CS452 Computer Graphics

Murtaza Taj

murtaza.taj@lums.edu.pk

Lecture 22: Particle Systems

Thurs, 14th Nov 2019

References

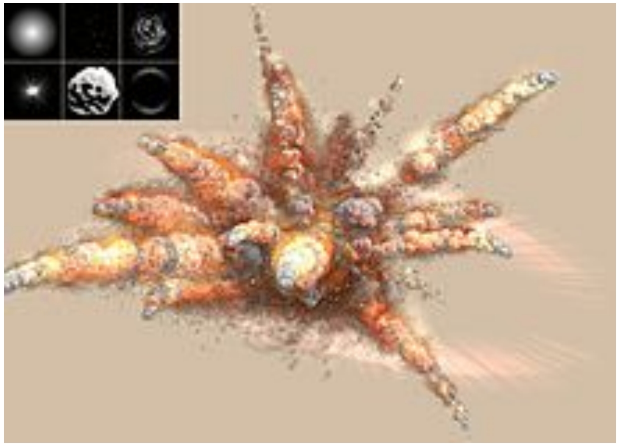
- Adrien Treuille, CSD 15-869 : The Animation of Natural Phenomena, CMU
- http://www.cs.cmu.edu/~15869-f10/index.html

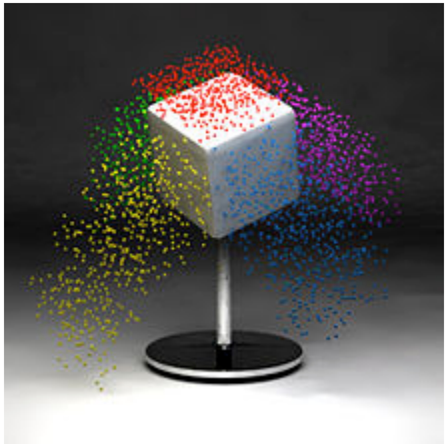
The Animation of Natural Phenomena

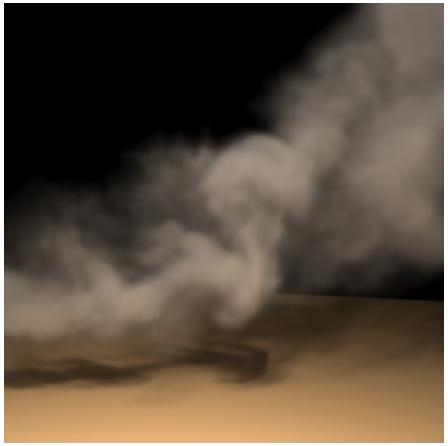
Number:	CSD 15-869
Location:	GHC 4215
Instructor:	Adrien Treuille
TA:	Jeehyung Lee
Office Hours:	Mondays 3-4pm (Smith 232)
Time:	TR 3:00 - 4:20pm (Starts Sept 8!)

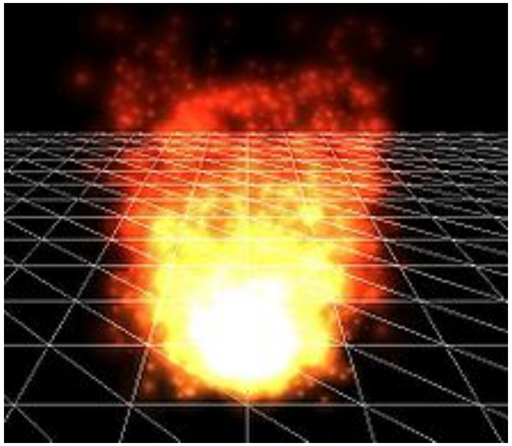
Particle Systems

- Fire
- Hairs
- Cloths
- Fluids
- Smoke





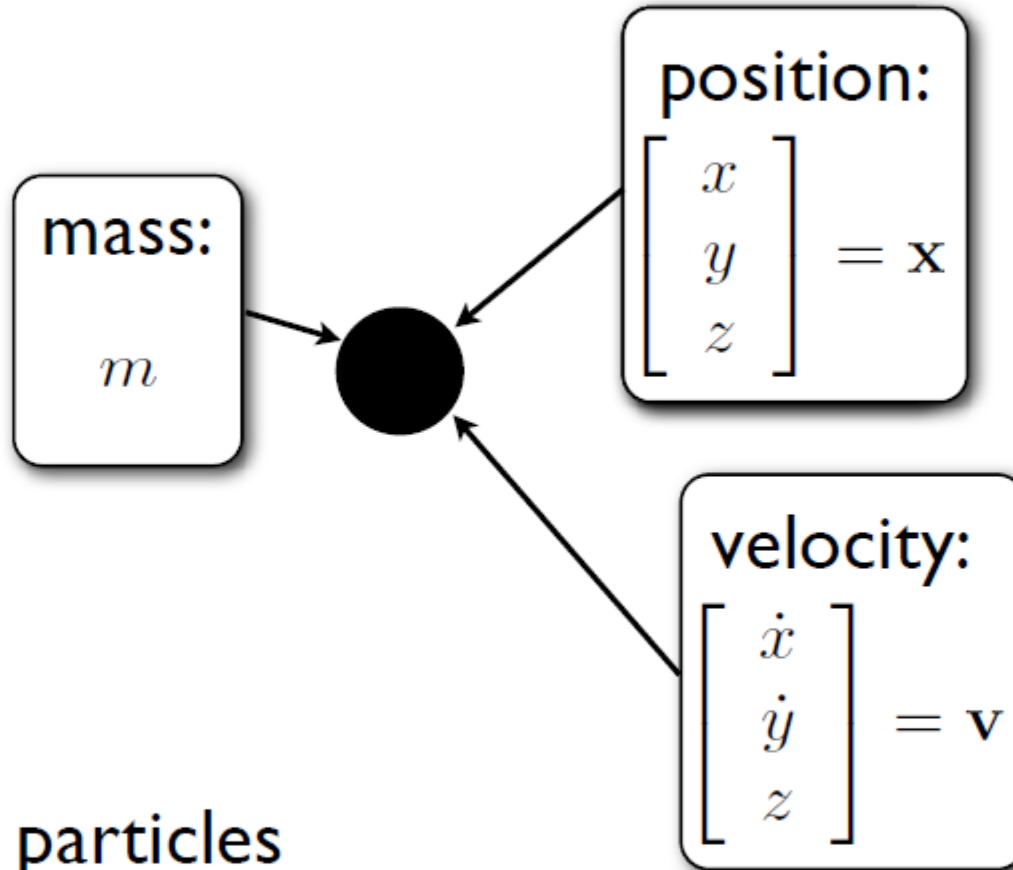




Particle System - Examples

- Demo video - SIGGRAPH 2010

What is a particle?



- $\therefore$ we can’t change a particles velocity, only it’s *acceleration*
  - *FORCES!*

Newtonian Particle

- Differential equations: f=ma
- Forces depend on:
- Position, velocity, time

$$\ddot{x} = \frac{f(x, \dot{x})}{m}$$

Second Order Equations

$$\ddot{x} = \frac{f(x, \dot{x})}{m}$$

Has 2nd derivatives

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= \frac{f(x, \dot{x})}{m} \end{aligned}$$

Add a new variable v to get a pair of coupled 1st order equations

Particle Structure

Position in phase space

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{v} \\ \mathbf{f} \\ m \end{bmatrix} \begin{array}{l} \leftarrow \text{position} \\ \leftarrow \text{velocity} \\ \leftarrow \text{force accumulator} \\ \leftarrow \text{mass} \end{array}$$

Differential Equation Solver

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ f/m \end{bmatrix}$$

Euler method: $x(t+\Delta t) = x(t) + \Delta t\,\dot{x}$

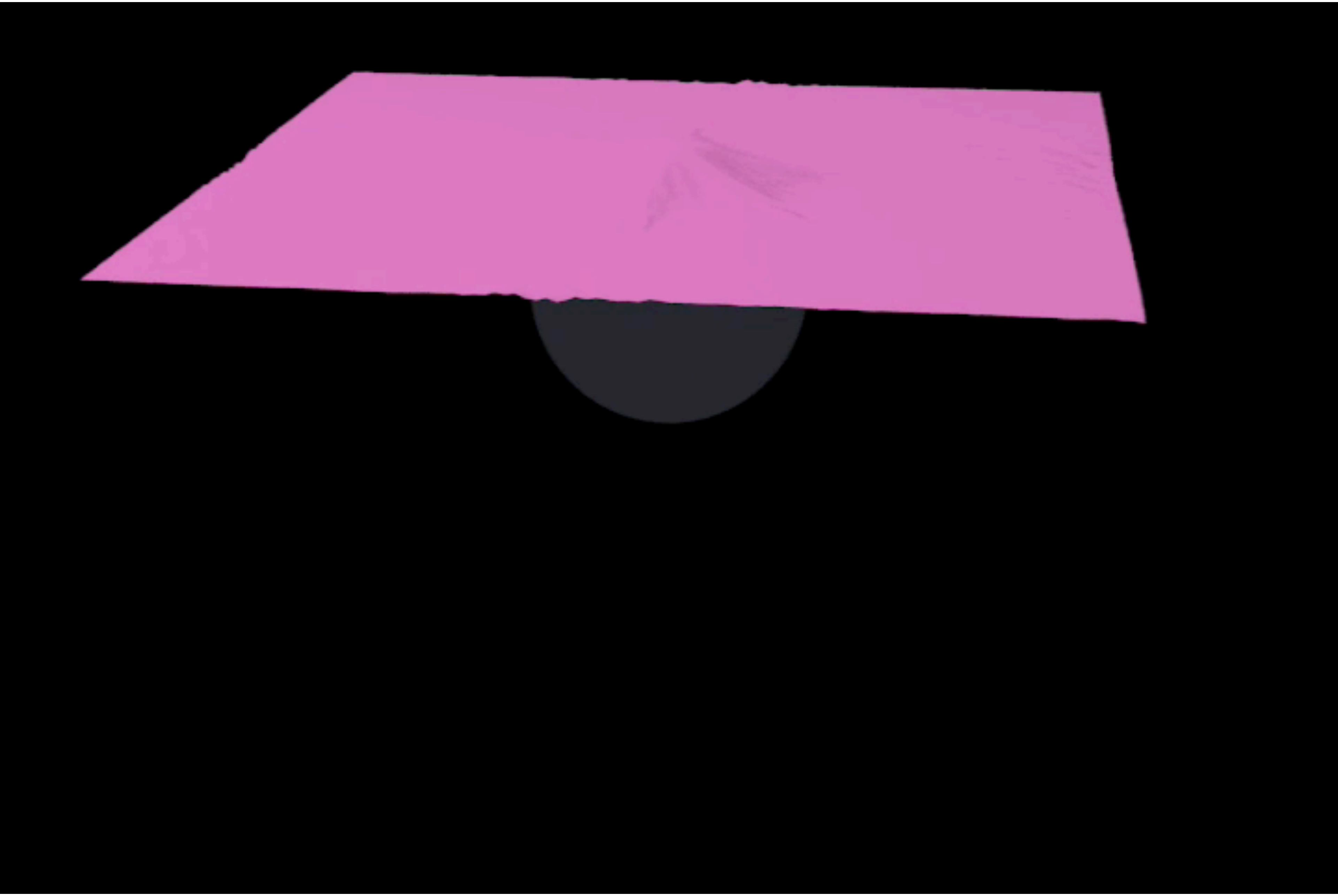
$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta t\,\dot{x}$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \Delta t\,\dot{v}$$

Higher order solvers perform better: (e.g. Runge-Kutta)

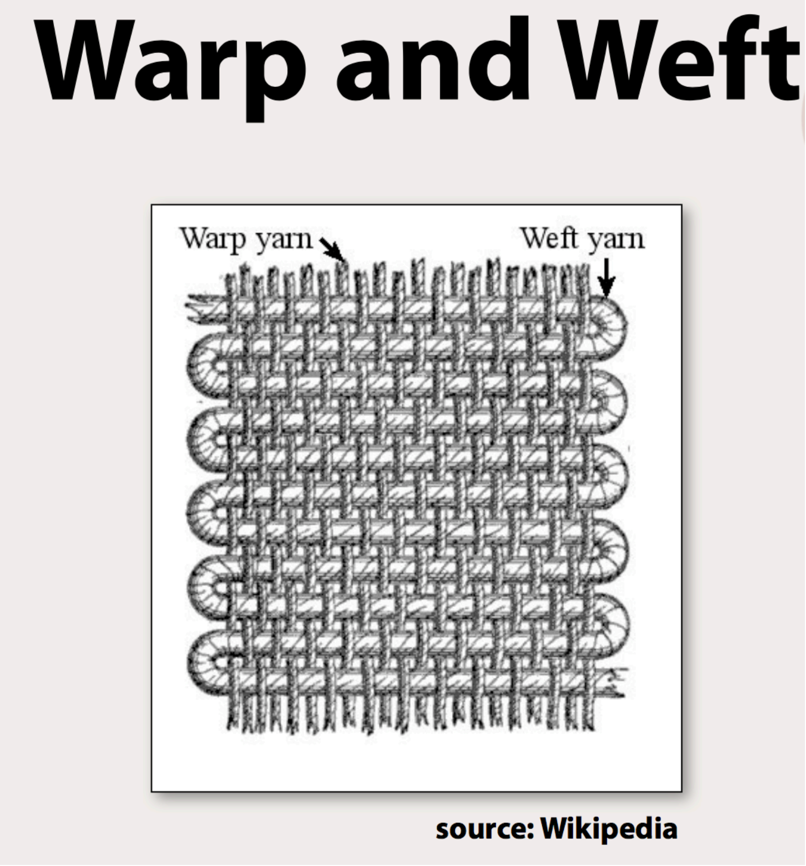

What is cloth?
Two basic types...
Woven
Knit

Knitted Cloth



What is a Cloth?

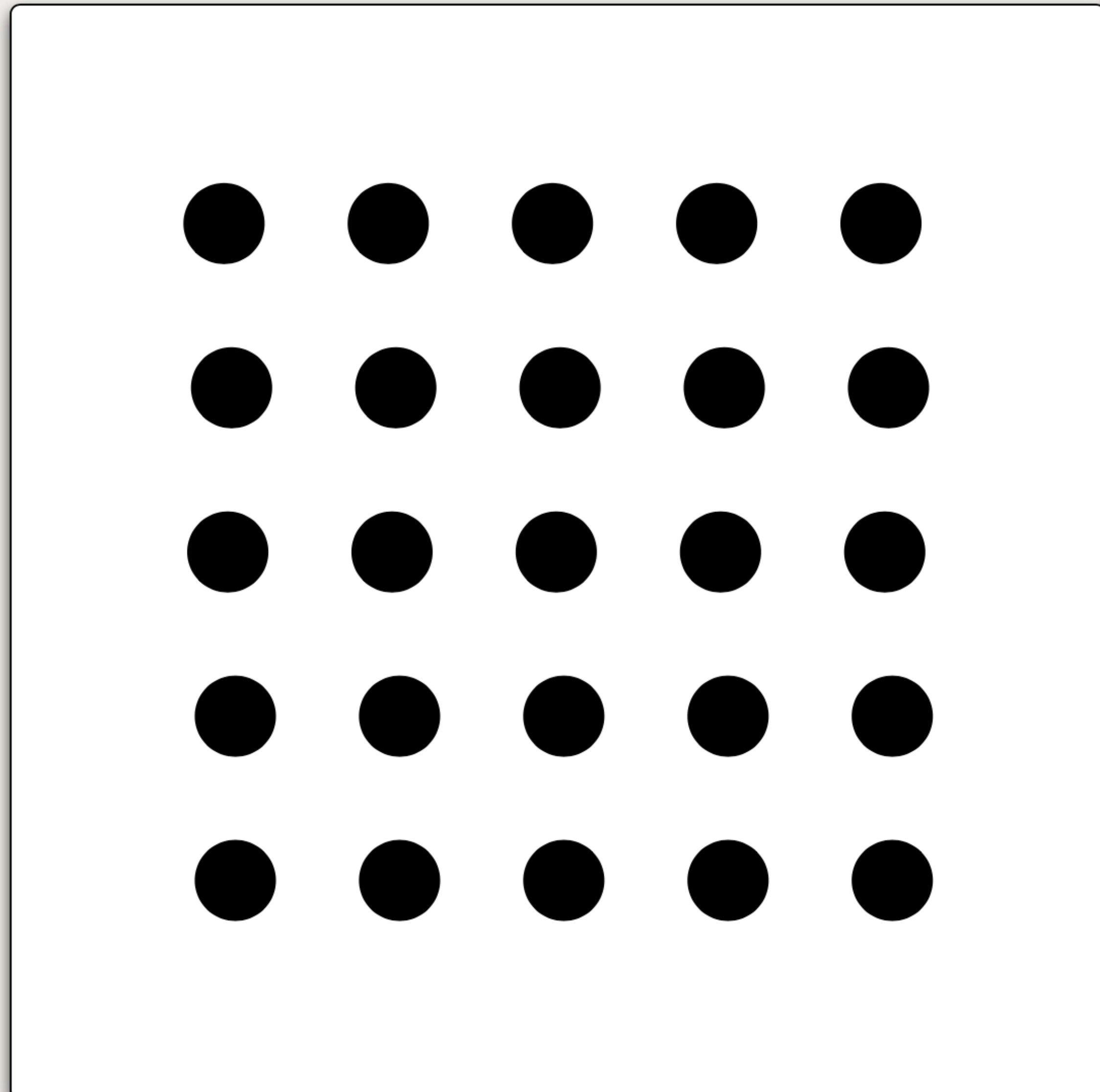
- 2 basic types: woven and knitted
- We'll restrict to woven cloth
  - Warps and Weft



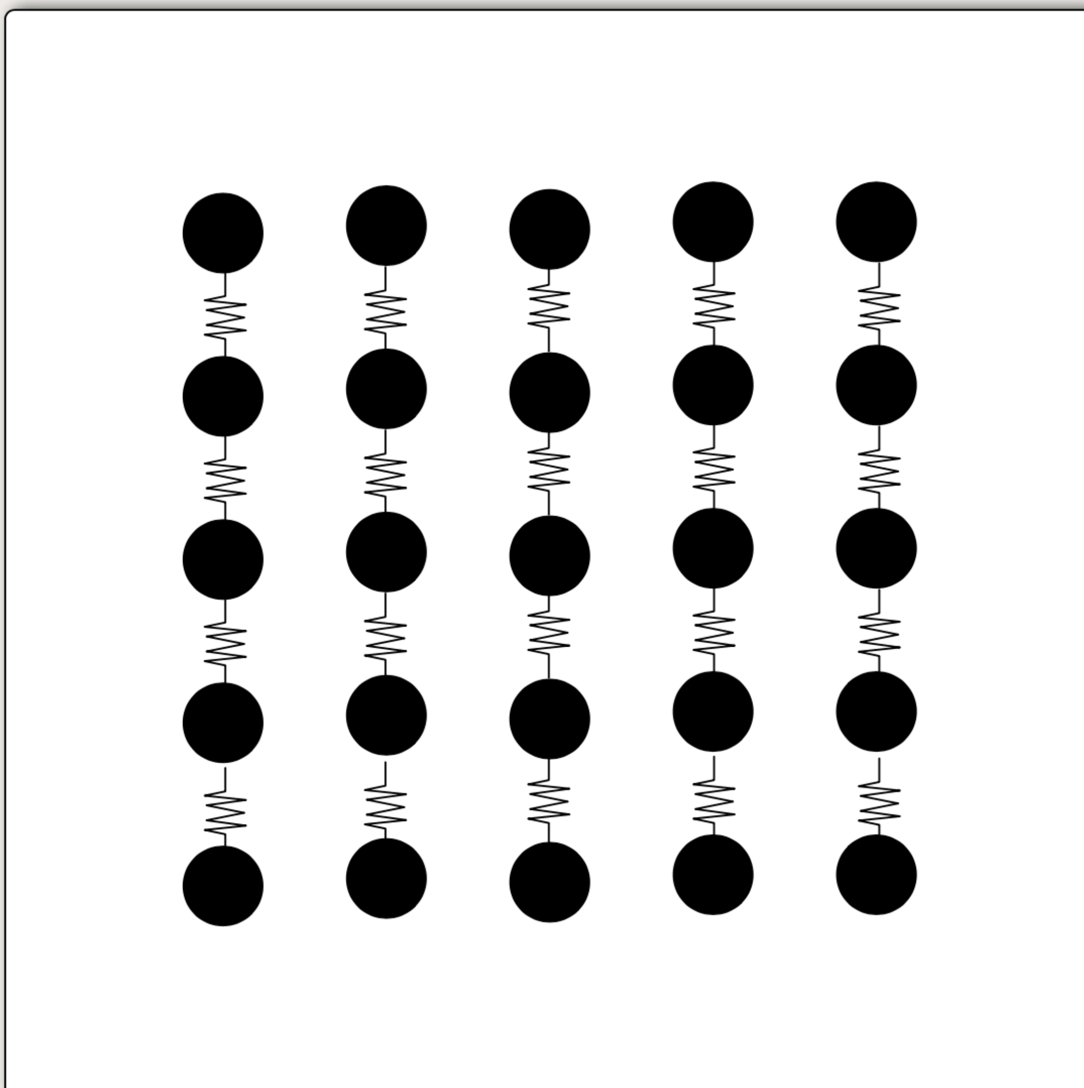
Cloths has resistance to ..

- Stretching
- Shearing
- Bending

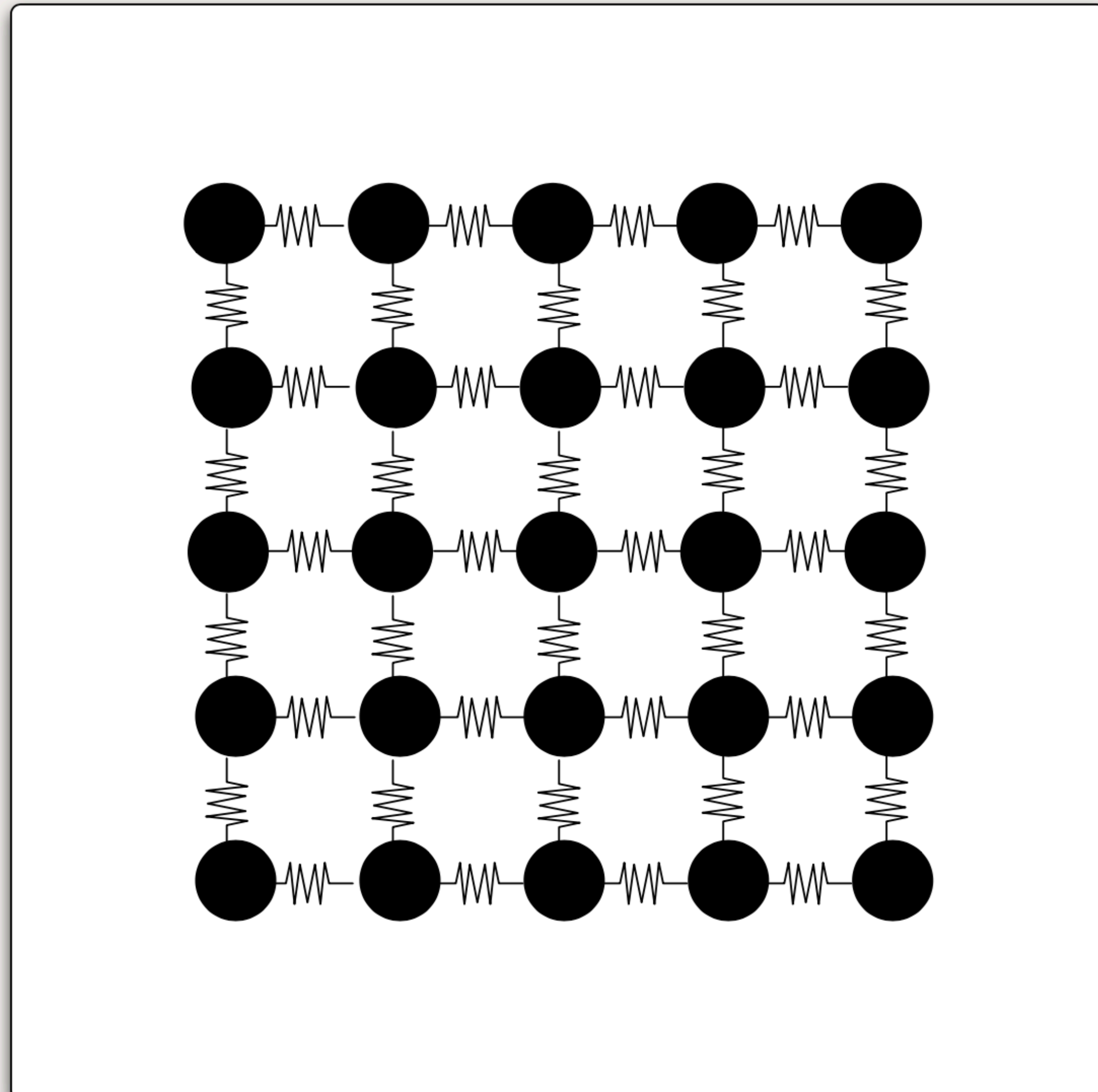
Basic Model



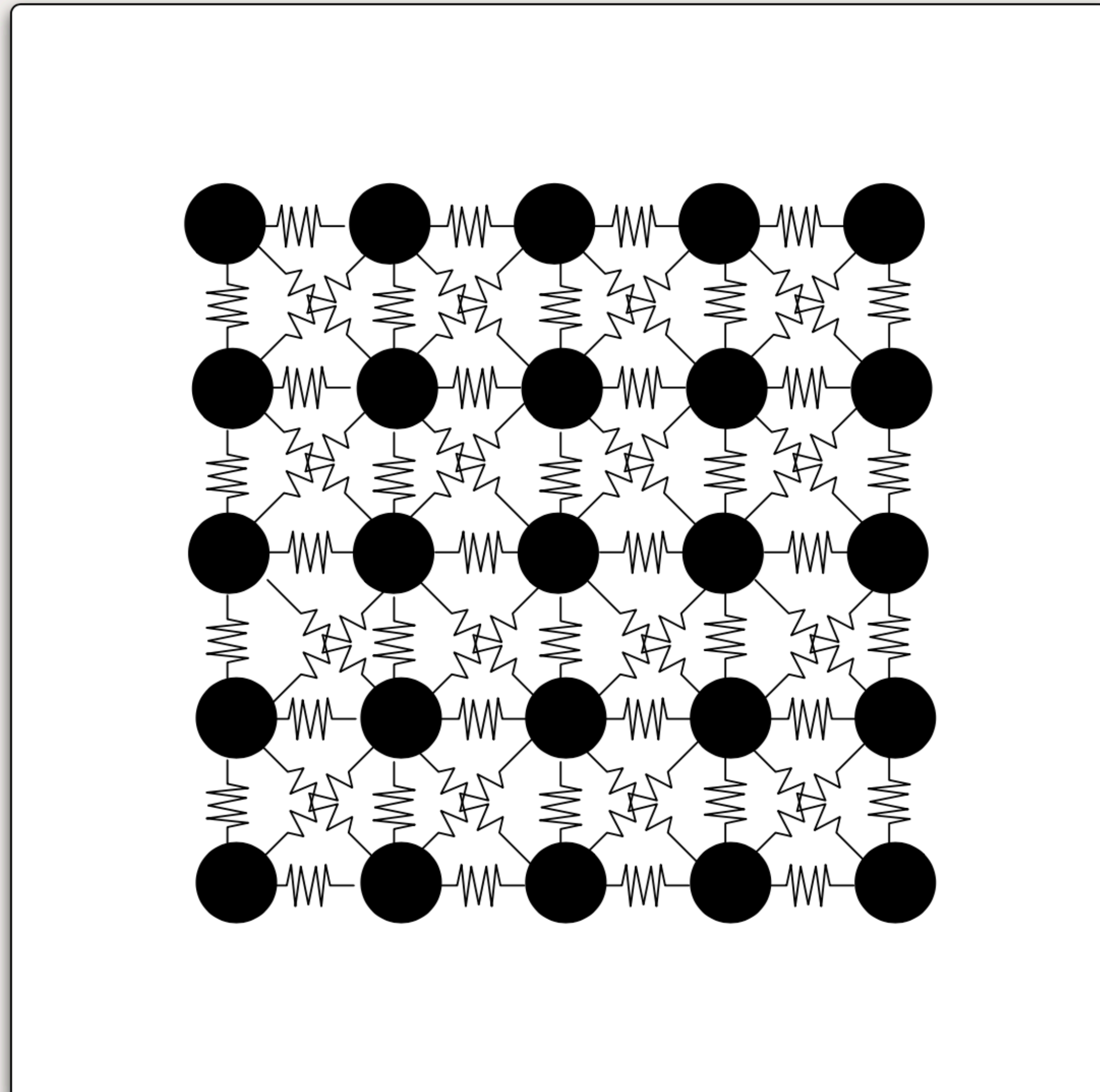
Warp Springs



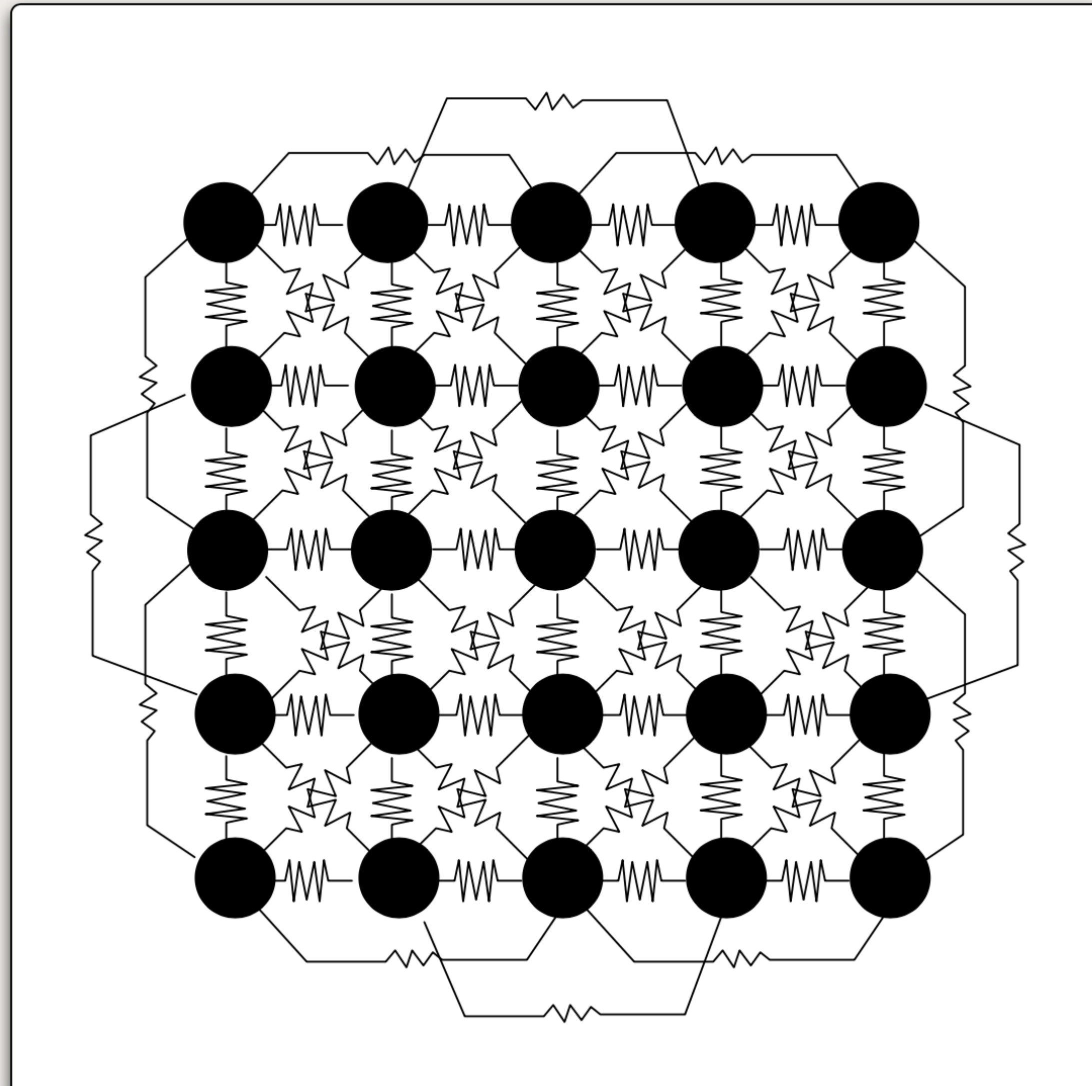
Weft Springs



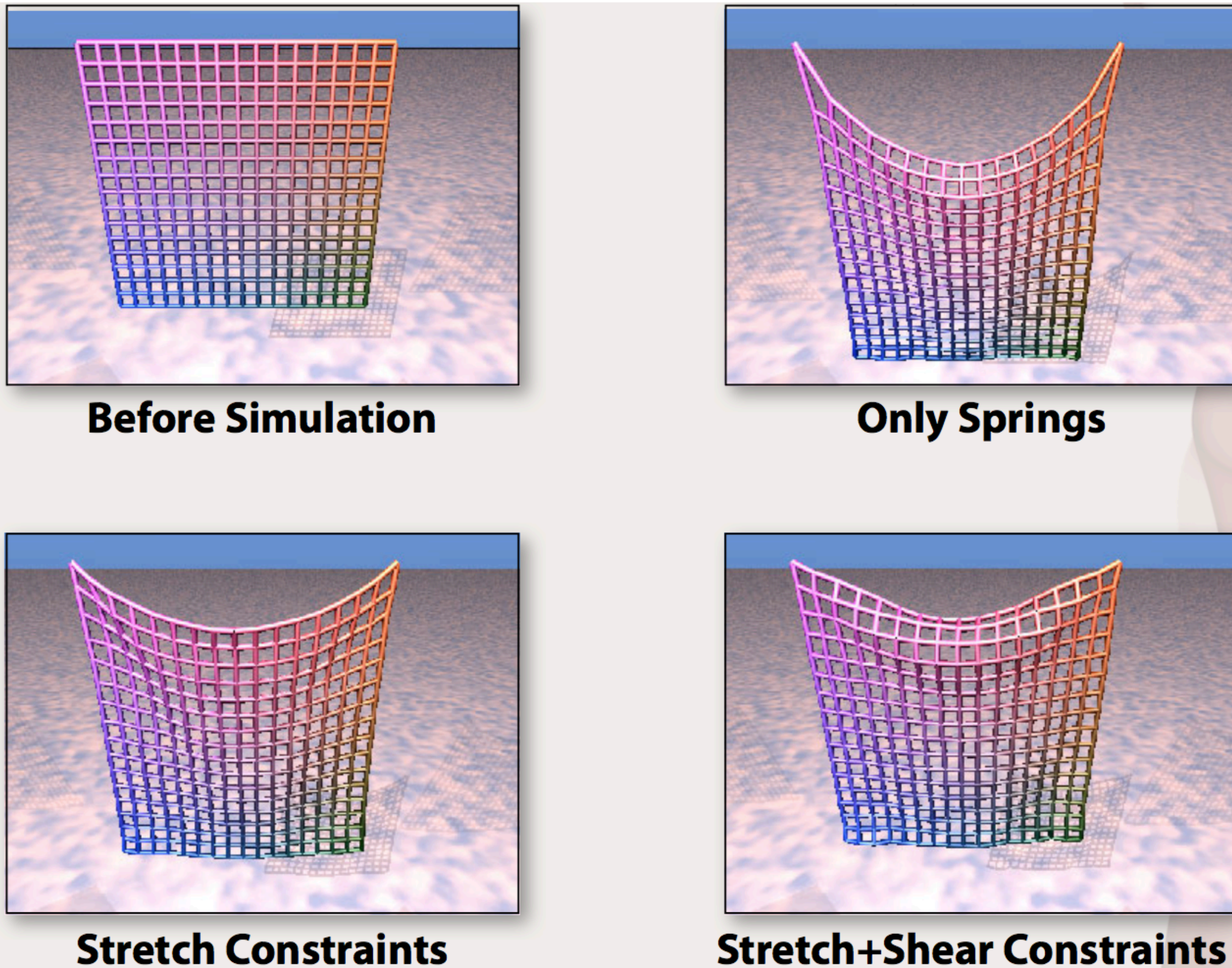
Shear Springs



Bend Springs



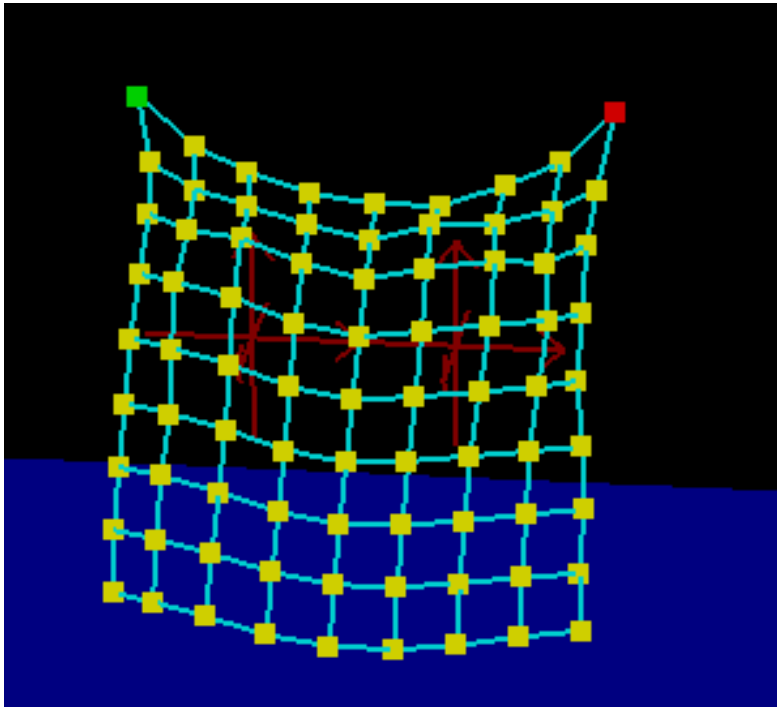
Springs vs. Constraints

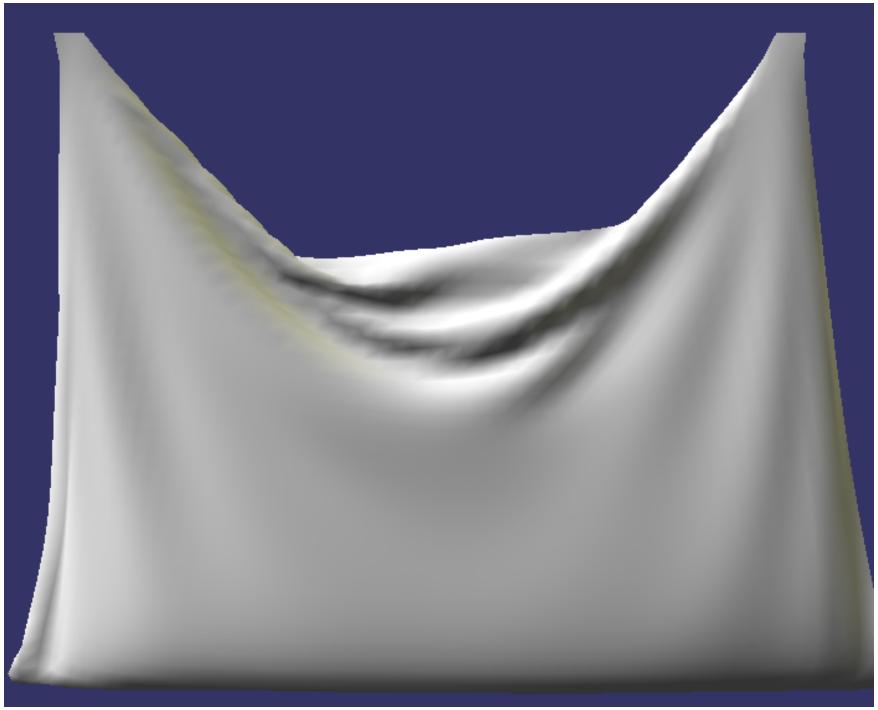


Source: Xavier Provot
Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior

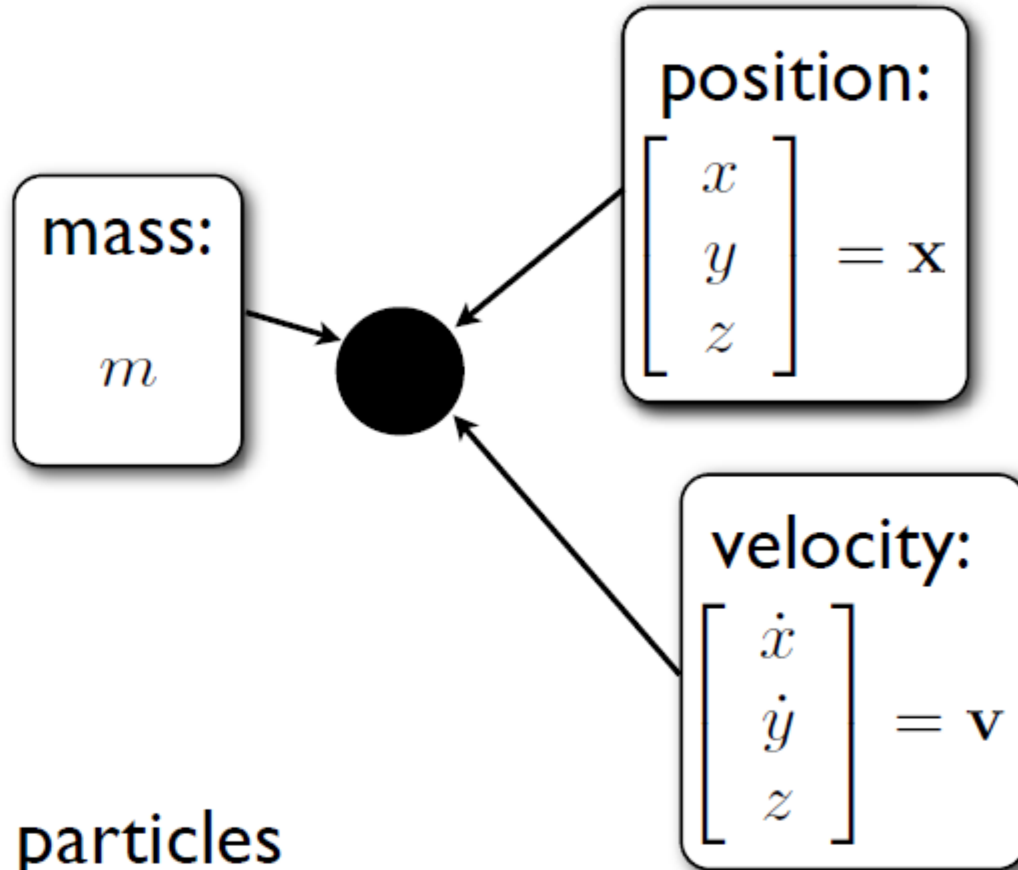
Woven Cloth - Mass Spring System

- Forces
  - Internal: Springs
  - External: Gravity, wind, collision





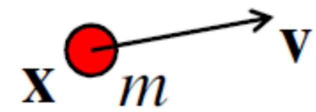
What is a particle?



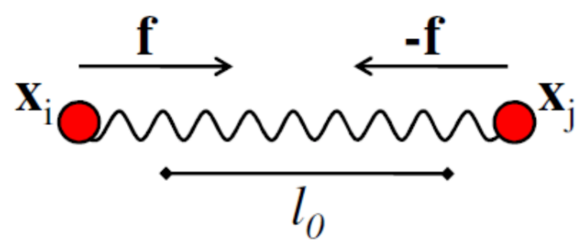
- $\therefore$we can’t change a particles velocity, only it’s *acceleration*
  - *FORCES!*

Mass Spring Physics

Mass point: mass m, position x, velocity v



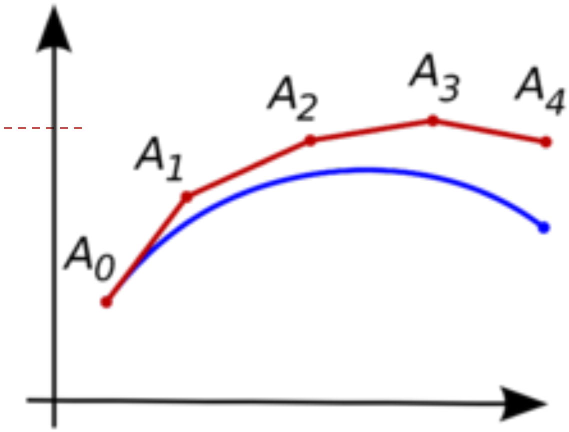
Spring motion based on Hooke's Law
(Linear Strain model)



$$F_s = k_s * (||p_a - p_b|| - l)$$

where k_s is the spring constant
p_a , p_b are the position of point mass
l is the rest length

Time Integration: Explicit Euler



ODE (Newton):

$$\dot{\boldsymbol{v}} = \frac{\boldsymbol{f}}{m}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{v}$$

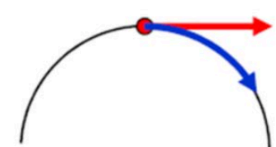
Using Explicit Euler Integration:

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \Delta t \boldsymbol{v}_i^t$$

$$\boldsymbol{v}_i^{t+1} = \boldsymbol{v}_i^t + \Delta t \frac{1}{m_i} \sum_j \boldsymbol{f}(\boldsymbol{x}_i^t, \boldsymbol{v}_i^t, \boldsymbol{x}_j^t, \boldsymbol{v}_j^t)$$

Assumes velocity and force constant within timestep Δt

- Correct solution would be $\boldsymbol{x}(t = \Delta t) = \boldsymbol{x}(t) + \int_t^{t+\Delta t} \boldsymbol{v}(t) dt$



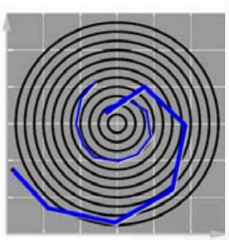
Explicit Integration Issues

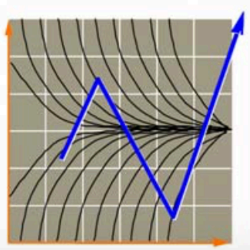
Accuracy

- Can be improved with higher-order implicit schemes e.g. runge-kutta; or by decreasing size of time-steps (usually at the cost of efficiency)
- Not always critical in real-time applications

Stability

- Leads to over-shooting
- Critical for real-time applications e.g. games





Verlet Integration

- Verlet integration computes a point mass's new position at time $t+dt$ as follows:

$$x_{t+dt} = x_t + v_t * dt + a_t * dt^2$$

where x_t is the spring constant
v_t is the current velocity
a_t is the current total acceleration from all forces
d_t is a time step

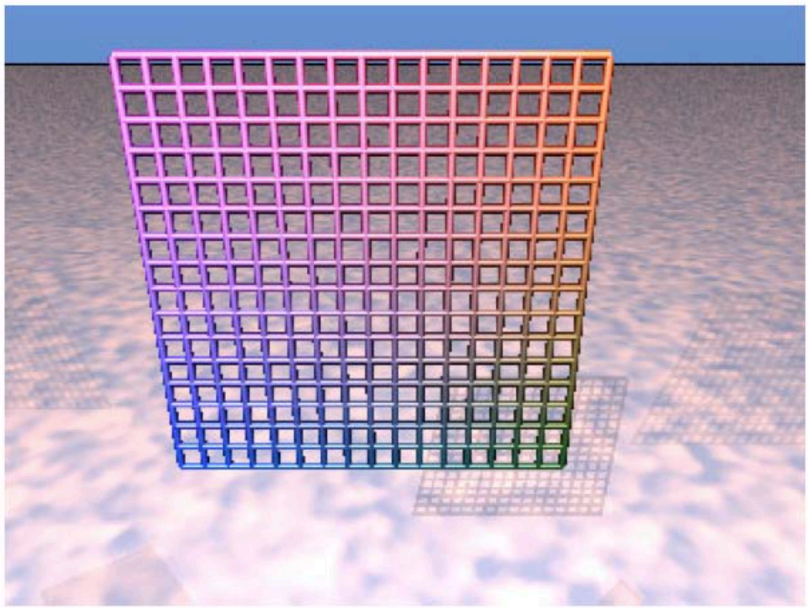
- In Verlet integration, we approximate $v_t * dt = x_t - x_{t-dt}$

$$x_{t+dt} = x_t + (x_t - x_{t-dt}) + a_t * dt^2$$

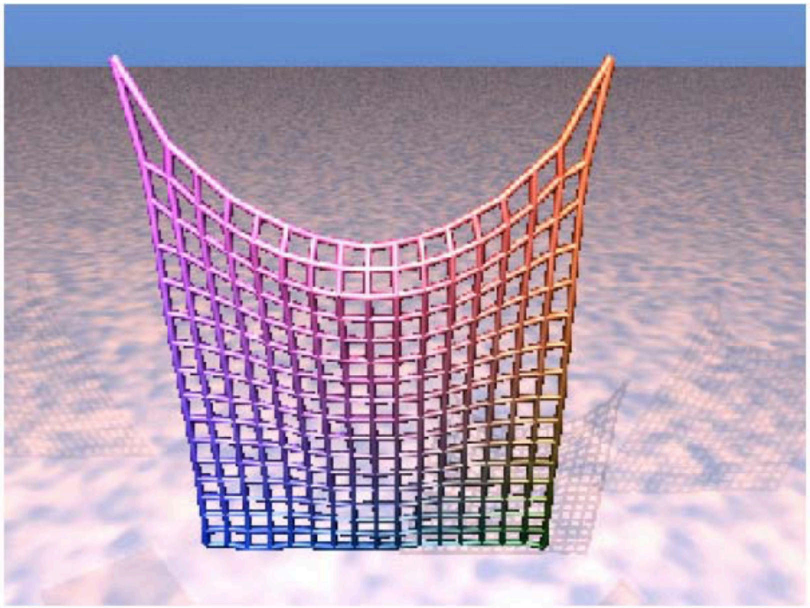
- We can add some dampness d to help simulate loss of energy

$$x_{t+dt} = x_t + (1 - d) * (x_t - x_{t-dt}) + a_t * dt^2$$

Problems with basic model



(a) Initial position



(b) After 200 iterations

Problems:

- Edge springs stretch more than other springs: locally concentrated deformation (super-elasticity) ~ unrealistic, high oscillation
- Hard to associate constants (e.g. bend) with real physical parameters

Increasing Stiffness

Obvious solution to reducing super elastic effect but has problems

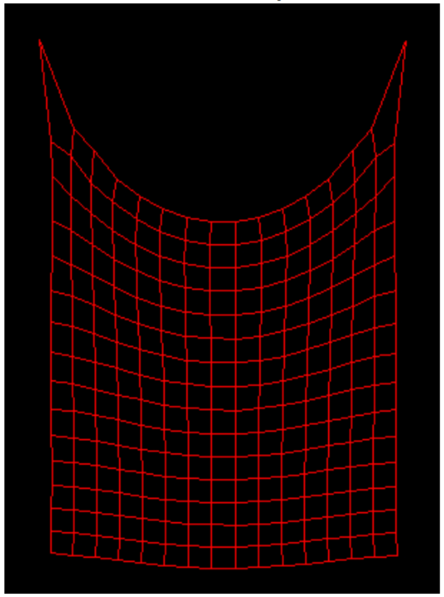
High stiffness can lead to instability (Provot uses explicit integration):

- Must take more shorter time steps to ensure stability
- Leads to more processing time for same length of animation
  - For a stiffness value k_S of a mass μ, and a natural period of oscillation defined as

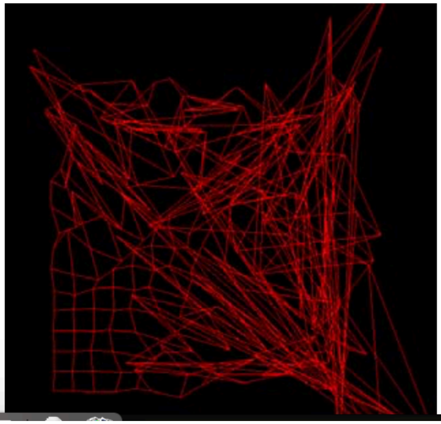
$$T_0 \approx \pi \sqrt{\frac{\mu}{k_S}}$$

the timestep Δt must be less than T_0

Small timestep



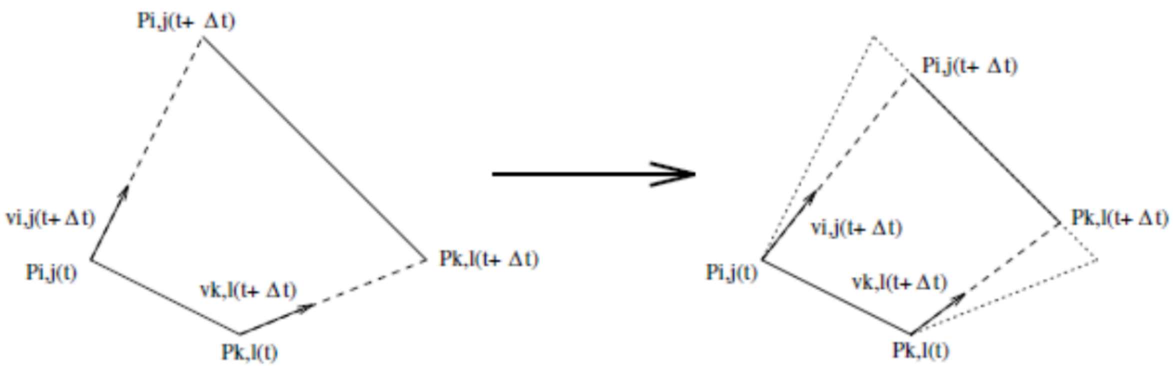
Large timestep



Constraining Deformations

The "Rigid Cloth" Technique [Provot95]

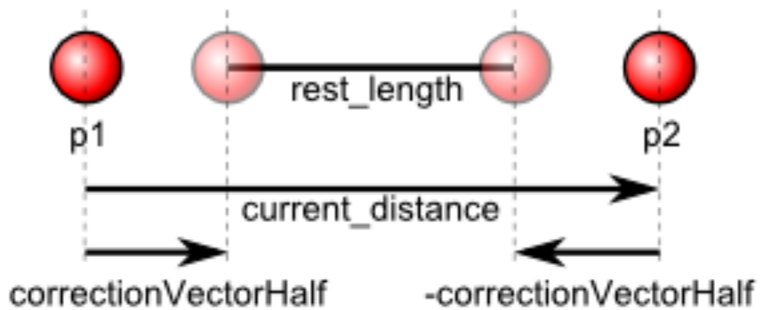
- **Goal**: avoid super elastic effect without excessively decreasing Δt
- **Solution:**
  - Simulate Mass-spring systems as usual
  - Calculate deformation rate of each spring
  - Iff deformation rate > critical deformation rate τ_C
    - Apply "dynamic inverse procedure" to limit deformation to τ_C
    - e.g. If τ_C= 0.1, springs should not ever exceed 110% length



Adjustment of a "super-elongated" spring linking two loose masses [Provot 95].

Constraining Deformations

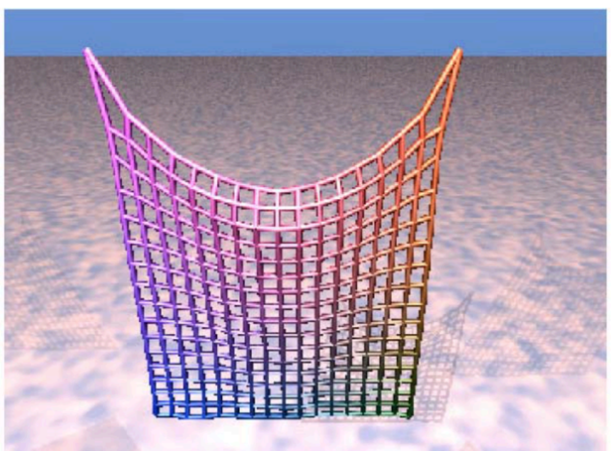
- Dynamic Inverse Procedure



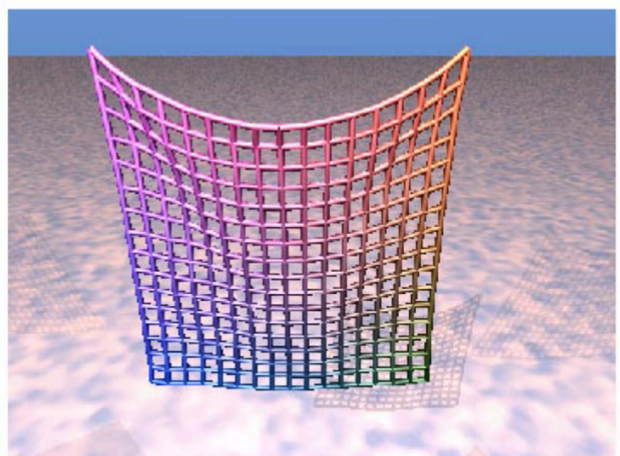
https://viscomp.alexandra.dk/?p=147

Results

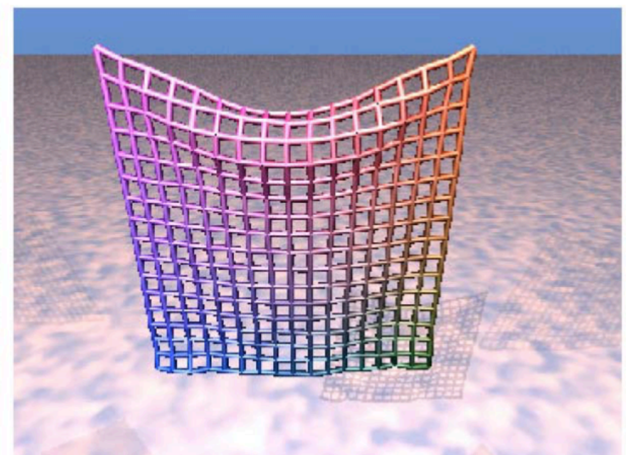
After 200 frames of animation



No constraints



Constraints Applied to structural springs

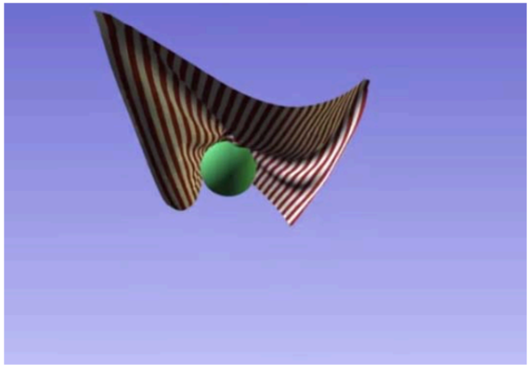


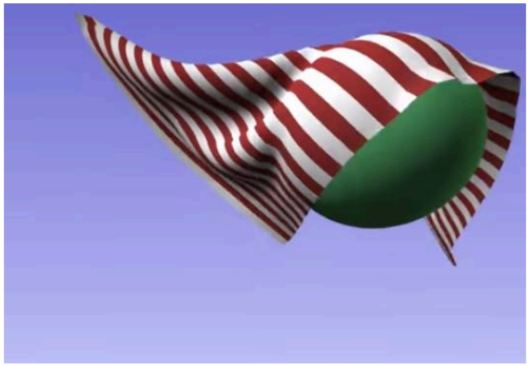
With Constraints applied to structural and shear springs

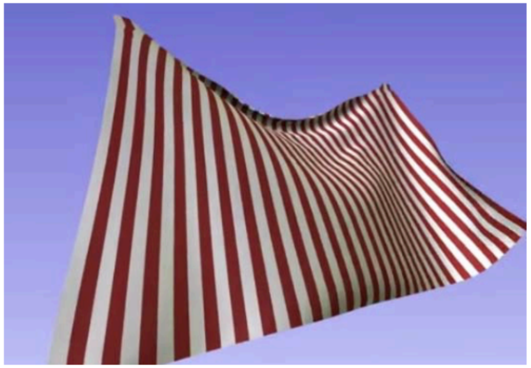
Try it yourself: Cloth Simulation

Jesper Mosegaards's Cloth Simulation Tutorial

- Summarizes most of the points discussed here
- Featuring:
  - Particle system
  - Verlet integration
  - Iterative constraint satisfaction
  - Forces e.g. wind
  - Collisions (with sphere)
- https://viscomp.alexandra.dk/?p=147

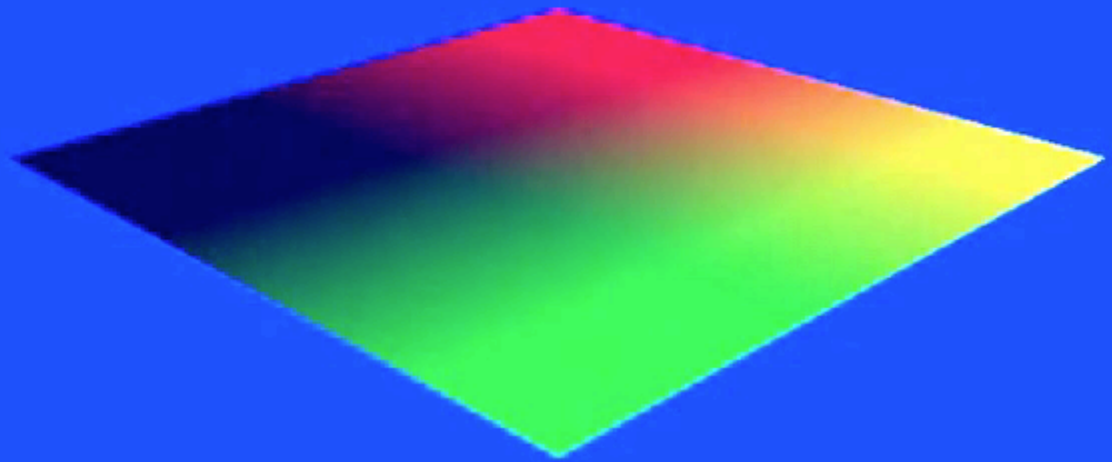






Also: Ben Kenwright "Position-based Dynamics (e.g. Verlet)" Practical Tutorial, Edinburgh Napiers University

- http://games.soc.napier.ac.uk/study/pba_practicals/Practical%2003%20-%20Position%20Based%20Dynamics.pdf



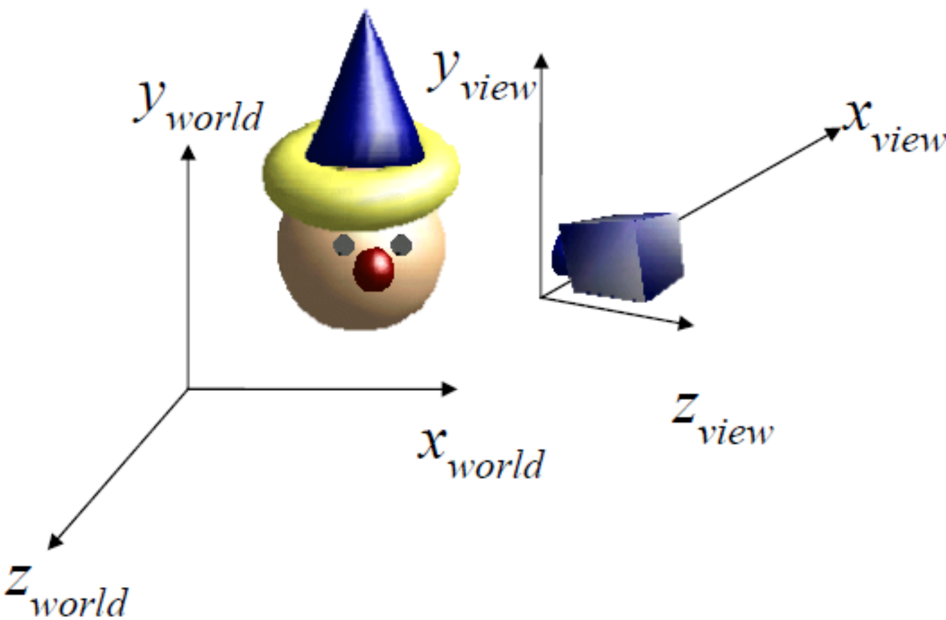
Two-Scale Particle Simulation

Barbara Solenthaler
Markus Gross

ETH Zurich

Next: Viewing

World coordinates



Viewing coordinates:
Viewers (Camera) position and viewing plane.