香 港 大 學
THE UNIVERSITY OF HONG KONG

# Physics-based animation

## *Tutorial: Simulating cloth*

Floyd M. Chitalu & Prof. Taku Komura

# Overview

- Cloth dynamics simulation
- Extras
  - Alternative numerical integrators
  - User input
  - Handling collisions
- Resources

# Cloth dynamics simulation

- We will model cloth using a mass-spring system.
- Our (simple) system will therefore have three types of forces
  - Spring stiffness
  - Spring damping
  - Gravity

# Spring force due to stiffness

$$\mathbf{f}_p = k \left( \frac{\|\mathbf{x}_q - \mathbf{x}_p\|}{r} - 1 \right) \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}.$$

# Spring force due to damping

$$\mathbf{f}_p = k_d \left( \frac{\mathbf{v}_q - \mathbf{v}_p}{r} \cdot \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|} \right) \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$

# Total forces along a spring

$$\mathbf{f}_p = \left[ k_s \left( \frac{\|\mathbf{x}_q - \mathbf{x}_p\|}{r} - 1 \right) + k_d \left( \frac{(\mathbf{v}_q - \mathbf{v}_p) \cdot (\mathbf{x}_q - \mathbf{x}_p)}{r\|\mathbf{x}_q - \mathbf{x}_p\|} \right) \right] \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$

# Assigning spring force

$$\mathbf{f}_q = -\mathbf{f}_p$$

# We also have gravity!

- Applied per vertex/node of the mass-spring system

$$\mathbf{f}_g$$

# Updating the simulation

- The simulation state at the current timestep

$$\mathbf{v}_p(t)$$

$$\mathbf{x}_p(t)$$

# Updating the simulation

- The simulation state at the next timestep

$$\mathbf{v}_p(t + \Delta t)$$

$$\mathbf{x}_p(t + \Delta t)$$

# Updating the simulation

- Here is *one way* compute the state of the next timestep

$$\mathbf{v}_p(t + \Delta t) = \mathbf{v}_p(t) + \Delta t \cdot \frac{\mathbf{f}(\mathbf{x}_p, t)}{m_p}$$

$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + \Delta t \cdot \mathbf{v}_p(t + \Delta t)$$

"Symplectic Euler"

# Here is what you can do.

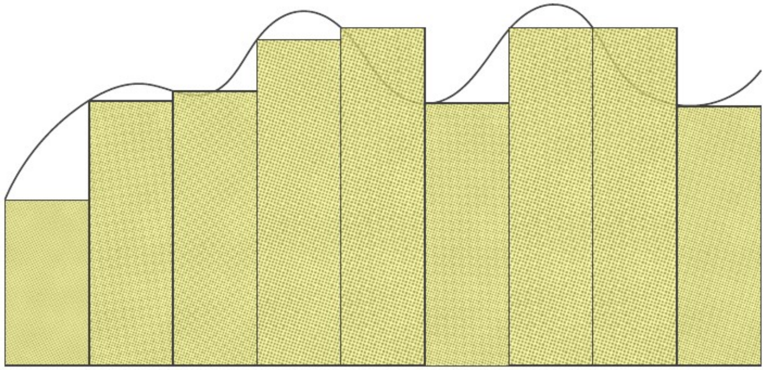
```
for Particle p : particles do
    p.frc = 0
    p.frc += p.mass*gravity
end for
for Spring s : springs do          → j - i
    Vec3 d = particles[s.j].pos - particles[s.i].pos
    double l = mag(d)
    Vec3 v = particles[s.j].vel - particles[s.i].vel
    Vec3 frc = (k_s*((l / s.r) - 1.0) + k_d*dot(v / s.r, d / l)) * (d / l)
    particles[s.i].frc += frc
    particles[s.j].frc -= frc
end for
for Particle p : particles do
    p.vel += dt*(p.frc / p.mass)
    p.pos += dt*(p.vel)
end for
```

# Demo ...

# Try other integrators too

- Forward (Explicit) Euler

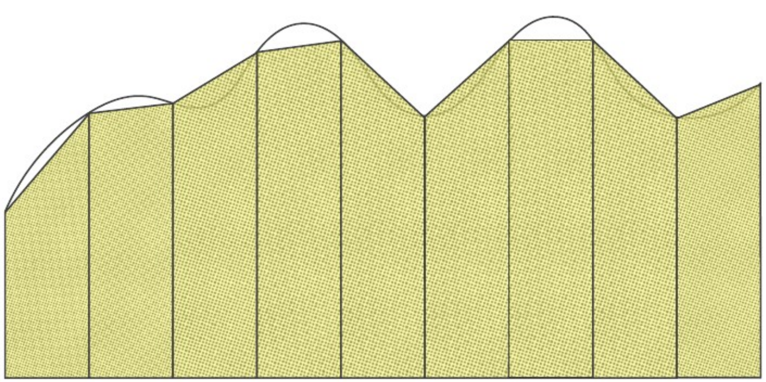
$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + \Delta t \cdot \mathbf{v}(\mathbf{x}_p, t)$$



# Try other integrators too

- Trapazoidal Rule

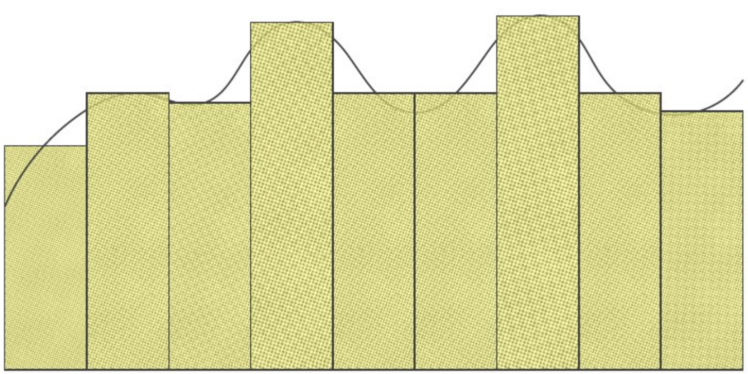
$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \frac{\Delta t}{2}\left(\mathbf{v}(\mathbf{x}(t), t) + \mathbf{v}(\mathbf{x} + \Delta t\mathbf{v}(\mathbf{x}, t), t)\right)$$



# Try other integrators too

- The mid-point method

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \Delta t \left( \mathbf{v}\left(\mathbf{x} + \frac{\Delta t}{2}\mathbf{v}(\mathbf{x}, t), t\right) \right)$$



# Try other integrators too

- … and more
  - See Tuesday's lecture notes

# Other things to try

- User interation
- Collision processing
- Multiple cloths
- Implicit integration ...

# Getting started ...

- Open the zip file provided to get started.
- You need to
  - know C++ (a bit)
  - Know how to configure CMake
  - ...
  - Glance through the README file for instructions

# Additional resources


- Adam W. Bargteil, Tamar Shinar, and Paul G. Kry. 2020. An introduction to physics-based animation. In SIGGRAPH Asia 2020 Courses (SA '20). Association for Computing Machinery, New York, NY, USA, Article 5, 1–57. DOI: https://doi.org/10.1145/3415263.3419147
- Matthias Müller, Jos Stam, Doug James, and Nils Thürey. 2008. Real time physics: class notes. In ACM SIGGRAPH 2008 classes (SIGGRAPH '08). Association for Computing Machinery, New York, NY, USA, Article 88, 1–90. DOI: https://doi.org/10.1145/1401132.1401245