Cloth Simulation with Large Time Steps

JUSTIN JIANG, McGill University

Code: https://github.com/jjiang99/COMP-559-Final-Project
Video: https://youtu.be/63CZ-WYfjSo

Additional Key Words and Phrases: cloth, numerical integration, collision detection, conjugate gradients, spring forces

ACM Reference Format:
Justin Jiang. 2021. Cloth Simulation with Large Time Steps. 1, 1 (April 2021), 3 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

An extremely popular way to simulate the behaviour of cloth is to use a grid of particles connected by springs. This paper examines the solution presented by Xavier Provot in 1995, augmented with ideas taken from Baraff and Witkin's paper from 1998. Provot's implementation uses standard explicit Euler integration and struggles with stability under certain conditions. As a result, Baraff and Witkin suggest the use of an implicit (backwards) Euler integrator with a conjugate gradient solver, which resolves the stability issues experienced by Provot.

5 distinct test systems were generated to show off the implementation. 3 of the systems deal with object collisions to demonstrate the cloths ability to bend and conform to any shape it is pressed against while the other two show the behaviour of the cloth when its corners are pinned. Wind can be toggled on and off to add another dimension to the simulation.

2 RELATED WORK

Provot [1995] An overview of cloth simulation and structure of particle-mass grids. David Baraff [1998] Advanced numerical integration for stable cloth systems

3 METHODS

Provot recommends the traditional mass-spring grid, but suggests adding additional springs. The standard grid springs are referred to as the structural springs, these are simply unit length vertical and horizontal springs that make up the square grid. In addition to the grid, diagonal springs are added, connecting the corners of the squares, these are referred to as the shear springs as they are present to resist shear forces. Lastly, there are vertical and horizontal springs that connect particles of distance 2 away, again making squares similar to the structural springs, but with side-length 2. These springs are referred to as flexion (bend) springs and their purpose is to simulate bending forces [2]. The Baraff and Witkin implementation is of the same structure but does away with the extra bend springs. Both result in convincing simulations but the extra bend springs does make the cloth noticeably more resistant to bending.

Author's address: Justin Jiang, McGill University, justin.jiang@mail.mcgill.ca.

2021. XXXX-XXXX/2021/4-ART $15.00
https://doi.org/0000001.0000001

The conjugate gradient (CG) solver is used to solve the system of equations representing the force and displacement on any given particle. It solves problems of form: $A\Delta v = b$, where A and b are exactly as defined in equation (16) from Baraff and Witkin [1], and Δv is the velocity update that is required to advance the particle system in time.

The two papers differ in their choice of numerical integrator used to calculate the velocity update of the particles. Provot uses explicit Euler while Baraff and Witkin use implicit Euler. This is significant as stability issues may arise form the Provot implementation. In Provot's paper, he mentions a phenomenon called the "super-elastic" effect. The "super-elastic" effect describes the elongation of the springs that are tied directly to pinned particles, and how they are relatively much more stretched than those that are not, sometimes exceeding a deformation rate of 100%. This is unrealistic in terms of cloth as cloth does not stretch by such a large amount. The obvious solution to this issue is to increase the stiffness of the springs in the particle grid. Although this is a suitable solution, the explicit Euler integrator that is used becomes unstable if the time step Δt exceeds the natural period of the system T_0, unless the time step used is very small. The relation is a follows:

$$T_0 = \pi \cdot \sqrt{\frac{\mu}{K}} \tag{1}$$

Thus a critical stiffness emerges that will render the system unstable if exceeded:

$$K_c = m \cdot T_0^2 / \pi^2 \tag{2}$$

The significance of these conditions is that to achieve a more realistic cloth simulation, the time step must be very small, meaning that the system becomes very slow and visually unrealistic.

Wind was introduced into the system by way of a sine function. A position and time dependent sine function is used to calculate wind forces in both the z and y directions. The forces are added to the force vector prior to the CG solver so that they can be incorporated into the velocity update calculations.

Simple collision detection was implemented to demonstrate the ability of the cloth to deform and match the shape of any objects it is resting on. This is a naive $O(n^2)$ implementation that simply checks the coordinates of the particles against the objects that it is colliding against. When colliding with the sphere, each particle is checked against the center location of the sphere. If the distance of the particle is smaller than the radius of the sphere, the particle is projected back onto the surface of the sphere by calculating the normalized vector that is normal to the plane of the sphere, scaled by the radius, and subsequently added to the position of the colliding particle. Collisions are treated slightly differently for the table system. Particles are still checked against the bounds of the table, but if they are deemed to be colliding, they are projected to the closest of the 5 planes the table. The distance between the particle and each plane is calculated, the minimum one is recorded, and then the particle is projected back to said nearest plane.

4 RESULTS

The Runge-Kutta 4th order method (RK4) was chosen as the explicit integrator to be used for the Provot implementation as it has a larger region of stability compared to the other explicit methods, giving it a higher likelihood for systems to remain stable. The implicit Euler integrators and CG solver were implemented according to Baraff's paper and applied to the Provot particle grid model with expected results and behaviour. The explicit integrator is unstable at high stiffness and the velocities will diverge unless the system is integrated using a very small time step, resulting in a very slow simulation. The implicit integrator solves this issue and it stable at high stiffness while not requiring a small time step, so the simulation is smooth and realistic.

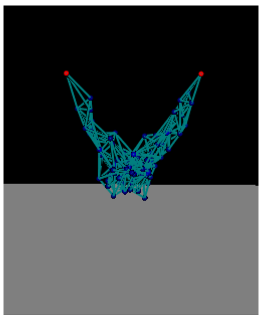

Fig. 1. Test system 3 using RK4 explicit integration

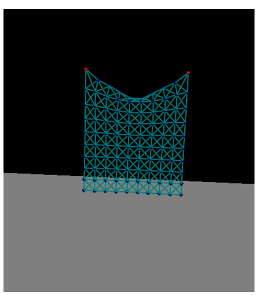

Fig. 2. Test system 3 using implicit integration

The simulation of wind provides another wrinkle to the motion of the cloth, displaying fluttering and the wave propagation that would appear in real life cloths. As seen below, the simulated cloth in wind deforms very similarly to a real cloth held at the top corners and being blown by a fan. The main attributes of note from the real cloth are the sag in the top center edge of the cloth drooping between the two pinned points, as well as the slight convex nature of the rest of the cloth, indicating that the cloth is catching the wind. Both of these features are represented in the simulated model, depicting accurate reproduction.

Collision detection, as expected, slows down when larger cloth systems are tested, since it is a simple $O(n^2)$ implementation. However, the implementation does provide realistic looking collisions that cause the appropriate folds and deformations in the cloth. The

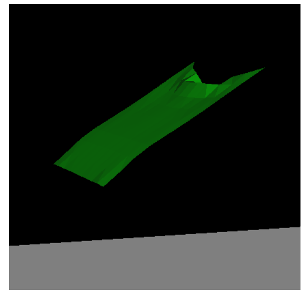

Fig. 3. Test system 3 + wind

Fig. 4. Real cloth in wind

main attributes to note for the sphere are the corners and the flat edges of the cloth. The real cloth is larger than the simulated cloth, relative to the sphere, meaning that the creases are much more pronounced than the simulation. That being said, the simulated creases are still in the expected areas - one in each corner and one along each edge of the cloth (the one along the edge is partially black as the cloth has collided with itself and self-collision was not implemented).

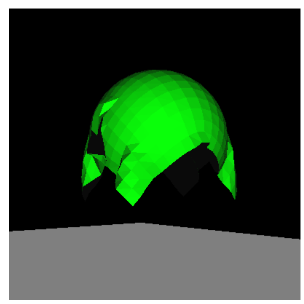

Fig. 5. Test system 5

Fig. 6. Real cloth on sphere

5 CONCLUSIONS

From the experimentation it can be concluded that the suggestions made by Baraff and Witkin should be well taken. The use of an implicit integrator greatly improved the performance and stability of the simulation. In addition to that, the added stability allows for the testing of more extreme types of cloth - whether that means using a very high stiffness, increased spring damping or experimenting with more vigorous external forces. This allows the user to test the boundaries of the simulated cloth without the fear of the instability that was present in Provot's model. The collision detection that was used in this project is sufficient for modelling collisions of a simple object with a small cloth, but is not efficient enough for large systems. A natural extension to this work would be to implement bounding volume hierarchy style collision detection, which is significantly quicker and more suitable if this implementation were to be scaled up. As seen in the system with the sphere, the cloth in this implementation can also fold onto itself, causing some visual artifacts. Therefore to further the realism of the cloth, adding self-collisions is another logical extension.

REFERENCES

A. W. David Baraff. 1998. Large Steps in Cloth Simulation. *SIGGRAPH* (July 1998), 12.

X. Provot. 1995. Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior. (1995), 8.