical properties and the forces acting upon it.

## 4. Evaluation

This section presents an analytical review of the developed simulation system, highlighting the results achieved and identifying areas for potential improvement.

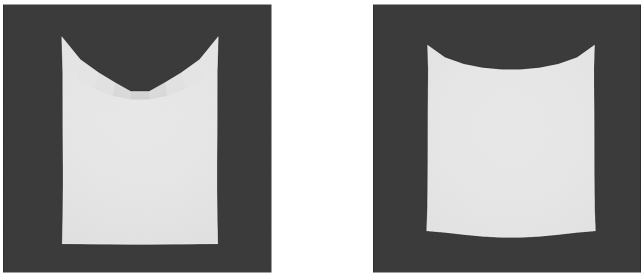


(a) (b)

Figure 4: Comparative visualization of cloth simulation without and with $k$ adjustment.

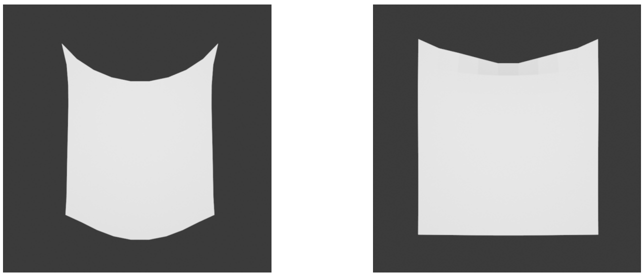
Figure 4 shows the cloth simulation without and with $k$ adjustment on heavy masses. In the image (a), without $k$ adjustment, the cloth appears to exhibit super elasticity. The simulation may lack the necessary resistance to deformation, resulting in a less realistic drape and an overstretched appearance. In the image (b), with $k$ adjustment, the simulation seems more realistic. The adjustment of the spring stiffness coefficient based on the forces applied provides a more accurate representation of how heavy cloth would naturally behave under gravity, avoiding the super elastic effect and maintaining the structural integrity of the cloth.

Figure 5 offers a visual evaluation of cloth simulations influenced by varying degrees of stretch stiffness. Image (a) exemplifies the effect of low stretch stiffness on the simulated fabric. In this instance, the cloth demonstrates a propensity for elongation, resulting in a more pronounced drape. Conversely, image (b) depicts the cloth under the influence of higher stretch stiffness. Here, the fabric maintains a more concise and structured drape, resisting the pull of gravity to a greater extent. The simulation restricts the fabric's extension, preventing it from stretching too far from its original form. This rigidity is reflective of materials such as canvas or denim, which are known for their durability and lesser propensity to stretch. The simulated cloth, therefore, retains more of its initial shape, suggesting a material composition that is inherently less fluid. This behavior is particularly relevant when simulating garments or scenarios where the fabric is expected to exhibit structural integrity and minimal deformation.



(a) (b)

Figure 5: Comparative visualization of cloth simulation with different stretch stiffness.



(a) (b)

Figure 6: Comparative visualization of cloth simulation with different shear stiffness.

Figure 6 demonstrates the impact of shear stiffness on cloth simulation, revealing how material properties translate into visual deformations. In the image (a), we see a simulation configured with low shear stiffness, which results in distinctive deformation patterns. The cloth appears to yield primarily at its corners, where the shearing forces are concentrated. This leads to an exaggerated stretching effect along the diagonals, causing the fabric to display an almost hyperbolic curvature that suggests a highly flexible material with minimal resistance to lateral forces. On the other hand, image (b) represents a cloth simulation with high shear stiffness, where the material exhibits far less pronounced diagonal stretching. This suggests a fabric with a strong resistance to shearing, typical of densely woven or stiffer materials. In such simulations, the cloth maintains its shape better under lateral forces, leading to a more uniform distribution of stress and less localized deformation. The result is a more stable and taut appearance, indicative of materials such as heavy cotton or wool blends commonly used in upholstery or outerwear.

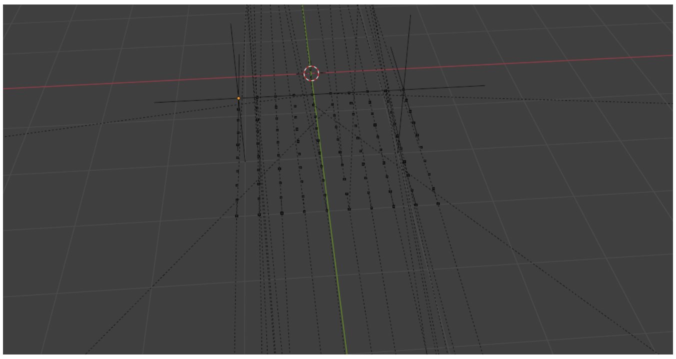


Figure 7: Breakdown Due to Large Timestep

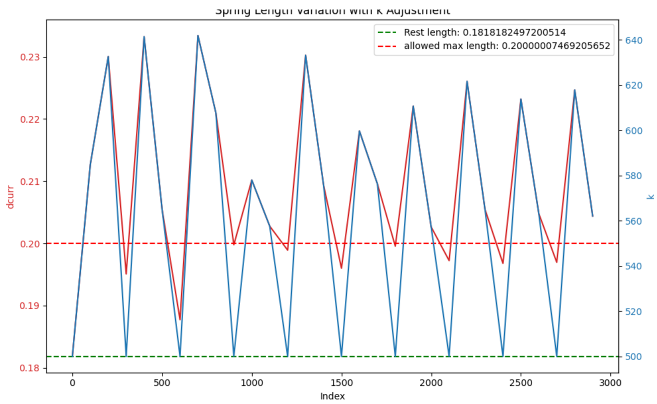



Figure 8: Spring length variation with k adjustment

The simulation's temporal resolution, determined by the timestep $\Delta t$, significantly influences its computational behavior. A larger timestep can accelerate the simulation, leading to quicker results. However, it compromises stability and accuracy, as the forces acting on the cloth are not updated frequently enough to capture the subtle changes in its dynamics. This often results in an accumulation of errors that escalate rapidly, leading to an explosion of energy within the system, which can be visually observed as an unnatural distortion of the cloth mesh or even a complete simulation breakdown as shown in Figure 7.

## 4.1. Limitations and Challenges

### 4.1.1. Balancing Simulation Parameters.

Our experiments have demonstrated that the simulation of cloth hanging naturally poses no significant issues with the current model. However, a delicate balance between input parameters — mass $m$, spring constant $k$, and damping $d$ — has been observed. Deviations beyond certain proportions between these parameters lead to simulation instability and potential collapse. This balance indicates a complex interplay where each parameter must be carefully tuned to ensure the simulation's integrity. Although a comprehensive study of the interactions between these parameters and their respective upper and lower bounds was beyond the scope of this project due to time constraints, the initial findings suggest a rich area for future investigation. Understanding the precise ratios and thresholds that govern stable simulations can reveal deeper insights into the simulation framework and contribute to the development of more robust models

### 4.1.2. Scope of Simulation and Lack of Collision Handling.

The scope of our simulation is currently limited to modeling an $n \times n$ mesh subjected solely to gravitational forces, which is an idealized scenario that does not encapsulate the full complexity of cloth behavior in real-world settings. Interactions such as collisions with other objects or environmental influences like wind have not been incorporated into our system, which restricts the utility of our simulation in more dynamic contexts. While the hanging cloth simulation provides a foundational understanding and a controlled environment to test the mass-spring system, real-world applications often involve complex interactions that require advanced collision detection and response mechanisms. The absence of collision handling means that the cloth cannot react to or interact with other entities within the environment.

### 4.1.3. Limitations of Stiffness Adjustment

While the simulation framework features a dynamic adjustment of spring stiffness, there remain challenges in maintaining spring lengths within their maximum allowable limits. Figure 8 shows the tracking of spring length and stiffness variations for a vertex at the edge of a hanging cloth that is closest to the pinning corner, with a mass of 0.1, stretch stiffness $k_{\text{stretch}} = 500$, and damping $d = 1$. The green dashed line represents the rest length, and the red dashed line indicates the allowed maximum spring length. As shown in Figure 8, While the method 6 of adjusting the stiffness based on their elongation offers a bit of control over the super elasticity effect, it observes that certain springs still exceed their maximum allowable length. This could be attributed to the simplicity of the stiffness adjustment method, which may not be sufficiently responsive to sudden or large forces. The adjustment is directly proportional to the stretch beyond the rest length, which might not scale effectively under rapid or extensive deformations.

The stiffness adjustment method is fundamentally passive; it adjusts the spring constant only after a stretch has occurred. In cases where the cloth is under sharp or heavy loads, the adjustment lag causes the spring length to instantly exceed a predefined threshold before the corrected stiffness is applied. As a result, this can lead to peaks in a stretch, as shown in the case where the spring length exceeds the allowable maximum. Furthermore, the correction is linear and this method may not capture the complex nonlinear material behavior under different stress conditions. More sophisticated methods, such as the position and velocity correction method proposed by Zhong et al. [ZX09], provide more precise control schemes. This method proposes that if the deformation rate of a spring exceeds a specific critical value, the two ends of the spring move towards each other in the axis until the deformation rate drops to the critical value. Although calculations are quite extensive, this kind of active approach can preemptively solve the overstretching problem and ensure that the spring length stays within the specified range. However, due to the time constraints of this project, we did not explore these additional methods.

## 5. Conclusion

As we come to a conclusion of our investigation into cloth simulation with a mass-spring model built in Blender, we have come across the system's advantages as well as its drawbacks. Through the successful implementation, it has been shown that it is feasible to simulate fabric hanging under the force of gravity, displaying realistic behavior when mass, spring stiffness, and damping are adjusted appropriately. But this voyage has also revealed nuanced thresholds and balances in the parameter space, illuminating the narrow boundary between computational instability and precise simulation.

### 5.1. Future Work

To expand the boundaries of this research and enhance this mass-spring model's capabilities for cloth simulation, several future works are proposed:

1. **Bend Spring Integration:** To more accurately represent the behavior of cloth, particularly its ability to fold and wrinkle, incorporating bend springs is essential. This would allow the simulation of detailed cloth characteristics, capturing the nuances of fabric drape and texture.
2. **Parameter Space Exploration:** A systematic study to determine the optimal ranges of the simulation parameters would provide valuable insights. Understanding the upper and lower limits for mass, spring stiffness, and damping would ensure stability across various simulation scenarios and lead to a set of best practices for parameter selection.
3. **External Force Dynamics:** The current model primarily considers gravity. Introducing additional external forces, such as wind or water, would add layers of complexity and realism, simulating the cloth's behavior in diverse and challenging environments.
4. **Collision and Self-Collision:** Integrating collision detection and response systems would improve the realism of the simulations, allowing the cloth to interact with its environment and itself. This would enable the simulation of more dynamic scenes where cloth can drape over objects or form complex folds without penetrating other surfaces or its own mesh.
5. **User Interface:** A user interface (UI) can bridge the gap between complex simulation parameters and user experience, providing a more intuitive means for users to interact with the simulation system.

## References

[BGO07] BAYRAKTAR S., GUDUKBAY U., OZGUC B.: Practical and realistic animation of cloth. In *2007 3DTV Conference* (2007), IEEE, pp. 1–4.

[BW23] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023, pp. 767–778.

[dFA20] DE FIGUEIREDO ASSUNÇÃO R.: Cloth simulation framework.

[Fou23] FOUNDATION B.: Blender 4.1 python api documentation, 2023. Accessed: 2024-04-1. URL: https://docs.blender.org/api/current/index.html.

[MP16] MOZAFARY V., PAYVANDY P.: Study and comparison techniques in fabric simulation using mass spring model. *International Journal of Clothing Science and Technology 28*, 5 (2016), 634–689.

[OU] OROSZ A., UTLEY S.: Tearing cloth in blender.

[Stu18] STUYCK T.: Mass-spring models. In *Cloth Simulation for Computer Graphics*. Springer, 2018, pp. 21–30.

[ZX09] ZHONG Y., XU B.: Three-dimensional garment dressing simulation. *Textile Research Journal 79*, 9 (2009), 792–803.