Soft body simulation using a spring mass particle system and a pressure model

RIDWAN KURMALY AND WEN XUAN WU, McGill University

Code: https://github.com/wenxuan27/Soft-Body-Simulator
Video: https://www.youtube.com/watch?v=1tz_Blu0F6k

Additional Key Words and Phrases: soft body simulation, numerical integration, collision detection, pressure model, spring-mass model

ACM Reference Format:
Ridwan Kurmaly and Wen Xuan Wu. 2021. Soft body simulation using a spring mass particle system and a pressure model. 1, 1 (April 2021), 3 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

In this project, we introduce a soft body simulator using both a spring-mass truss structure implementation and a pressure model implementation. The pressure model was implemented using a simplified version of the empirical ideal gas pressure equations.

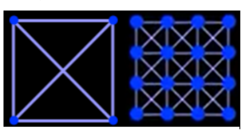
2 SPRING-MASS MODEL

The most basic component of our soft body model is the spring mass model. Here, the idea is the simulate ideal weightless springs with mass particles at both ends following Hook's law: $F_s = ks$.

We can also apply to it the gravitational force: $F_g = mg$ And then combine all forces acting on a particle into one equation to integrate: $\sum F_i = m_i \frac{\partial^2 r_i}{\partial t^2}$

3 PARTICLE SYSTEM

Using the basic component of the spring-mass model, we can expand it into particle systems forming truss structures. This is what serves as the basic building blocks of our soft body model. We form 4 by 4 "grids" of spring-mass systems which are completely connected.



Here the triangular structures formed by the diagonal springs give the structure enough rigidity to not crumble and collapse. Using this basic component as a building block, we can make larger truss structures of any size.

4 COLLISION DETECTION AND RESOLUTION

For our purposes, we used a simple line-circle collision detection to detect collisions between mass particles and the ground. This works fairly well as the primitives of our systems are simple rigid bodies. The ground is simply a flat surface and the spring mass system is simply composed of particles which are here represented using circles. Distance between a line $q + l$ and a point p: $d = (p - q) \bullet l$

Author's address: Ridwan Kurmaly and Wen Xuan Wu, McGill University, ridwan.kurmally@mail.mcgill.ca,wen.x.wu@mail.mcgill.ca.

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

Once we detected a collision between the particle and the ground, we can resolve it by simply applying a reflection by the normal vector of the surface that the particle collided with at the point of contact. In addition, to prevent any overlap, we can also reflect the current position of the particle by the normal vector of the ground at the point of contact. For collision detection between mass particles in a particle system, we simply used a circle-circle collision detection.

5 PRESSURE BASED METHOD FOR SOFT BODIES

We applied a similar idea as in "Pressure Model of Soft Body Simulation" [Matyka et al. 2003].

5.1 Particle System of the pressure model

In the pressure model, truss structures are unnecessary. Instead, we want our particle system to form a simple enclosed polygon similar to a ball where springs act on the particles from the outside compressing the polygon and the pressure inside would apply forces from the inside to expand the polygon.

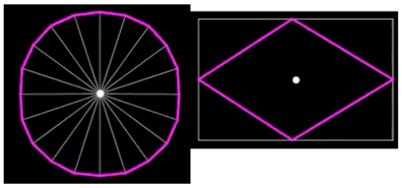
5.2 Ideal gas Law

In thermodynamics, we have an empirical equation to approximate the state of a hypothetical ideal gas. This approximation is good under many conditions and for many gases. $PV = nRT$ Where P, V, T, n and R are the pressure, volume, temperature, amount of substances (in moles) and the ideal gas constant respectively. We can easily derive this equation into $P = V^{-1}nRT$ And set n and T to be constants to get the pressure. Since we are working in 2d, we would like to use an equivalent equation for our flatland world. For that we can simply switch the volume V to the area A. $P = A^{-1}nRT$ For our purposes, we considered nRT to simply be a constant coefficient and then introduced a new scaling factor k variable to linearly scale the pressure as needed giving us: $P = A^{-1}k\ nRT$

5.3 AREA APPROXIMATION

To calculate the area of the enclosed polygon shape of our particle system at each frame, we considered different methods. First, a bounding box method where we approximate the area of particle system using a 2D rectangle formed by the minimum and the maximum x and y values of all the particles in the system. We then use the area of the rectangle to approximate the area of the shape. $A_{rectangle} = b \bullet h$

Our second method consists of using triangulating the enclosed polygon by adding a "center" point and connection every particle of the polygon with this center point to form triangles that we can then use to calculate the area. The center point here is simply defined by the center of the bounding box of the enclose polygon (halfway point in either direction). The area of a triangle defined by the points A, B and C is: $A_{A,B,C} = \frac{1}{2}\left\|\bar{AB} \times \bar{BC}\right\|$



We can sum of the area of all the triangles to approximate the area of the enclosed polygon. This technique is very accurate when the polygon is convex, however when it becomes concave which can happen when the polygon is deformed during a collision especially when the "center" point ends up being outside of the polygon. In this case, it is actually the bounding box area method is preferred. However, in most case, the enclosed polygon would not suffer that intense deformations and thus the triangulation area method is preferred overall.

5.4 Pressure Forces

For our purposes, the pressure force is a force acting on a surface in the direction of the normal of the surface. In our model, only particles experience forces. We used a simplified 2D formula to get the exact pressure force acting on each particle. $F_p = PL$ Where L is simply the length of the one-dimensional surface that the pressure is acting on.

6 METHODS

6.1 NUMERICAL METHOD

For the integration of the motion of particles in the spring mass systems, and the pressure-based systems, we have used an RK4 integrator. It appears to have been very stable, although smaller time steps have been needed with parameter configurations with a particularly high spring constant. All results presented here have been calculated using RK4.

6.2 Algorithm

Here is the general algorithm that we used to compute the solutions at each time step of the simulation:

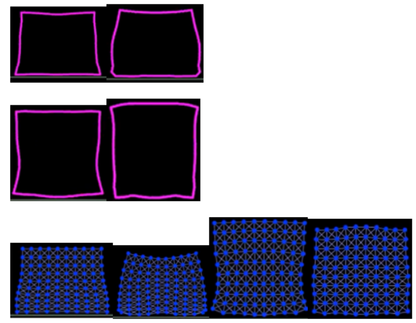
For each particle of the particle system: Accumulate the forces of gravity and of the springs acting on the particle.

Then, we compute the area of the enclosed polygon formed by the particle system using either the bounding box approximation or the exact triangulation method.

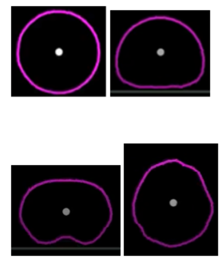
For each edge e of the enclosed polygon: Compute the length of the edge. Compute the pressure force F_p acting on the surface of the edge using the computed length. Accumulate F_p onto both particles that the edge e is connected to. Integrate the equations using our RK4 integrator. Detect and resolve collisions. Update the positions of the particles. *step 2 and 3 only applies to soft bodies particle systems that uses the gas pressure model. For the other model, simply run the entire algorithm without step 2 and 3.

7 RESULTS

Truss structure spring-mass model :



Spring-mass with Pressure Model



The results are fairly accurate. We can clearly see that in both models, the soft bodies went through severe deformations and still reformed to their original shape. However, in the truss structure based model, when the impact velocity is too high, it can lead to the shape collapsing on itself. To deal with this issue, we simply added particle-particle collisions. The RK4 integrator was able to get us fairly stable solutions, although, it did struggle when the spring constants were set very high. To counter that, we simply lowered the step size.

7.1 Rendering

Once the physics computations have been solved, it becomes fairly trivial to get the rendering. We can simply get all the of the outer particles of the particle systems and use those to create an outline of our soft body object. Or we could even simply get all of the particles as a point cloud. Then, we could simply give this soft body object a material and render it with the material. For our purposes, we simply kept it at an outline of the shapes as this paper is not focused on rendering techniques.

8 CONCLUSION

Comparing the truss structure-based spring-mass model with the pressure-based spring-mass model, we can see that the first is more suited for simulating soft bodies which are made of discrete entire pieces whereas the second model is more suited for simulating soft bodies made of an elastic outer layer containing either a liquid or a gas inside. The second model is also more suited for simulation of rounded and circular shaped soft bodies whereas the first is more for soft bodies that are formed with polygons (or polyhedrons in 3 dimensions).

9 REFERENCES

Zhang, Zhou. (2020). Soft-Body Simulation With CUDA Based on Mass-Spring Model and Verlet Integration Scheme. 10.1115/IMECE2020-23221.

Maciej Matyka, Mark Ollila. (2003). Pressure Model of Soft Body Simulation [https://ep.liu.se/ecp/010/007/ecp01007.pdf]