Mass Spring Method Cloth Simulation: Verlet and Forward Euler and R-K

Zhelin Zhu
Shanghaitech University
zhuzhl@shanghaitech.edu.cn

Tianyue Zhou
Shanghaitech University
zhouty1@shanghaitech.edu.cn

Xihe Yu
Shanghaitech University
yuxh@shanghaitech.edu.cn

Abstract

This project did a cloth simulation and explored three numerical methods applied in this simulation. We divided cloth into masses, used springs to connect masses, and then applied Forward Euler, Verlet Integration, and Runge Kutta methods in the numerical calculation. We evaluated the three numerical methods and compared their performance.

1 Introduction

The field of computer graphics has long been fascinated by the challenge of simulating realistic cloth behavior. From virtual garments in video games and animations to virtual reality experiences, cloth simulation plays a crucial role in enhancing visual realism and immersion. In this project, we delve into the realm of cloth simulation and explore the application of three distinct numerical integration methods: Forward Euler, Verlet Integration, and Runge Kutta methods.

Cloth simulation involves the creation of virtual fabrics that exhibit accurate movement, stretching, and bending, mimicking the behavior of real-world textiles. Achieving such realism requires complex algorithms and numerical techniques that accurately model the physical properties of cloth, including mass, structural constraints, and external forces.

The Forward Euler method is a simple but widely used numerical integration technique. By approximating the derivatives of position and velocity, it allows us to update the cloth's state from one time step to the next. We investigate the application of Forward Euler to cloth simulation and analyze its advantages, limitations, and trade-offs in terms of accuracy and stability.

The second numerical integration scheme we investigate is the Verlet Integration method. Based on the concept of position Verlet and velocity Verlet algorithms, it may offer improved stability and energy conservation properties compared to Forward Euler. We explore the implementation of Verlet Integration for cloth simulation, assessing its ability to produce visually realistic and physically accurate cloth behavior.

The third method is the R-K method, which can be used to calculate and analyze the deformation and dynamic behavior of cloth under various conditions. For example, the R-K method can be used to simulate the deformation of the cloth when it is subjected to external forces or gravity, as well as the strain and stiffness changes of the cloth under different temperature and humidity conditions.

Throughout this project, we aim to compare and contrast the three numerical integration methods—Forward Euler and Verlet Integration and R-K in the context of cloth simulation. By analyzing their strengths and weaknesses, we aim to expand our understanding of numerical methods and their

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

impact on visual realism and computational efficiency. The more specific code and algorithm analysis is in the *readme.md*.

2 Our Methods

2.1 Problem Definition

First, we modeled cloth as mass spring. We divided cloth into masses and used springs to connect them. In this way, the cloth can be discretized and we can compute its movement.

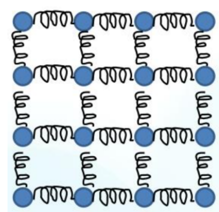

Figure 1: Mass Spring and Discretization

To be more specific, we consider the cloth connected by springs as having the following three types of springs.

1. structural-springs,which connect the (i,j)-(i,j+1) ,(i,j)-(i+1,j)
2. shear-springs,which connect the (i,j)-(i+1,j+1) and (i+1,j)-(i,j+1)
3. bending-springs,which connect the (i,j)-(i+2,j),(i,j)-(i,j+2)

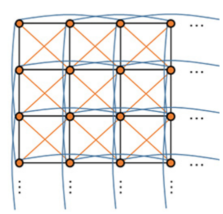

Figure 2: Mass Spring Model

For each mass, given the velocity and displacement in time t, we need to solve its movement equation to get its velocity and displacement in time $t+1$. And we let the left-top and right-top corner to be fixed.

$$\begin{aligned} \dot{v}(t) &= a(t) = \frac{f(t)}{m} \\ \dot{x}(t) &= v(t) \end{aligned} \tag{1}$$

We can use Forward Euler, Verlet Integration, or Runge-Kutta Methods to approximately solve the movement equation.

2.2 Forward Euler

Forward Euler Method is a numerical method for solving differential equations. It is one of the simplest and most commonly used methods. The method assumes that the function maintains a constant slope within each small time interval and estimates the unknown function values through iterative calculations.

Specifically, the Forward Euler Method discretizes the derivative in the differential equation using finite differences and then uses the iteration formula $y_{n+1} = y_n + hf(x_n, y_n)$ to calculate the value of the function at the next time step, where h is the time step size and $f(x_n, y_n)$ represents the slope of the function at (x_n, y_n).

Although the Forward Euler Method is very easy to implement and understand, its error increases as the time step size h increases. In dealing with some nonlinear or high-order differential equations, unstable situations may occur.

Basic Idea:
By approximating the derivatives of position and velocity, it allows us to update the cloth's state from one time step to the next. We used explicit Euler method in this problem:

$$\begin{aligned} v^{t+1} &= v^t + \frac{f(t)}{m}\Delta t \\ x^{t+1} &= x^t + v^t \Delta t \end{aligned} \tag{2}$$

2.3 Verlet

Verlet is a numerical method for solving differential equations, particularly in the context of classical mechanics. It is a type of symplectic integrator that conserves energy and momentum, making it very useful for simulating physical systems.

The Verlet algorithm uses position and velocity information to predict the motion of particles over time. It updates the positions of particles using their velocities and the acceleration at the current time step, and then updates the velocities based on the difference in position between the current and previous time steps. This approach is more accurate than some other methods because it takes into account both the current and past states of the system.

One advantage of the Verlet method is that it does not require the use of derivatives or matrix inversions, which can be computationally expensive. It is also relatively easy to implement and has good stability properties. However, it may not be as accurate as some other numerical methods for certain types of problems.

Basic Idea :
By Tayler Formular, we can get the expansion of $x(t + \Delta t)$ and $x(t - \Delta t)$:

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{a(t)\Delta t^3}{2} + \frac{b(t)\Delta t^3}{6} + O(\Delta t^4) \tag{3}$$

$$x(t - \Delta t) = x(t) - v(t)\Delta t + \frac{a(t)\Delta t^3}{2} - \frac{b(t)\Delta t^3}{6} + O(\Delta t^4) \tag{4}$$

By adding 3 and 4, we can get the displacement $x(t + \Delta t)$:

$$x(t + \Delta t) = 2x(t) - x(t - \Delta t) + a(t)\Delta t^2 + O(\Delta t^4) \tag{5}$$

By subtracting 4 from 3, and divided by $2\Delta t$, we can get the velocity $v(t)$:

$$v(t) = \frac{x(t + \Delta t) - x(t - \Delta t)}{2\Delta t} + O(\Delta t^2) \tag{6}$$

In verlet integration, velocity is needed for calculating displacement.

2.4 Runge-Kutta Methods

The Runge-Kutta method, also known as RK method, is a numerical method for solving ordinary differential equations (ODEs). This method is based on the Taylor series expansion and uses a linear combination of different orders of derivatives to estimate the value of the function at the next time step.

Specifically, the Runge-Kutta method discretizes the derivative using finite differences and then uses an iteration formula to calculate the value of the function at the next time step. This method can be of second order, fourth order or higher. The most commonly used variant is the fourth-order Runge-Kutta method.

Compared with other numerical methods, the Runge-Kutta method has higher accuracy and stability, and can handle some complex nonlinear problems. However, it requires more computational time and may not be suitable for real-time applications.

Basic Idea :
Expand $y(x_{n+1})$ at Taylor at point x_n:

$$\begin{aligned} y(x_{t+1}) &= y(x_n) + hy`(x) + \frac{h^2}{2}y``(x_n) + ... + \frac{h^p}{p!}y^p(x_n) + \frac{h^{p+1}}{(p+1)!}y_{(p+1)}(\xi) \\ &= y(x_n) + h\Delta(.) + \frac{h^{p+1}}{(p+1)!}(\xi) \\ &= y(x_n) + h\Delta(.) + O(h^p) \end{aligned} \tag{7}$$

Ignoring the high-order fractions in the above equation and using Delta as the solution format, the result is a p-order algorithm:

$$y(x_{n+1}) = y(x_n) + h\Delta(\cdot) \tag{8}$$

$$\varphi(\cdot) \approx \Delta(\cdot) \tag{9}$$

$$\varphi(\cdot) - \Delta(\cdot) = O(h^p) \tag{10}$$

Define $f(x, y(x)) = y'(x)$

$$\begin{aligned} y(x_{n+1}) - y(x_n) = &\int_{x_n}^{x_{n+1}} f(x, y(x))\, dx \\ \approx &h\sum_{i=1}^{r} c_i f(x_n + \lambda_i h, y(x_n + \lambda_i h)) \end{aligned} \tag{11}$$

We used R-K3 in our simulation:

$$\begin{aligned} y_{n+1} =& y_n + \frac{h}{6}(K_1 + 4K_2 + K_3) \\ K_1 =& f(x_n, y_n) \\ K_2 =& f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2) \\ K_3 =& f(x_n + h, y_n - hK_1 + 2hK_2) \end{aligned} \tag{12}$$

2.5 Errors of the three methods

Algorithm	Local Error
Forward Euler	1.6 x 10^{-2}
Verlet	2.56 x 10^{-4}
R-K	4.096 x 10^{-6}

Table 1: Error Analysis of Three Algorithms

3 Simulation

Figure 3: Forward Euler on cloth simulation

Figure 4: Verlet on cloth simulation

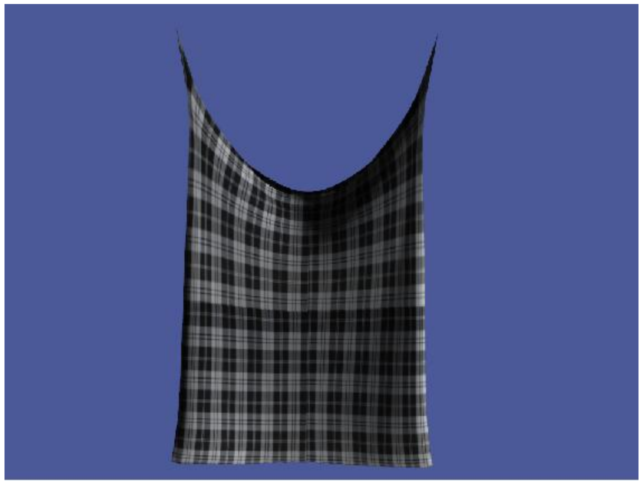

Figure 5: Runge kutta on cloth simulation

We used the Forward Euler method, Verlet algorithm, and Runge Kutta methods to calculate the position, velocity, and acceleration of the cloth mesh at each time step. Among them, velocity and acceleration can be calculated according to Newton's second law and Hooke's law. And then we implement a mesh with 40 rows and 40 columns,width=20,height=20, with stiffness=800.

By numerically integrating the position, velocity, and acceleration of the cloth grid, we can simulate the movement of the cloth under gravity.

In our project, the Forward Euler method performed better than the Verlet. Our Verlet method may cause wrinkles and folds. It is probably caused by incorrect parameter settings. For example, setting the time step too large or too small, insufficient calculation accuracy, unreasonable initial conditions, and so on can affect the accuracy of the results. The Runge Kutta method performs better than the Forward Euler method. It has higher accuracy and therefore leads to a vivid effect on the fabric's behavior.

3.1 Numerical result

Algorithm	Stable Time
Forward Euler	$8s$
Verlet	$3s$
R-K	$15s$

Table 2: stable time of Three Algorithms

4 Conclusion

Cloth simulation is an important problem in the field of computer graphics and computer animation. Forward Euler's method and Verlet's algorithm are both commonly used numerical integration methods that can be used to calculate the position, velocity, and acceleration at each time step. The R-K algorithm is an explicit algorithm that is much more complex than the other two. However, the R-K algorithm can be broken down into multiple stages, each with a different level of precision, so it can provide high computational accuracy. For scenarios such as cloth simulation, the R-K algorithm can be better than other analytical methods in terms of accuracy and result quality.

Overall, the application of the Forward Euler method and the Verlet and R-K algorithm to cloth simulation can help us better understand the basics of physics simulation and computer animation. At the same time, their application also helps to develop efficient physics engines and animation software, providing important support for many industries, such as game development, virtual reality, medical simulation, and scientific research.

5 Teamwork

Zhelin Zhu is responsible for the main part of the code. Tianyue Zhou performed Runge Kutta and Verlet integration methods. Xihe Yu is responsible for the physical model analysis. Both Tianyue Zhou and Xihe Yu are responsible for the main part of the report, and help Zhelin Zhu modify the code.