CHAPTER 4

Mass-Spring Models

"May the force be with you"

Obi-Wan Kenobi; ... and a lot of Star Wars fans.

In this chapter, we'll look at how the internal cloth forces are computed.

4.1 INTRODUCTION

You've probably heard somewhere before that all material is constructed out of atoms. As such, it's no surprise that also cloth is made out of atoms and neighboring atoms exert forces on each other preventing excessive stretching or compression. In computer graphics, we can take this idea and represent the continuous cloth by a discrete set of points. Of course, not quite as many as the number of atoms... Unless you're a very patient person! Continuing on this idea of having point masses exerting forces on each other to retain certain properties naturally leads to the mass-spring model.

4.2 COMPUTING MASSES

The name of the model is probably a little bit of a give-away but mass-spring models are none other than point masses connected by springs. Let's say you have some geometry that you would like to simulate as cloth—you can simply take the N vertices of the triangles as the point masses in our simulation model. Besides being a point, point masses have mass.

A good method to determine the mass is to have a surface density ρ with units $\left[\frac{\text{kg}}{\text{m}^2}\right]$ defined for a material. We model the cloth using 2D triangle elements without thickness. Heavier material will have a higher density and vice versa. We can loop over all the triangles and compute the mass as the triangle surface times the density to obtain the mass of that triangle. This mass is then equally distributed by adding one third of the triangle mass to all three vertices of the triangles. This is an approximation but works well in practice. A single particle will have mass contributions from all triangles it is part of. This area is assumed to be the area in the reference and thus undeformed configuration. This is the configuration the geometry would be in when it is undeformed by forces acting on it.

A commonly used trick is to add some additional mass to the particles that are on the hem of the garments. This extra mass is not necessarily proportional to the triangle area. The reason this results in increased realism is that the hem is often folded double and stitched. We can model this heavier double layer of cloth by modifying the masses. This is much easier than actually representing the geometric fold-over of the hem.

Later in this book, we'll see that in order to express our equations of motion conveniently, we will conceptually have a $\mathbb{R}^{3N \times 3N}$ dimensional mass matrix $\mathbf{M}$. This matrix has the particle masses on the diagonal and is zero otherwise.

Of course, since this matrix has such a simple structure, we only have to store an array of length N with one scalar mass value per particle. The full mass matrix $\mathbf{M}$ will appear in some of the equations that follow. Just to be completely clear, you don't want to construct this as a full dense matrix in your code but it is defined as follows:

$$\mathbf{M} = \begin{bmatrix} m_0 & 0 & 0 & 0 & \dots & 0 \\ 0 & m_0 & 0 & 0 & \dots & 0 \\ 0 & 0 & m_0 & 0 & \dots & 0 \\ 0 & 0 & 0 & m_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & m_{N-1} \end{bmatrix}. \tag{4.1}$$

4.3 COMPUTING FORCES

Assuming you're human: from experience with wearing clothes and interacting with textiles in everyday life, we know that cloth is not supposed to stretch or shear all that much. On the other hand, cloth tends to bend out of plane easily creating wrinkles and folds. These types of deformations are visualized in Figure 4.1.

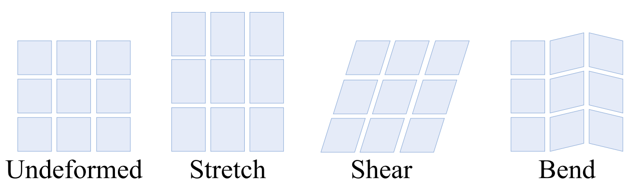

Figure 4.1: A simple visualization of stretching, shearing, and bending deformations of a square cloth patch.

In order to model resistance to deformations, we can simply construct a spring connecting every pair of neighboring particles. A simple mass-spring system containing nine particles is shown in Figure 4.2. The springs that we just constructed can be thought of as two different types that serve a different purpose. The springs that are shown in green resist stretching of the lattice and the purple springs counteract shearing forces.

A lot of the interesting visual information such as wrinkles and folds results from the cloth bending. A way to incorporate this in our model is to connect 2-ring neighbor particles with a spring, skipping the particle in between. These are called bend springs and are shown in yellow.

It might be interesting to keep these types separated because the spring constant depends on the type of spring. This will make it easier to set material parameters on the model so we have dials to control stretch and shear resistance separately. Most materials have a lower resistance to shearing. Varying the shear stiffness affects the visual behavior dramatically. As a guideline, stretch springs will have very stiff constants whereas shear and bend springs will have small values. Obviously, there is not a complete separation between stretching, shearing, and bending. For example, shear springs will also have some effect on stretching.

Keep in mind that it is totally up to you which particles you connect with springs. Just know that this will eventually have a profound effect on the way the cloth behaves. This will be more clear later on.

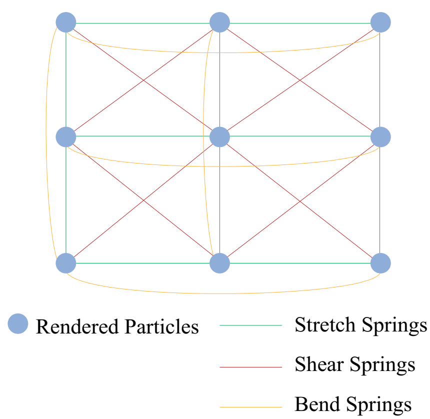

Figure 4.2: A simple mass-spring system consisting of nine particles connected by stretch springs and shear springs. Bend springs connect with every other particle.

4.3.1 ENERGY MINIMIZATION

Physical systems are always trying to reach a minimal energy state. Just think of a marble rolling down a hill, decreasing its potential energy. We can model this is by defining energies and by minimizing them. The forces are those that will try to bring the system in a state that has lower energy until we reach an equilibrium state. This is also known as the second law of thermodynamics.

Now, if you remember from your calculus class, the gradient of a function points in the direction of steepest ascent. But, we don't want to reach a higher energy state, so intuitively, it makes sense for the forces to be the negative gradient of the energy potential function $E(\mathbf{x}) \in \mathbb{R}$

$$\mathbf{f}(\mathbf{x}) = -\frac{\partial E(\mathbf{x})}{\partial \mathbf{x}}. \tag{4.2}$$

Note that this is only the case for *conservative forces*. A force is conservative when the total work done in moving the particle between two points is independent of the path taken. Another way of saying this is that when the particle moves in a loop but starts and ends in the same position, then the net work done will be zero.

These conservative internal forces only depend on the current particle positions. Other forces such as friction and collisions don't. As such, external forces such as collisions or friction forces are not defined by potential energies and are discussed in Chapter 10. To make this more clear, let's start with the most simple example of a conservative force: gravity.

Let's say that in our coordinate system, gravity acts along the z-axis. The gravitational acceleration g is roughly equal to 9.81 $\left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]$ depending on where in the world you are. The potential energy for a point mass i due to gravity is $E_g(\mathbf{x}_i) = m_i g x_{iz}$ with x_{iz} the component of the particle position along the z-axis. Following Equation (4.2), the resulting force will be

$$\begin{aligned}\mathbf{f}_i(\mathbf{x}) &= -\frac{\partial E_g(\mathbf{x})}{\partial \mathbf{x}_i} \\ &= -\left[\frac{\partial E_g}{\partial x_{ix}}, \frac{\partial E_g}{\partial x_{iy}}, \frac{\partial E_g}{\partial x_{iz}}\right] \\ &= -\begin{bmatrix} 0 \\ 0 \\ m_i g \end{bmatrix}.\end{aligned} \tag{4.3}$$

That looks a lot like Newton's second law of motion, $\mathbf{f} = \mathbf{M}\mathbf{a}$, doesn't it?

4.3.2 SPRING POTENTIAL ENERGY AND FORCE

Continuing with our cloth simulation, energy is stored in the springs whenever the spring is not at its rest length, i.e., when it is compressed or stretched. *Hooke's law* gives us an expression for determining the potential energy stored in a spring with rest length L and spring constant k with units $\left[\mathrm{kg/s^2}\right]$. This value k is also known as the *spring stiffness constant*. It plays an important role as it expresses how much the spring will resist deformation and it provides us with a dial to model different materials.

For a spring connecting particle i and j, we have the potential energy

$$E_{ij}(\mathbf{x}) = \frac{1}{2}k\left(||\mathbf{x}_i - \mathbf{x}_j|| - L\right)^2. \tag{4.4}$$

$||\cdot||$ is the Euclidian distance; see Equation (A.4). Now that we have our energy function for the spring, we can compute the forces that the spring exerts on particle i and j by taking the derivatives. The force on particle i is found as

$$\begin{aligned} \mathbf{f}_i(\mathbf{x}) &= -\frac{\partial E_{ij}(\mathbf{x})}{\partial \mathbf{x}_i} \\ &= -k\left(||\mathbf{x}_i - \mathbf{x}_j|| - L\right)\frac{(\mathbf{x}_i - \mathbf{x}_j)}{||\mathbf{x}_i - \mathbf{x}_j||} \end{aligned} \tag{4.5}$$

and, similarly, the force on particle j is computed as

$$\begin{aligned} \mathbf{f}_j(\mathbf{x}) &= -\frac{\partial E_{ij}(\mathbf{x})}{\partial \mathbf{x}_j} \\ &= k\left(||\mathbf{x}_i - \mathbf{x}_j|| - L\right)\frac{(\mathbf{x}_i - \mathbf{x}_j)}{||\mathbf{x}_i - \mathbf{x}_j||}. \end{aligned} \tag{4.6}$$

Looking at Figure 4.3, it should come as no surprise that $\mathbf{f}_i = -\mathbf{f}_j$. A spring connecting two particles will either pull or push on the particles on opposite directions with the same amount of force along the same axes. When the springs are in their rest position, they they will not exert any forces on the point masses. Note that this is a conservative force, just like gravity. The force the spring exerts is independent of the path taken and only depends on the endpoints.

The energy function is quadratic in the particle positions. The force is computed as the negative gradient and will therefore be linear in the positions. The resulting force will scale linearly with the amount of stretching or compression. This is called a linear spring or also, a *Hookean* spring. A graph showing the spring response for different spring stiffnesses k is shown in Figure 4.4.

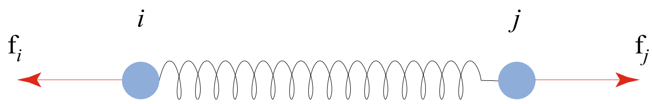

Figure 4.3: A single spring connecting particle i and j applies equal and opposite forces to the particles along the direction connecting the particles.

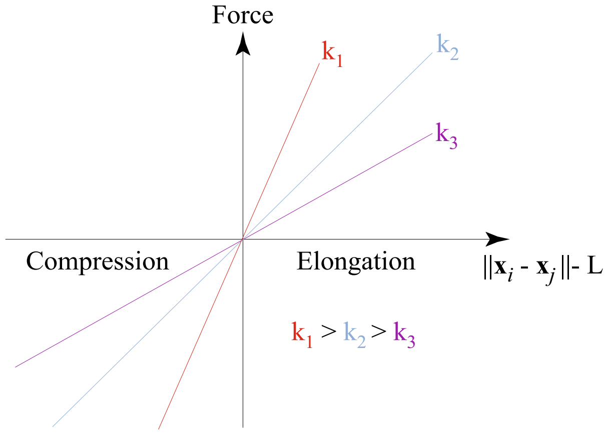

Figure 4.4: The spring force will be linear in the amount of stretching or compression. A larger spring constant will result in a bigger force response for a certain elongation or compression. The horizontal axis shows the deviation from the rest length.

4.3.3 SPRING DAMPING FORCE

Obtaining stable simulation results is critically dependent on having damping forces in the system. We hinted at this when discussing the stability of the oscillatory equation and integration using explicit integration in Section 3.3.2 of the previous chapter. The most simple way to model a damping force for a particle is to add a force that opposes the motion. For a particle i connected to particle j we have the damping force acting on particle i as

$$\begin{aligned} \mathbf{d}_i(\mathbf{x}) &= -k_d\left(\mathbf{v}_i - \mathbf{v}_j\right) \\ &= -\mathbf{d}_j(\mathbf{x}) \end{aligned} \tag{4.7}$$

with k_d the damping coefficient. This mimics the real-world behavior of energy dissipation. Note that this damping model is easy but far from perfect. It prevents bending of the cloth and it penalizes rigid rotations of the spring. Adding a small amount of damping will result in stable simulations. Adding too much damping will make the cloth seem to behave as if it were underwater.

4.4 PUTTING IT ALL TOGETHER

Let's say we start at time t_n. At each step we want to advance our time by a step h. In order to do so we need to compute our update in positions $\Delta\mathbf{x}$ and velocities $\Delta\mathbf{v}$. Following the discretized Newton's law of motion given in Equation (3.5), we come up with the following system:

$$\begin{aligned} \Delta\mathbf{x} &= h\mathbf{v}_n \\ \Delta\mathbf{v} &= h\left(\mathbf{M}^{-1}\mathbf{f}(\mathbf{x}_n, \mathbf{v}_n)\right) \end{aligned} \tag{4.8}$$

with $\Delta\mathbf{x} = \mathbf{x}_{n+1} - \mathbf{x}_n$ and $\Delta\mathbf{v} = \mathbf{v}_{n+1} - \mathbf{v}_n$. For every particle in the system, we can compute all the internal and external forces that are acting upon it and accumulate this in a single force vector. The internal forces are computed as the negative energy gradient and the external forces are added to these internal forces. This will allow us to find the velocity update $\Delta\mathbf{v}$. Given $\Delta\mathbf{x}$ and $\Delta\mathbf{v}$, the next state can trivially be found as

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{x}_n + \Delta\mathbf{x} \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + \Delta\mathbf{v}. \end{aligned} \tag{4.9}$$

Phew! Now that we finished all of that, we finally have a working cloth simulator, congratulations! Now is a good time to pat yourself on the back. You might notice however that the results aren't always as great as you hoped. Particularly, the solution might *explode* (not in an awesome special-effects-kind-of-way). The true solution will deviate dramatically from the computed solution unless you take very small time steps. The approximation visualized in Figure 3.1 can be pretty crude. Small discretization errors accumulate and the approximation quickly becomes worse and worse.

4.5 TEARABLE CLOTH

Beyond the Basics

At the beginning of our exposition, we talked about how springs model the internal elastic forces of the cloth. The more the springs are extended, the stronger the resulting force will be. Can we keep stretching the cloth indefinitely? Probably not, right?

Typically, cloth will stretch a small amount without too much resistance. However, stretching beyond this point will result in very strong forces that will resist this deformation. For example, cloth usually doesn't stretch much under its own weight. This can be modeled using advanced techniques which are briefly mentioned in the discussion of this chapter. A different way to handle this is to implement tearable cloth.

Some materials rip when stretched too far. As it turns out, this is actually very easy to model in our simulations and results in interesting dynamic motion. When the springs are stretched a certain fraction too far from their rest length, we can assume the cloth breaks and tears. In our model, we can simply remove this spring from the mass-spring network. This will disconnect the particles in question, creating a tear. This is the most simple approach to the tearing phenomenon. A more advanced method can be found in the work of Metaaphanon et al. [2009].

4.6 OTHER MASS-SPRING APPLICATIONS

Beyond the Basics

In this chapter, we explained how mass-spring systems can be used to model the dynamics of cloth. We wanted to quickly inform you of the fact that mass-spring systems have multiple additional applications in physics-based animation. The focus of this book is cloth simulation so we won't go into too much detail here but we will point you to further reading.

4.6.1 HAIR SIMULATION

Mass-spring systems have been successfully used for the simulation of hair dynamics. The model presented by Selle et al. [2008] incorporates collisions, friction, and torsion and is capable of producing clumping and sticking behavior. Mass-spring systems have also been used by Iben et al. [2013] to generate highly art directed curly hair. The method has proven to be incredibly successful in production.

A simple mass-spring system for hair is shown in Figure 4.5. Note that in addition to the geometric particles there are ghost particles that are necessary to model the hair dynamics. These ghost particle won't be used to render the hair geometry, hence the name.

4.6.2 SOFT BODY DYNAMICS

An extension to three dimensional deformable objects can easily be made. Just like we modeled cloth using triangles where the particles are connected by springs along the edge, we can model deformable volumes using tetrahedra. The geometry is discretized using tetrahedra, representing the full volume. This is also known as a tetrahedralization. A single tetrahedron is visualized in

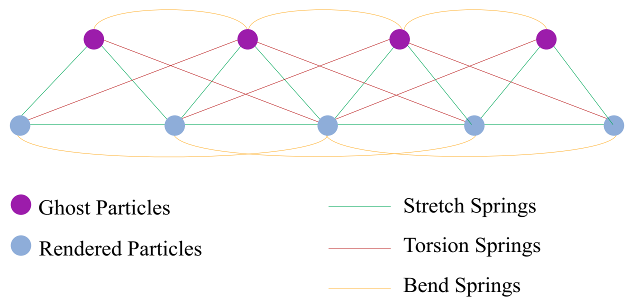

Figure 4.5: Visualization of a possible mass-spring discretization for a single hair strand. Just like cloth simulations, the dynamics are modeled using a variety of spring connections between particles.

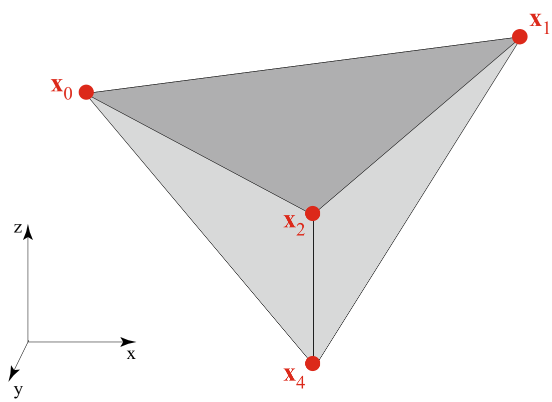

Figure 4.6: Visualization of a single tetrahedron element. The 3D element consist of four connected particles.

Figure 4.6. The element consists of four vertices and the particles are connected using springs along the edges. This will make the tetrahedron want to preserve volume when subjected to external forces. For a more elaborate discussion, we refer to the work of Teschner et al. [2004].

4.7 CONCLUSION

We have introduced the most simple implementation for a cloth solver. All the particle states for the next time step can be computed based on information of the current time step. We presented a method that models the cloth dynamics by using different types of linear springs to incorporate stretching, shearing and bending in the cloth material. The linear spring will have a force response linearly related to the amount of stretch or compression. This makes simulation simple but doesn't realistically reproduce physical cloth behavior. Typically, cloth will be able to stretch a small amount after which it will resist stretching much stronger. This can be modeled using nonlinear springs or piecewise linear springs. More advanced techniques such as strain limiting can also be used. The resistance to shearing and bending is much smaller than the resistance to stretching.

The forces exerted by these springs on the point masses can be computed by thinking of the problem as an energy minimization of the entire particle system. We saw that the forces are thus computed as the negative gradients of the Hookean energies.

Mass-spring models are not the only way to model cloth. More sophisticated techniques using finite element methods are not uncommon in the literature. We provide a more advanced continuum method in Chapter 7.