# Computer Animation 2022
# Exercise 4 - Cloth

Digital ART Laboratory

**Deadline:2022.04.28, 23:59**

## 1 Goals and Rules

In this exercise, you are going to simulate cloth using the spring system.

**Setup** Please update (`git pull`) the framework code at the git repository and follow the instruction to compile and run the code. If you meet performance problems while solving tasks, please make sure your build is release (`cmake -DCMAKE_BUILD_TYPE=RELEASE ../`).

(repo link: `http://dalab.se.sjtu.edu.cn/gitlab/courses/ca-framework-2022`)

**Submission.** Compress your solution and name it in form **NAME_ID_ex4.zip**, then submit it in Canvas. **You need to provide a document clarifying your numerical algorithm and comparison of methods.** Note: You only need to submit the part of the code that you modified, i.e., `3_cloth` directories.

## 2 Spring System for Cloth

The problem of cloth simulation has existed for more than two decades in the graphics community. Different methods have been proposed to solve it under various scenarios. In this exercise, we are going to using a discrete spring system to simulate the dynamics of cloth.

The cloth is represented by a collection of discrete particles, shown in Figure 1. There are three types of spring connecting these particles:

- Structural constraint: a particle with index $(i, j)$ has structural constraint with adjacent particles with index $(i+1, j), (i-1, j), (i, j+1), (i, j-1)$.
- Shear constraint: a particle with index $(i, j)$ has shear constraint with particles with index $(i+1, j+1), (i-1, j-1), (i-1, j+1), (i+1, j-1)$.
- Bending constraint: a particle with index $(i, j)$ has bending constraint with particles with index $(i+2, j), (i, j+2), (i-2, j), (i, j-2)$

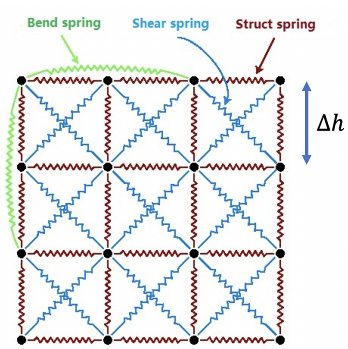


**Figure 1** *Spring constraints*

Let $k_{bend}, k_{shear}, k_{struct}$ be stiffness coefficients of these three types of spring, and define rest length of them as initial distance between particles, i.e., $L_{bend} = 2\Delta h, L_{shear} = \sqrt{2}h, L_{struct} = \Delta h$, where $\Delta h$ is the grid space of the discretized cloth. So the force produced by the structural spring between particles $a$ and $b$ are:

$$\hat{\mathbf{p}}_{ab} = \frac{\mathbf{p}_a - \mathbf{p}_b}{\|\mathbf{p}_a - \mathbf{p}_b\|}$$
$$\mathbf{f}_{struct-a} = k_{struct}(L_{struct} - \|\mathbf{p}_a - \mathbf{p}_b\|)\hat{\mathbf{p}}_{ab}$$
$$\mathbf{f}_{struct-b} = -\mathbf{f}_{struct-a}$$

Similar to what we did in `exercise 1`, we add damping force to the system so that the cloth eventually will be in a stable state. The damping force between particles $a, b$ is modelled as

$$\mathbf{f}_{da} = k_d((\mathbf{v}_b - \mathbf{v}_a) \cdot \hat{\mathbf{p}}_{ab})\hat{\mathbf{p}}_{ab}$$
$$\mathbf{f}_{db} = -\mathbf{f}_{da}$$

where $\mathbf{v}_a, \mathbf{v}_b$ and $\mathbf{p}_a, \mathbf{p}_b$ are velocities and positions, $\mathbf{f}_{da}, \mathbf{f}_{db}$ are damping force on particles $a$ and $b$ respectively, and $k_d$ is the damping coefficient. The above damping equation is more general compared to the one we used in `exercise 1`. With the damping force defined, we have the final force for particle $a$ as

$$\mathbf{f}_a = m\mathbf{g} + \sum_i \mathbf{f}_{struct-ai} + \mathbf{f}_{shear-ai} + \mathbf{f}_{bending-ai} + \mathbf{f}_{dai} \tag{1}$$

where summation over $i$ means looping over neighbor particles of particle $a$.

Up to this point, we can simulate the cloth using the explicit Euler method (symplectic Euler, mid-point, or RK4). However, in this problem, we are using springs with very large stiffness (5000 or even higher). So the explicit method suffers from the stability problem and needs a quite small time step to get an 'ok' result. Following `Implicit Methods` Section in [1] [2], the discrete spring system can simulate implicitly with equation:

$$(M - \Delta t\frac{\partial \mathbf{f}}{\partial \mathbf{v}} - \Delta t^2\frac{\partial \mathbf{f}}{\partial \mathbf{p}})\Delta \mathbf{v} = \Delta t(\mathbf{f}_n + \Delta t\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\mathbf{v}_n) \tag{2}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta \mathbf{v} \tag{3}$$

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \Delta t\mathbf{v}_{n+1} \tag{4}$$

where $\Delta t$ is time step, $M$ is a $3n \times 3n$ ($n$ is the number of particles) diagonal matrix containing mass of particles, $\mathbf{f}$ is the concatenated force vector of shape $3n$, $\mathbf{v}$ and $\mathbf{p}$ are the concatenated velocity vector and the concatenated position vector. Equation 2 is a linear system of form $A\mathbf{p} = \mathbf{b}$ with sparse, SPD (symmetric and positive definite) $A$, which is quite common in physics simulation. This type of linear system can be solved using a number of efficient iterative solvers. In the code framework, we have provided a basic structure of the `Conjugate Gradient` solver. You need to form the matrix $A$ and right-hand-size vector $\mathbf{b}$ to get the final result. For derivation of $\frac{\partial \mathbf{f}}{\partial \mathbf{v}}$ and $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$, see Appendix for some hints.

For particles with fixed positions, mathematically, we should remove them from the linear system, which means deleting rows and cols related to fixed particles in the $A$ matrix. In the implementation, we could also handle these fixed particles by zeroing out all entries in these rows and cols (also the corresponding entries in right-hand-size vector $\mathbf{b}$), and filling in an identity submatrix.

## 2.1 Questions (20 Points)

- Simulate cloth using the explicit Euler method. (6 Points)
- Simulate cloth using the implicit Euler method. (8 Points)
- Simulate using a combination of a half step of explicit Euler and a half step of implicit Euler. (2 Points)
- Doc. Compare the performance, stability of these three methods. (4 Points)
- (Extra points) Simulate cloth fall down and collide with a table ($z = 0$ plane).

# References

[1] David Baraff and Andrew Witkin. Physically based modeling. SIGGRAPH 2001 Course Notes. Available online at `https://graphics.pixar.com/pbm2001/`.

[2] David Baraff and Andrew Witkin. Physically based modeling: Principles and practice. SIGGRAPH 1997 Course Notes. Available online at `http://www.cs.cmu.edu/~baraff/sigcourse/`.

# Appendix A Derivations of $\frac{\partial \mathbf{f}}{\partial \mathbf{v}}$ and $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$

Both $\frac{\partial \mathbf{f}}{\partial \mathbf{v}}$ and $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$ have shape $3n \times 3n$. These two matrices are sparse and only have entries between neighboring particles. For example, if there is a spring force between particle $a$ and $b$, we will have a small non-zero $3 \times 3$ matrix at $[I_a, I_a+1, I_a+2] \times [I_b, I_b+1, I_b+2]$ of the original large matrix, where $I_a$ and $I_b$ are index of particles $a$ and $b$. So as long as we have derivations between two particles and fill them in the big matrix, we will get the final large matrix used in Equation 2. For a single spring force, we have the following equations:

$$\begin{aligned}
\hat{\mathbf{p}}_{ab} &= \frac{\mathbf{p}_a - \mathbf{p}_b}{\|\mathbf{p}_a - \mathbf{p}_b\|} \\
\mathbf{f}_a &= k(L - \|\mathbf{p}_a - \mathbf{p}_b\|)\hat{\mathbf{p}}_{ab} \\
\mathbf{f}_a &= -\mathbf{f}_b \\
\frac{\partial \mathbf{f}_a}{\partial \mathbf{p}_a} &= k(-\frac{\partial \|\mathbf{p}_a - \mathbf{p}_b\|}{\partial \mathbf{p}_a}\hat{\mathbf{p}}_{ab}^T + (L - \|\mathbf{p}_a - \mathbf{p}_b\|)\frac{\partial \hat{\mathbf{p}}_{ab}}{\partial \mathbf{p}_a}) \\
&= -\frac{\partial \mathbf{f}_b}{\partial \mathbf{p}_a} = -\frac{\partial \mathbf{f}_a}{\partial \mathbf{p}_b} = \frac{\partial \mathbf{f}_b}{\partial \mathbf{p}_b}
\end{aligned}$$

Differential relates to vector $\mathbf{x}$ norm has following relation:

$$\begin{aligned}
\hat{\mathbf{x}} &= \frac{\mathbf{x}}{\|\mathbf{x}\|} \\
\frac{\partial \|\mathbf{x} - \mathbf{y}\|}{\partial \mathbf{x}} &= \frac{\mathbf{x} - \mathbf{y}}{\|\mathbf{x} - \mathbf{y}\|} = -\frac{\partial \|\mathbf{x} - \mathbf{y}\|}{\partial \mathbf{y}} \\
\frac{\partial \hat{\mathbf{x}}}{\partial \mathbf{x}} &= \frac{I - \hat{\mathbf{x}}\hat{\mathbf{x}}^T}{\|\mathbf{x}\|}
\end{aligned}$$

So substitute them into the spring force equation we have

$$\frac{\partial \mathbf{f}_a}{\partial \mathbf{p}_a} = k(-\hat{\mathbf{p}}_{ab}\hat{\mathbf{p}}_{ab}^T + (L - \|\mathbf{p}_a - \mathbf{p}_b\|)\frac{I - \hat{\mathbf{p}}_{ab}\hat{\mathbf{p}}_{ab}^T}{\|\mathbf{p}_a - \mathbf{p}_b\|})$$

The above equations are for $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$. And $\frac{\partial \mathbf{f}}{\partial \mathbf{v}}$ can be obtained in a similar way.

# Appendix B Debug Tips

- When there are no particles fixed, the cloth should fall with gravity, and no deformation happens.
- Under the default setup, the implicit method should be stable with structural spring force (no damping, no other spring force) and gravity.
- Make sure the $A$ matrix is symmetric.