Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

14: MASS-SPRING SYSTEMS & CLOTH

24/03/2016

HAVOK CLOTH DEMO (GDC 2010)

© Havok https://www.youtube.com/watch?v=wcds8eY93Zk

DEFORMABLE SOLIDS IN GAMES

1D: ropes, hair

2D: cloth, clothing

3D: fat, tires, organs

- In fact all real objects are 3D
- In practice, approximate object with a lower dimensional model if possible ~ for efficiency
- Dimension reduction substantially saves simulation time

Content based on [Mueller'08]

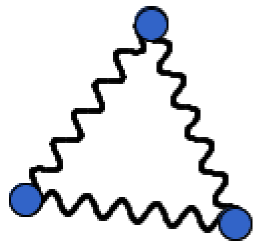
MASS SPRING SYSTEMS

Approach:

- Spatially discretize an object into component **point masses**
- Represent internal forces between mass points using massless elastic **springs**
- Compute positions and velocities at discrete time steps

Pros:

- Easy to implement
- Reasonably fast



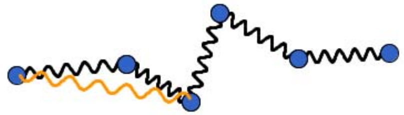
Some problems:

- Lack of preservation of volume
- Behaviour is heavily dependent on mesh topology and choice of springs
- Hard to find springs that correspond with real-world parameter values of specific materials

Content based on [Mueller'08]

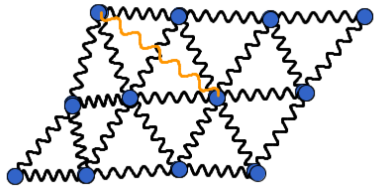
MASS SPRING SYSTEMS

Rope: Chain



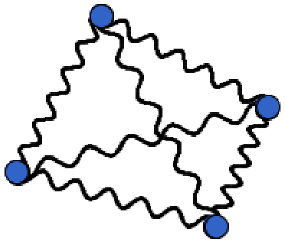
- Additional springs for bending and torsional resistance needed

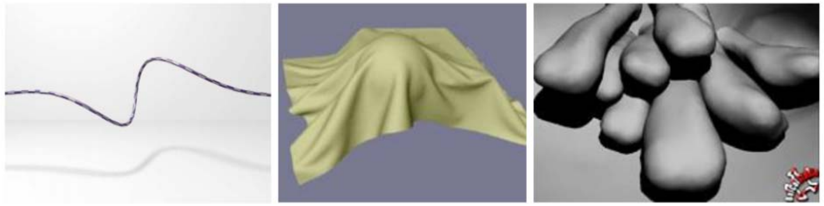
Cloth: triangle mesh



- Additional springs for shearing and bending resistance needed

Soft body: tetrahedral mesh



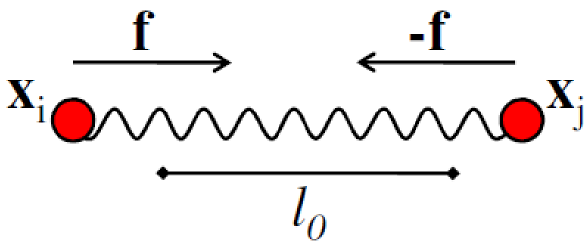


Content based on [Mueller'08] Images © Muller 2008, Top 3 images © Teschner 2003

MASS SPRING PHYSICS

Mass point: mass m, position x, velocity v

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



$$\mathbf{f} = \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|}\left[k_s\left(|\mathbf{x}_j - \mathbf{x}_i| - l_o\right) + k_d(\mathbf{v}_j - \mathbf{v}_i)\cdot\frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|}\right]$$

where: k_s, k_d are the stretching and damping coefficients

Content based on [Mueller'08] Images © Muller 2008

TIME INTEGRATION: EXPLICIT EULER

ODE (Newton):

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

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

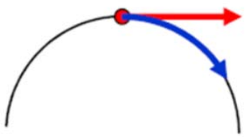
Using Explicit Euler Integration:

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

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

Assumes velocity and force constant within timestep Δt

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



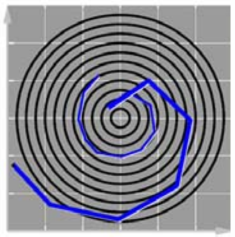
EXPLICIT INTEGRATION ISSUES

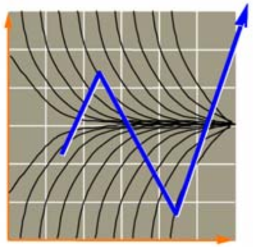
Accuracy

- 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





Content based on [Mueller'08] Images © Baraff et al

IMPLICIT INTEGRATION

Use values of next time-step on the right hand side of the equation

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

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

Reasoning:

- Don’t extrapolate blindly
- Arrive at a physically-based configuration

Content based on [Mueller’08]

IMPLICIT INTEGRATION

Need to solve: $\Delta\boldsymbol{v} = h\boldsymbol{M}^{-1}\left(\boldsymbol{f}_0 + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} h(\boldsymbol{v}_0 + \Delta\boldsymbol{v}) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}}\Delta\boldsymbol{v}\right)$

where $\boldsymbol{M}$ is an $n \times n$ mass matrix with diagonals representing masses,

h is the time step (a.k.a. Δt)

- rewrite as $\left(\boldsymbol{M} - h\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}} - h^2\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right)\Delta\boldsymbol{v} = h\left(\boldsymbol{f}_0 + h\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\boldsymbol{v}_0\right)$
- use Conjugate Gradient Method to solve a linear equation of the form

$$\boldsymbol{r} = \boldsymbol{A}\Delta\boldsymbol{v} - \boldsymbol{b}$$

where $A = \left(\boldsymbol{M} - h\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}} - h^2\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right)$

and $b = h\left(\boldsymbol{f}_0 + h\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\boldsymbol{v}_0\right)$

From Baraff and Witkin "Large Steps in Cloth Simulation" Siggraph 1998

A detailed Practical Explanationis provided in "Implementing the implicit Euler method for mass-spring systems" by Mikko Kauppila. available at: http://hugi.scene.org/online/hugi28/

IMPLICIT INTEGRATION ISSUES

Unconditionally stable (for any Δt)

Have to solve a system of equations for velocities

- n mass point, $3n$ unknowns
- Non-linear when the forces are non-linear in the positions (as with springs)
- Linearize force at each time step

Slow → need large time steps

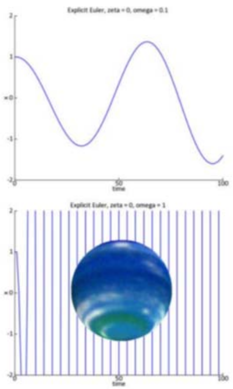
- Some temporal details disappear, numerical damping

See Implicit vs. Explicit Integration (John Foster) http://engineering.utsa.edu/~foster/me4603/files/integration.pdf

COMPARISON OF INTEGRATORS FOR A SPRING

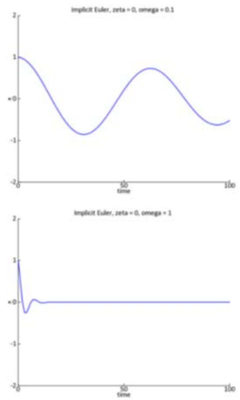
Explicit Euler:

- Fast but unstable
- $\boldsymbol{x}_2 = \boldsymbol{x}_1 + \boldsymbol{v}_1 \Delta t$
- $\boldsymbol{v}_2 = \boldsymbol{v}_1 + \omega^2 \boldsymbol{x}_1 \Delta t$



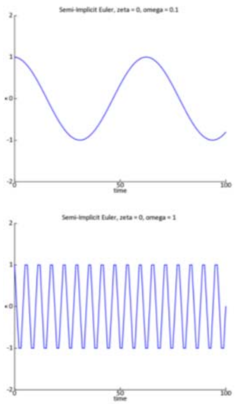
Implicit Euler:

- Slow but unconditionally stable
- $\boldsymbol{x}_2 = \boldsymbol{x}_1 + \boldsymbol{v}_2 \Delta t$
- $\boldsymbol{v}_2 = \boldsymbol{v}_1 + \omega^2 \boldsymbol{x}_2 \Delta t$



Semi-Implicit Euler

- Fast and “stable enough”
- $\boldsymbol{x}_2 = \boldsymbol{x}_1 + \boldsymbol{v}_2 \Delta t$
- $\boldsymbol{v}_2 = \boldsymbol{v}_1 + \omega^2 \boldsymbol{x}_1 \Delta t$



N.B. $\omega = \frac{k_S}{m}$ For any spring constant k_S and mass m

Semi-implicit Euler is used in Bullet, Box2D and Open Dynamics Engine (ODE)

Catto, E. “Soft Constraints: Reinventing the Spring” GDC 2011 Talk. 2011. https://box2.googlecode.com

VERLET INTEGRATION

Originated in molecular dynamics (central difference approximation: find x based on previous frame and next frame)

$$\boldsymbol{x}(t+\Delta t) = 2\boldsymbol{x}(t) - \boldsymbol{x}(t-\Delta t) + \boldsymbol{a}(t)\Delta t^2$$

Position

Acceleration

Change this to something like 1.99 to induce drag

- Velocity-free formulation:
  - velocity implicitly represented by current and previous positions
  - Need to store current and previous $\boldsymbol{x}$
- Not always accurate but fast and stable
- Everything depends on position – so good for constraints (e.g. Rag dolls, cloth)

```
// Verlet integration step

void ParticleSystem::Verlet() {
   for(int i=0; i<NUM_PARTICLES; i++) {
      Vector3& x = m_x[i];
      Vector3 temp = x;
      Vector3& oldx = m_oldx[i];
      Vector3& a = m_a[i];
      x += x-oldx+a*fTimeStep*fTimeStep;
      oldx = temp;
   }
}
```

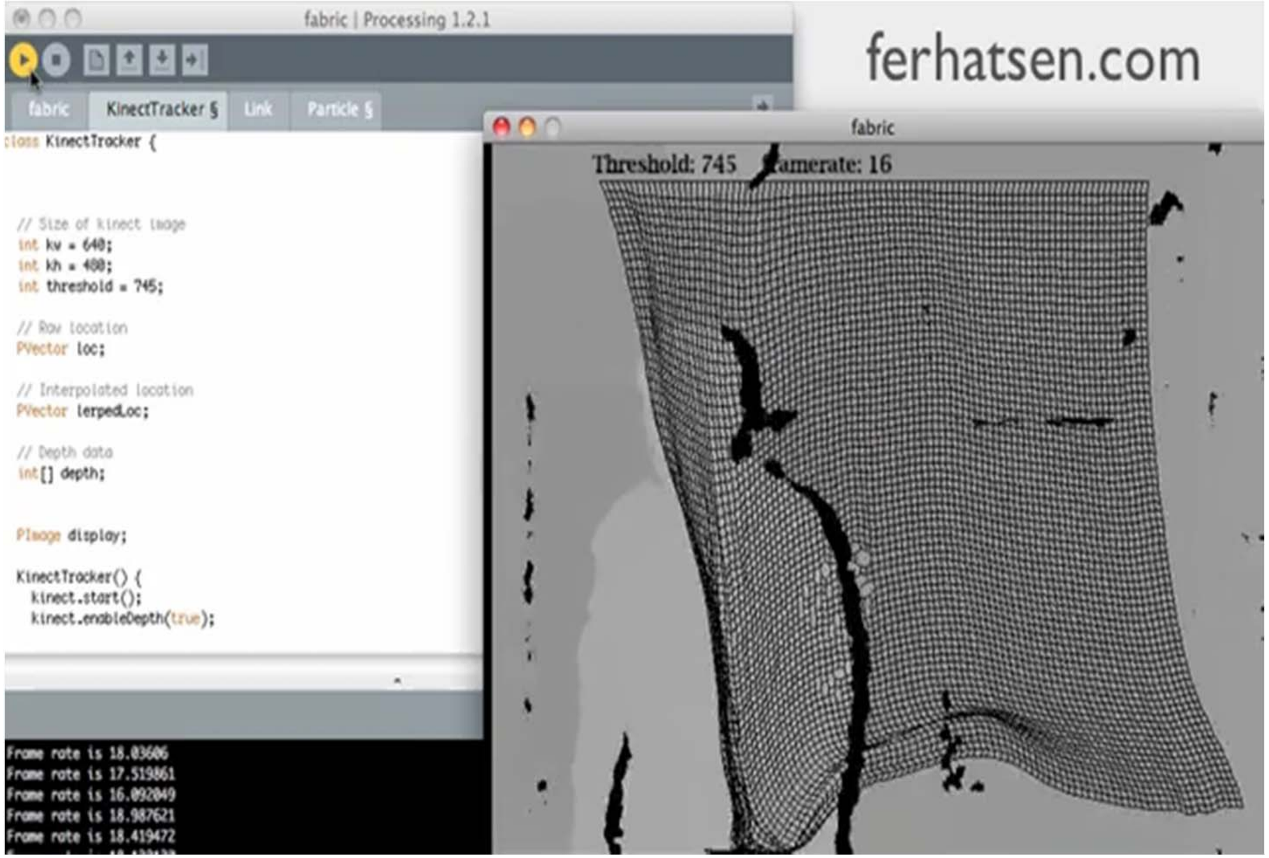
Jakobsen GDC 2001 talk: Advanced Character Physics http://www.floatingorigin.com/mirror/jacobson_01.shtml

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

MASS SPRING SYSTEM FOR CLOTH

See Curtain Demo

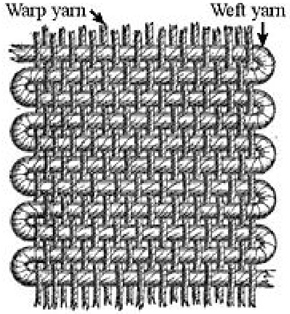
OPEN THE CURTAIN



Kinect + Cloth video by Ferhat Sen http://vimeo.com/20964926
Based on Curtain demo by Jared Counts. http://www.openprocessing.org/sketch/20140

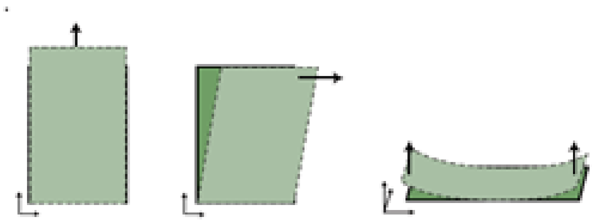
CLOTH

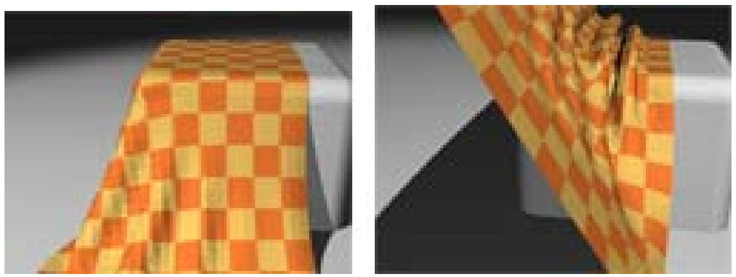
Flexible material consisting of a network of natural or artificial fibres



Basic behaviour:

- Stretch/Compression: Displacement along warp or weft direction.
  - Can't compress at all.
  - Stretched to a limit of ~ 10 percent.
- Shear: Displacement along diagonal directions.
- Bend: Curvature of cloth surface.
  - Easy to bend.
- Drape
- Wrinkle/Buckling





Yalcin & Yaldiz, Techniques for animating cloth, Lecture Notes. Bilkent University. 2009.

PROPERTIES OF CLOTH

Hard to simulate because it has,

- Many primitives and/or nodes within the model
- High degree of freedom at those nodes
- Not perfectly elastic, has stiffness against stretch
- Variety of properties.

Collision detection response is also hard: thin object, self collisions

Must decide between Simple Model vs. Realism.

Often modelled as elastically deformable solids

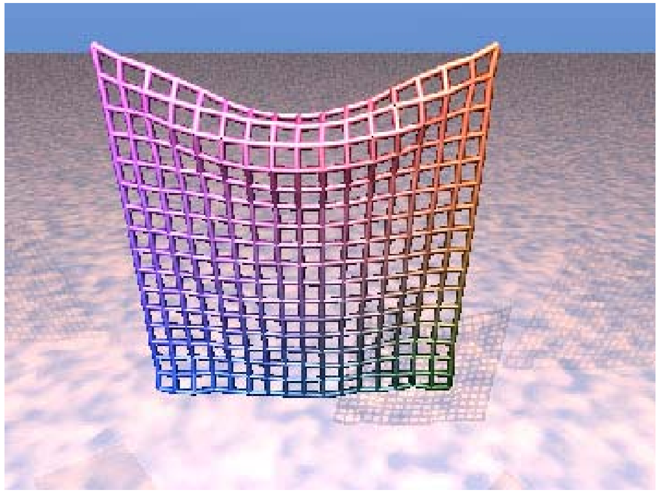
- However cloth behaviour is different from normal elastic models: Non linear response to strain

Yalcin & Yaldiz, Techniques for animating cloth, Lecture Notes. Bilkent University. 2009.

MASS-SPRING CLOTH SIMULATION

Provot [Provot95]: a lattice of springs chosen in a grid structure to model deformable dynamic properties.

- Springs are good for linear strain representation
- Not quite enough to capture 2d/3d elastic behaviour e.g. Shear & Bulk Modulus
  - Need to account for this some other way!

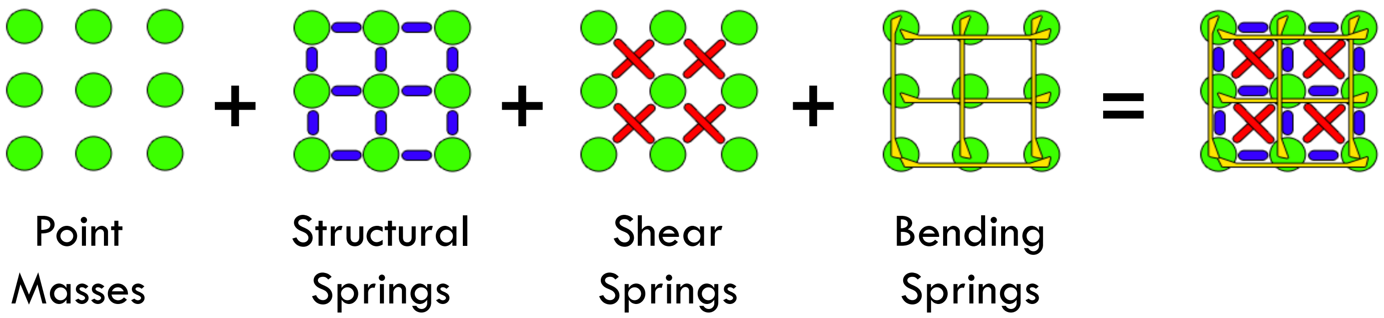


[Provot95] Provot. X, "Deformation constraints in a mass-spring model to describe rigid cloth behaviour" Graphics Interface 1995.

MASS-SPRING SYSTEM FOR CLOTH

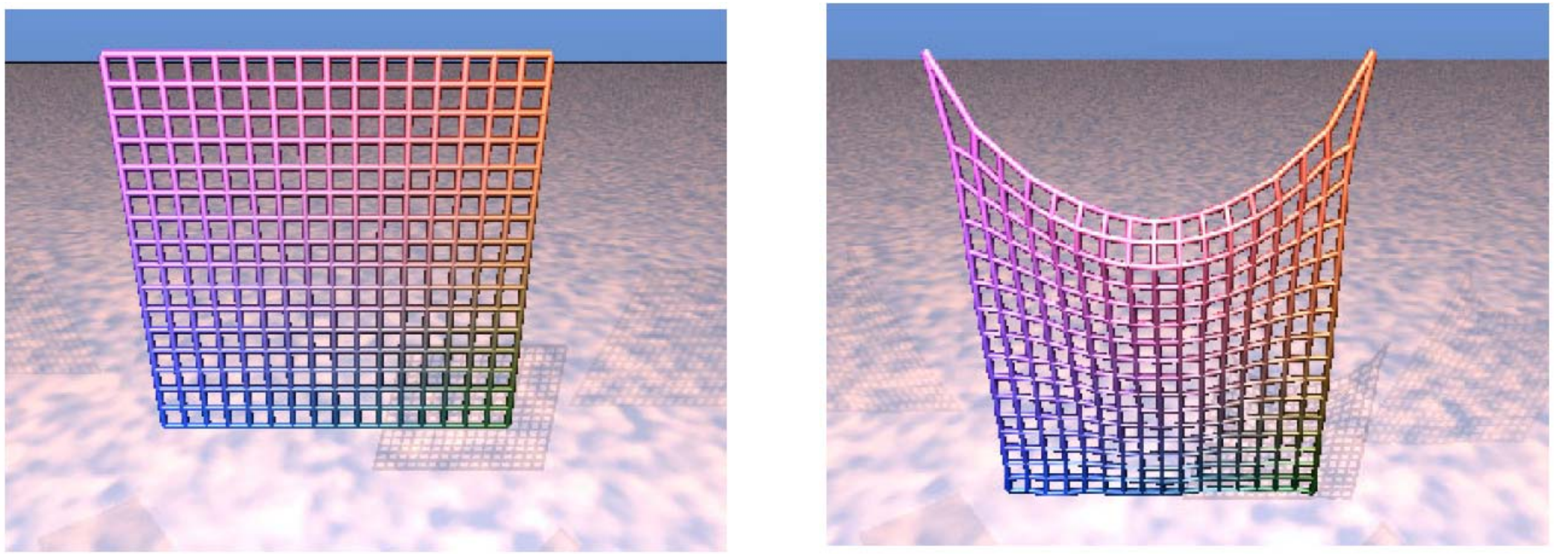
Point masses e.g. Particle system:

- Specify distance constraint using flexible spring model:
  - **Structural** springs : adjacent springs, resist stretching/compression. Typically high k_s
  - **Shear** springs : diagonals on grid, resist shearing
  - **Bend** (flexion) springs : between every 2nd or 3rd particle, Resist bending



Images © Yalcin and Yildiz

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

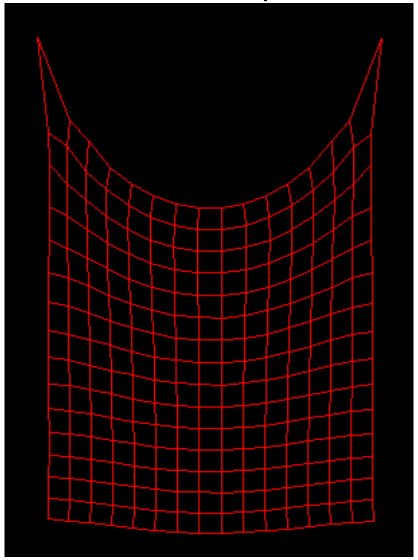
High stiffness can lead to instability (Provot uses explicit integration):

- 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

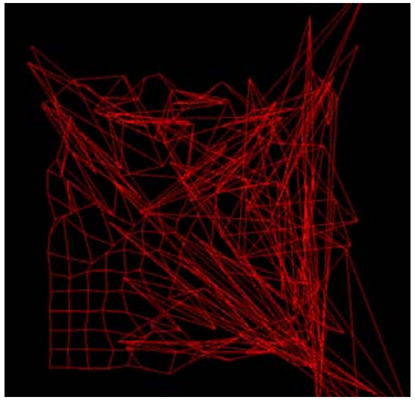
$$T_0 \approx \pi \sqrt{\frac{\mu}{k_S}}$$

the timestep Δt must be less than T_0

Small timestep



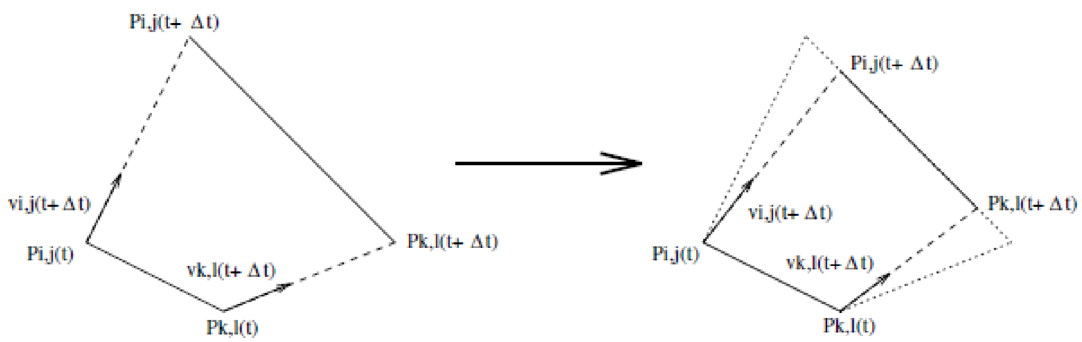
Large timestep



CONSTRAINING DEFORMATIONS

The "Rigid Cloth" Technique [Provot95]

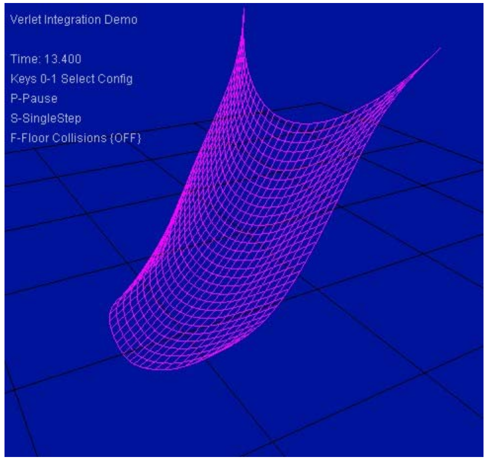
- **Goal**: avoid super elastic effect without excessively decreasing Δt
- **Solution:**
  - Simulate Mass-spring systems as usual
  - Calculate deformation rate of each spring
  - Iff deformation rate > critical deformation rate τ_C
    - Apply "dynamic inverse procedure" to limit deformation to τ_C
    - 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].

PRACTICAL EXAMPLE

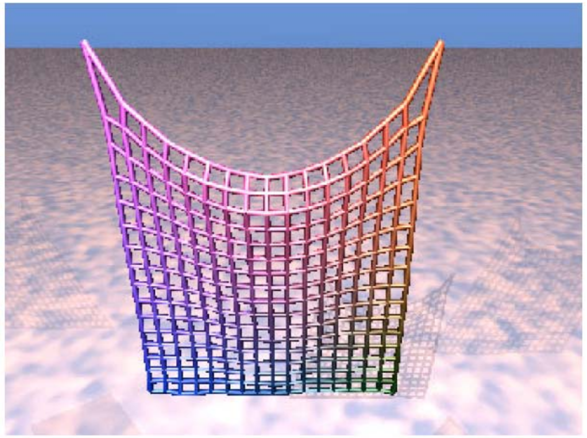
```
void SatisfyConstraints()
{
            const int numIterations = 10;
             for (int i=0; i<numIterations; i++)
            {
                  for (int k=0; k< m_numConstraints; k++)
                  {
                        // Constraint 1 (Floor)
                        if (g_floorCollisions)
                              for (int v=0; v<m_numPoints; v++)
                              {
                                if (m_points[v].curPos.y < 0.0f) m_points[v].curPos.y = 0.0f;
                              }
                         // Constraint 2 (Cloth)
                        Constraint* c = &m_constraints[k];
                        D3DXVECTOR3& p0 = m_points[c->index0].curPos;
                        D3DXVECTOR3& p1 = m_points[c->index1].curPos;
                        D3DXVECTOR3 delta = p1-p0;
                        float len = D3DXVec3Length(&delta);
                        float diff = (len - c->restLength) / len;
                        p0 += delta*0.5f*diff;
                        p1 -= delta*0.5f*diff;
                  }
                   // Keep these two points contraints to there original position
                  float gap = g_gap;
                  m_points[0].curPos           = D3DXVECTOR3(0,                g_disOffFloor, 0);
                  m_points[g_width-1].curPos   = D3DXVECTOR3((g_width-1)*gap,  g_disOffFloor, 0);
            }
      }
```



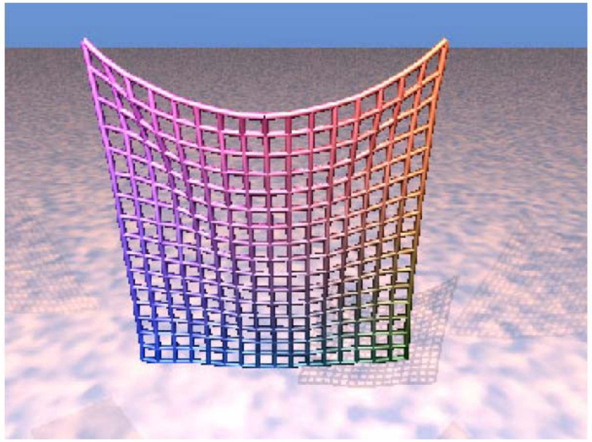
Code snippet © Ben Kenwright. Taken from: http://www.xbdev.net/physics/Verlet/index.php

RESULTS (1)

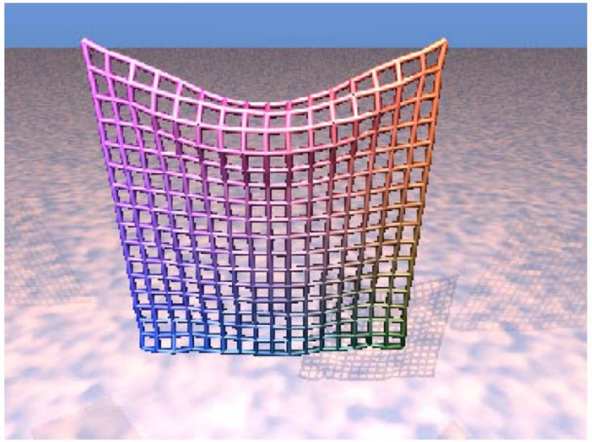
After 200 frames of animation



No constraints



Constraints Applied to structural springs

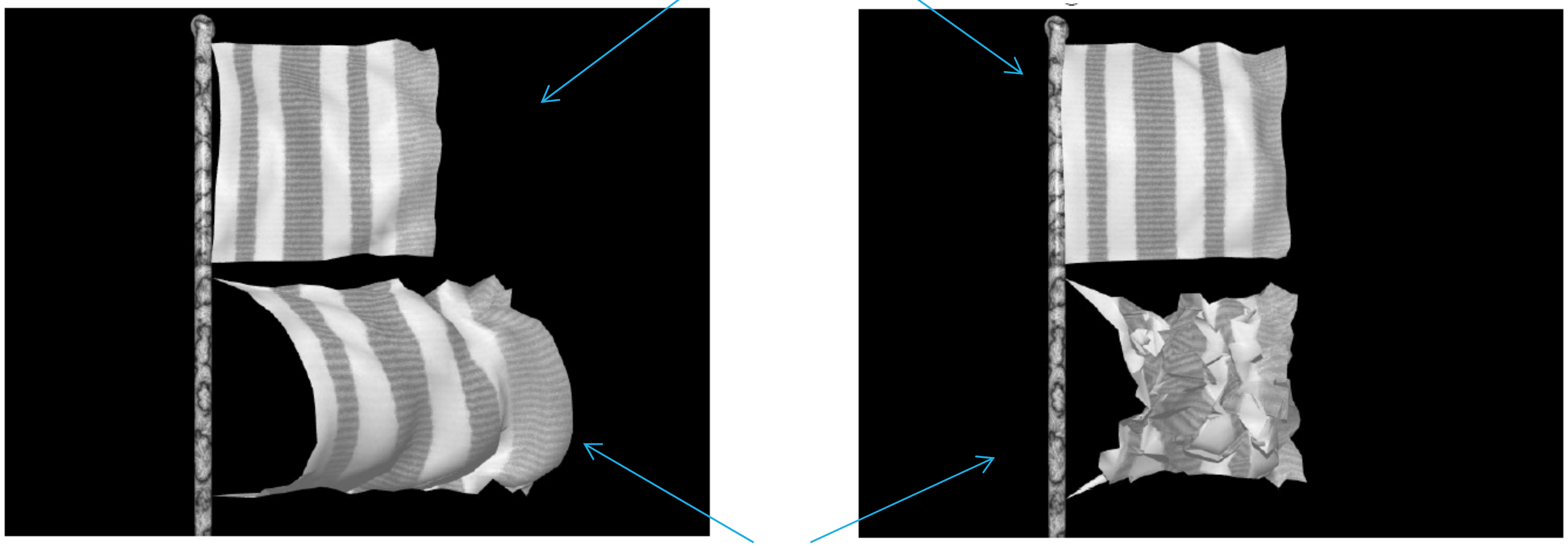


With Constraints applied to structural and shear springs

Images © Provot 1995

RESULTS (2)

“Rigid” Technique $\tau_C = 0.05$



Basic Spring-Mass

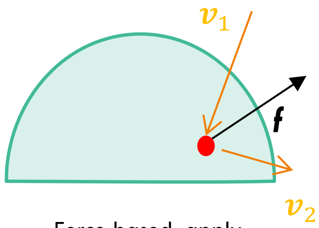
Low Stiffness **High Stiffness**

Images © Provot 1995

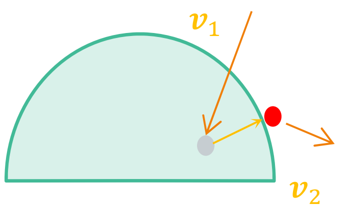
OTHER TECHNIQUES

Position based dynamics

- For cloth and other deformable objects
- Omit velocity layer deal with positions directly: Somewhat related to verlet + constraints discussed earlier
- Manipulate positions to satisfy constraints e.g.
  - if penetration is detected, instead of applying a force or impulse, move objects to enforce non-interpenetration and update to appropriate velocities



Force based: apply collision impulse



Position based: apply constraint and then find correct veloicity

Muller et al Position Based Dynamics. VRIPhys 2006.
http://matthias-mueller-fischer.ch/publications/posBasedDyn.pdf

OTHER TECHNIQUES

Baraff & Witkin: Large time steps in cloth simulation

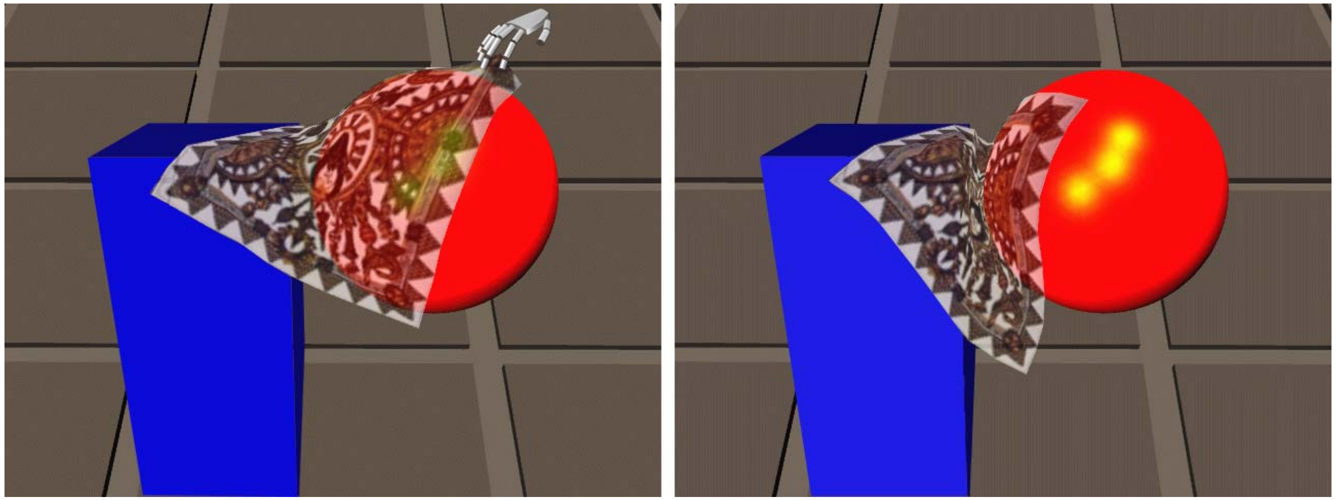
Uniform triangular mesh	**Continuum model:** solve internal energy equations	**Implicit Integration:** System of equations solved by modified Conjugate Gradient method	**Adaptive time step:** leads to graceful degradation

David Baraff and Andrew Witkin, "Large steps in cloth simulation" SIGGRAPH '98

OTHER TECHNIQUES

Cloth as Inverse Kinematics [Desbrun et al '99]. Hybrid approach:

Mass-spring system: Force based simulation	Solve with Inverse Kinematics to ensure stiffness	Implicit Euler integration



Mathieu Desbrun, Peter Schroder, and Alan Barr. "Interactive animation of structured deformable objects" In *Proceedings Graphics interface 1999.*

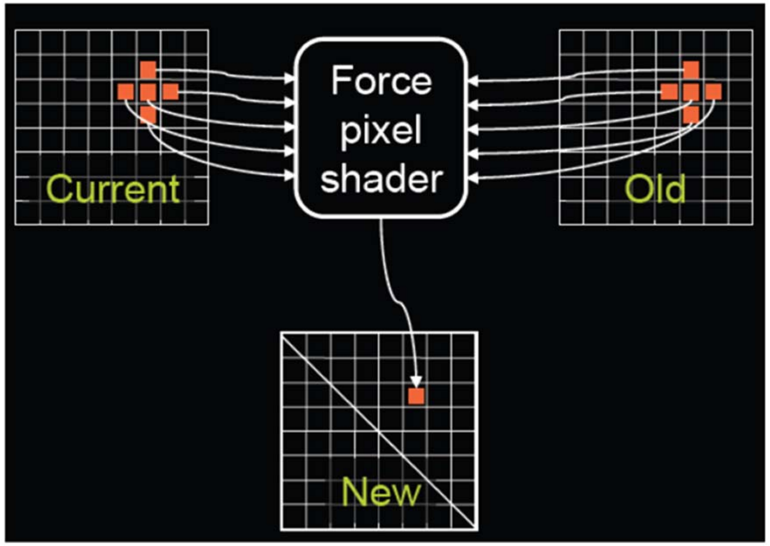
OTHER TECHNIQUES

Cloth on the GPU

- Particle collision and update can be done in shaders (several papers on this)

Overview:

- For every particle, apply external forces
- In each relaxation step, for each cloth particle
  - Evaluate the spring constraints and forces
  - For every intersectable scene geometry, check for collisions and solve collisions by moving the particle out of collided volume.



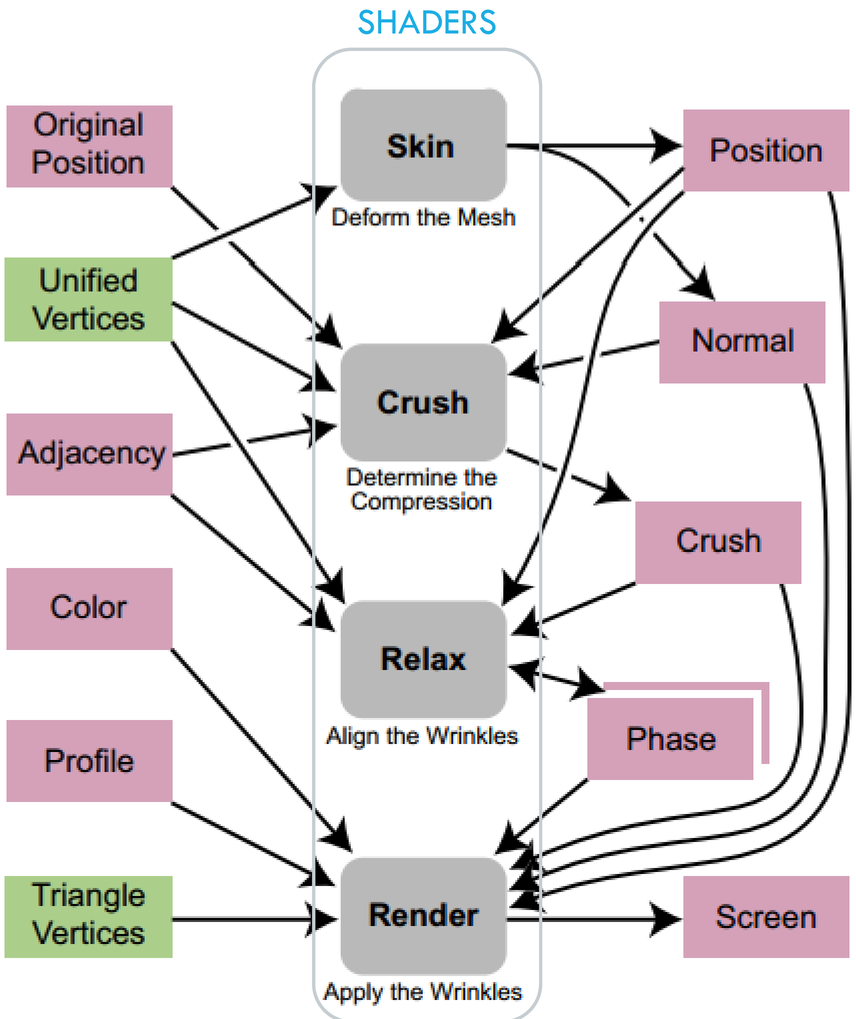
• Cloth is stored as a texture where each pixel is the position of a cloth vertex.
• 3 rotating textures are used to hold the previous frames, for Verlet integration
• To update cloth, draw a full-screen quad into one of the position textures and do the work in the pixel shader.
• To draw cloth, draw a vertex buffer where each vertex's texture coordinates lookup into the position texture, do lookup in the vertex shader

Cyril Zeller. Cloth Simulation - White Paper, nVidia. http://portal.acm.org/citation.cfm?id=1187158
http://developer.download.nvidia.com/whitepapers/2007/SDK10/Cloth.pdf

OTHER TECHNIQUES

Geometric solution on GPU

- Assumes tightly stretched over surface ~ apply geometric techniques to fake effects such as wrinkling



Jörn Loviscach. Wrinkling coarse meshes on the GPU. In Proceedings of Eurographics 2006.

MANY OTHER WORKS: SOME SURVEYS BELOW

N. Magnenat-Thalmann, F.Corder, M.Keckeisen, S.Kimmerle "Simulation of Clothes for Real-time Applications" Eurographics 2004 Tutorials

- http://cg.cs.uni-bonn.de/en/publications/paper-details/eg-tutorial-2004/

J. Long, K. Burns, J. Yang. "Cloth Modelling and Simulation – A literature Survey" in Proceedings of the Third international conference on Digital human modelling. 2011

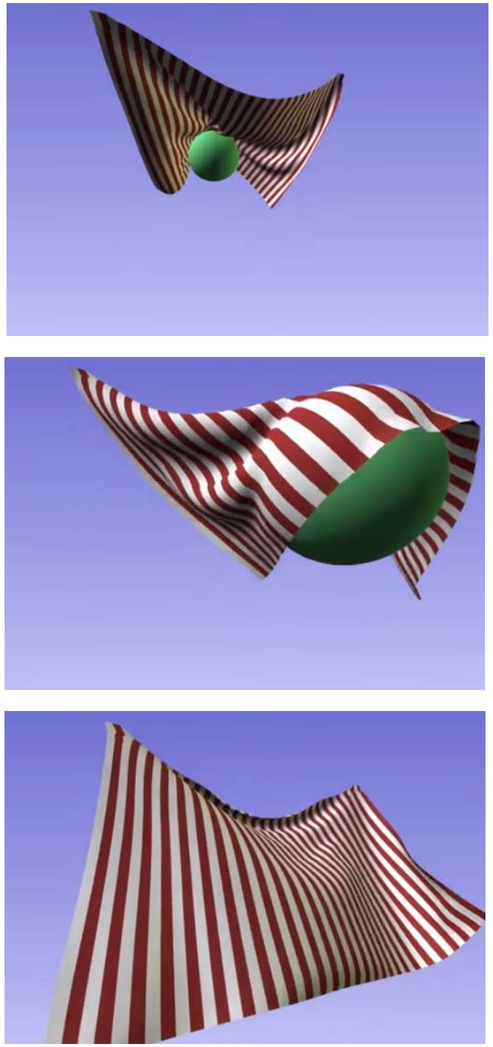
- http://link.springer.com/chapter/10.1007%2F978-3-642-21799-9_35

TRY IT YOURSELF: CLOTH SIMULATION

[N.B. Optional. No marks; unless this is chosen as a part of Assignment 6.
All external resources used should be cited]

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)
- http://cg.alexandra.dk/?p=147



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

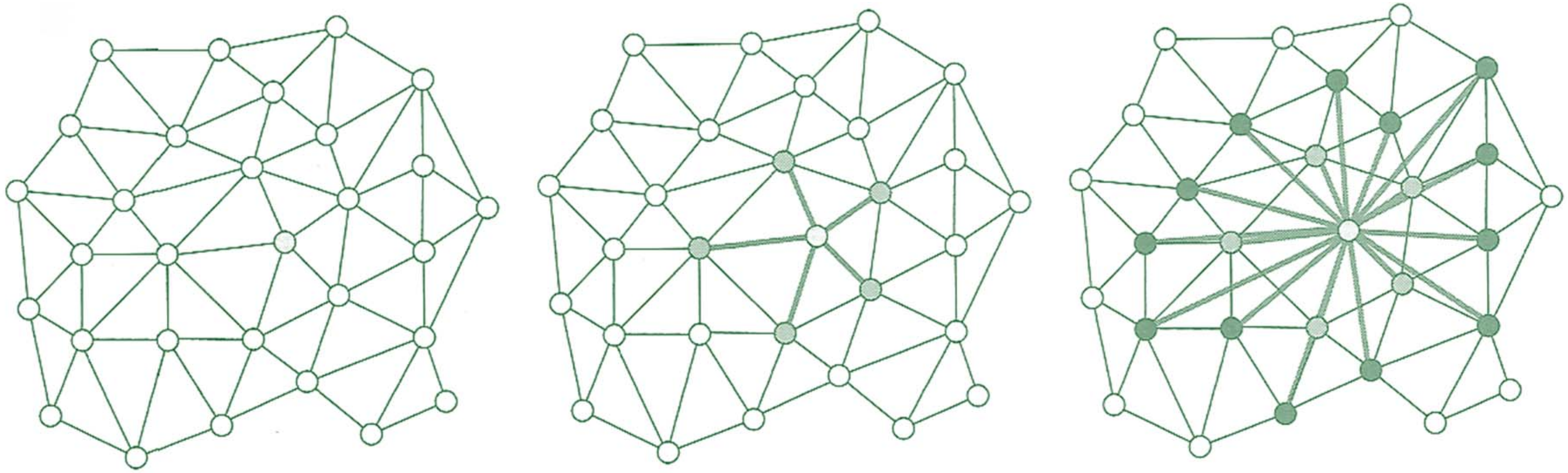
- http://games.soc.napier.ac.uk/study/pba_practicals/Practical%2003%20-%20Position%20Based%20Dynamics.pdf

OTHER MASS SPRING MODELS

NON-GRID STRUCTURES

For an unstructured mesh:

- Structural springs correspond to 1-neighbourhood
- Shearing and bending to a 2-neighbourhood

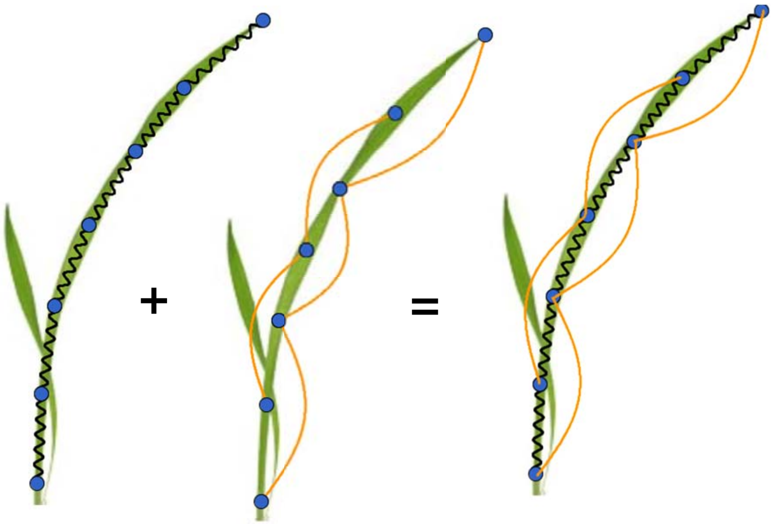


1 - neighbourhood

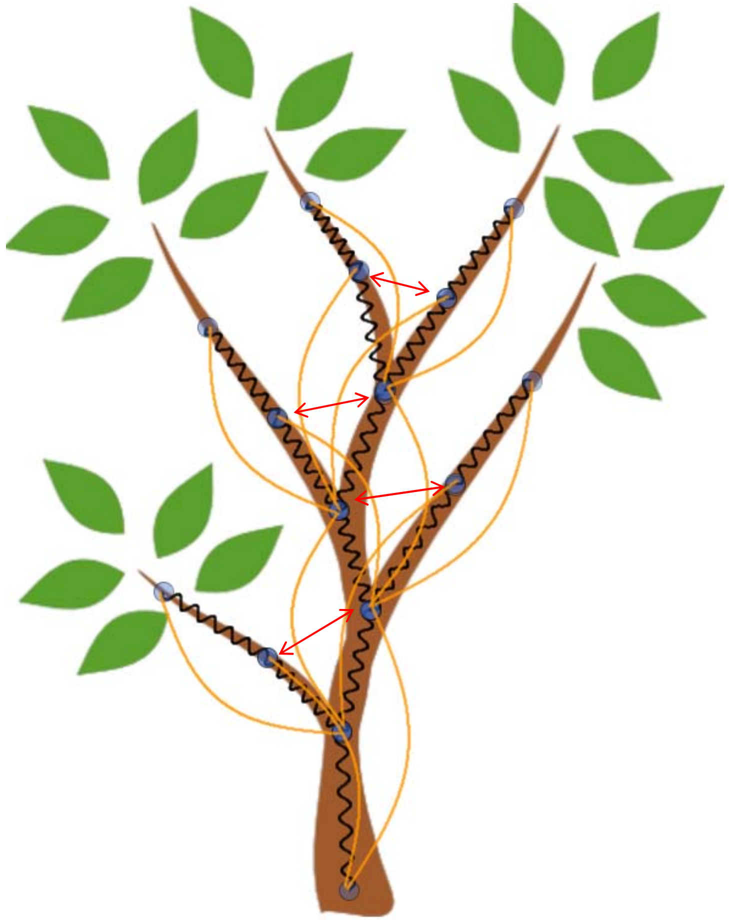
2 - neighbourhood

Images © Erleben et al

EXAMPLE: PLANTS AND TREES



Structural Springs | Bending Springs | Grass Straw

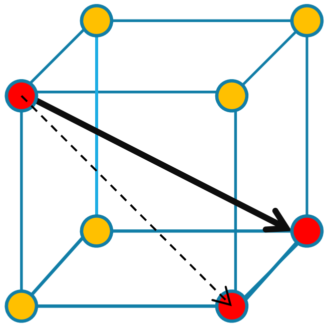


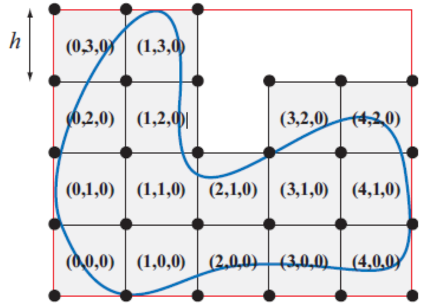
Spring model dynamic proxy for a tree

3D SOLIDS

Discretise object into 3d rectilinear grid (voxels)

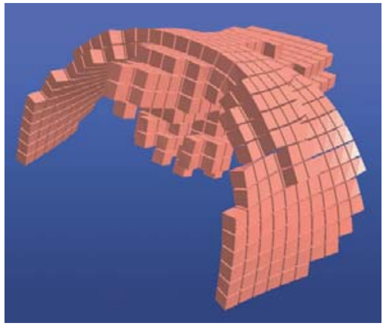
- In addition to shearing diagonals we need spatial diagonals that counter volume loss (approximate the bulk modulus)

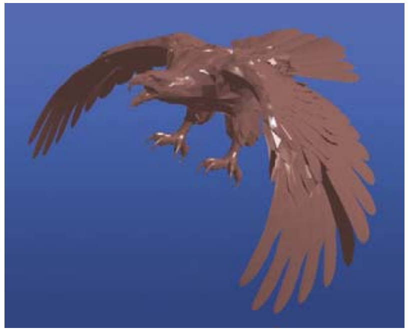




Unstructured solid meshes can be voxelised

- Mesh points in between grid nodes are updated by trilinear interpolation





Muller, M., Teschner, M., and Gross, M. **Physically-Based Simulation of Objects Represented by Surface Meshes.** In *Computer Graphics International* 2004.

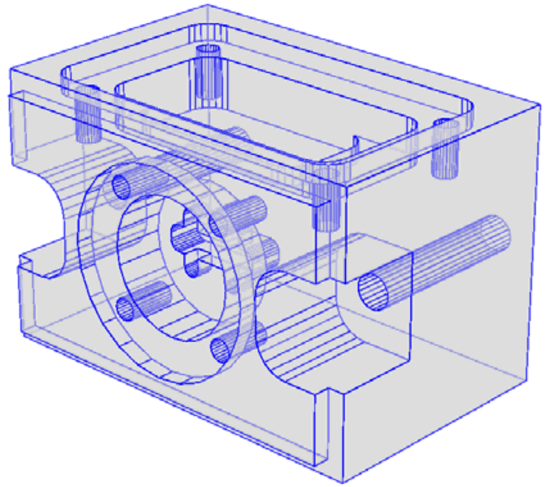
TETRAHEDRAL DECOMPOSITION

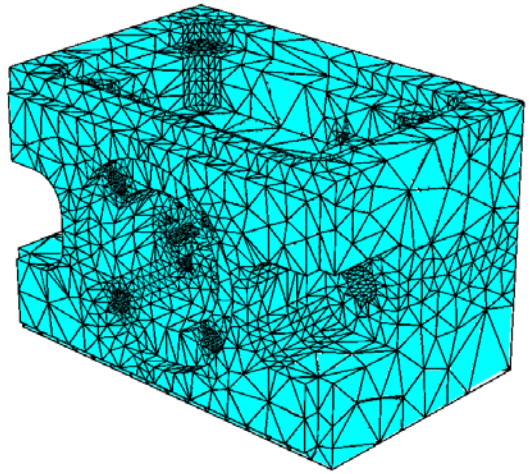
Split mesh up into tetrahedra

Note that interior details are important (not just surface)

Delauney tetrahedralization

Springs between vertices and 2+ neighbourhoods depending on rigidity

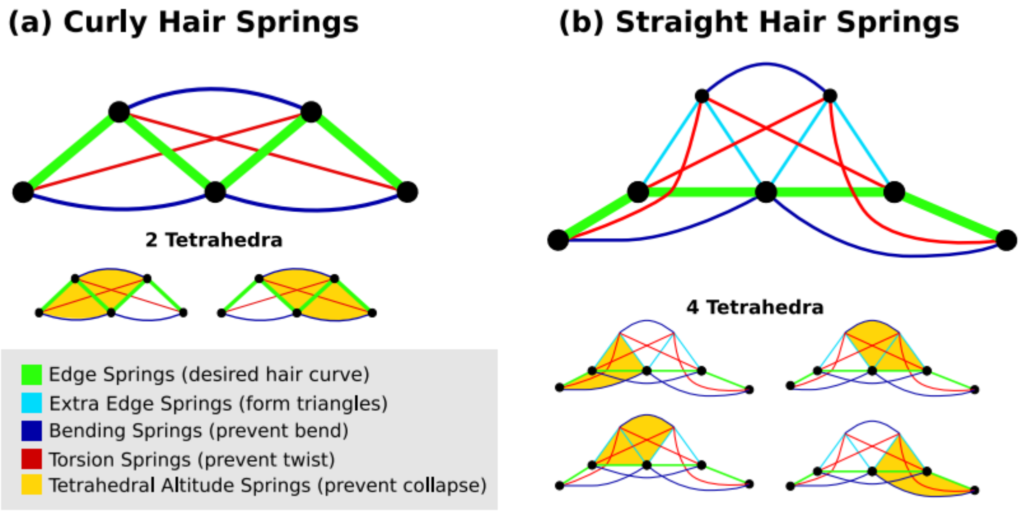


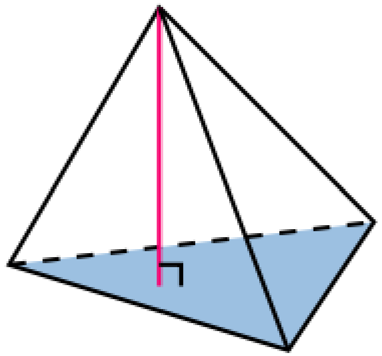


Various tetrahedralization solutions exist

- e.g. tetgen: http://tetgen.berlios.de/

HAIR SIMULATION





A. Selle, M. Lentine, R. Fedkiw. "A Mass-spring Model for Hair Simulation" Siggraph 2008

MASS SPRING SYSTEMS

Advantages:

- Simplicity – e.g. compared to FEM
- Nice extension of particle systems

Disadvantages:

- Volume preservation + Degenerate cases e.g. inverted tetrahedra
  - some solutions but can be expensive
- Not an accurate representation of complex solids: fine details incorrect
  - Solved with supplementary springs e.g. bending, torsional springs
  - But finding parameter values that fit real-world behaviour is difficult
  - Design of the mass-spring network can be impractical for complex structures

REQUIRED READING*

*questions in Assignment 7 may be based on Required Readings

Seminal paper on easy spring-mass cloth simulation (N.B. not the first cloth paper):

- Xavier Provot: “Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behaviour”
  - http://graphics.stanford.edu/courses/cs468-02-winter/Papers/Rigidcloth.pdf

Matthias Mueller – Realtime Physics. Siggraph 2008 Course Notes

- Chapter 3 Mass-spring Systems: Section 3.1 – 3.6
- http://www.matthiasmueller.info/realtimephysics/

OTHER REFERENCES

D. Baraff. "Implicit Methods for Differential Equations" – Siggraph 2001 Course Notes on Physically Based Modelling. Lecture D.

- http://www.pixar.com/companyinfo/research/pbm2001/pdf/notesd.pdf

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

John. T. Foster. "Brief Explanation of Integration Schemes",

- http://engineering.utsa.edu/~foster/me4603/files/integration.pdf

M. Adil Yalçın, Cansın Yıldız "Techniques for animating Cloth" unpublished survey, Bilkent University. 2009.

  - http://www.cs.bilkent.edu.tr/~cansin/projects/cs567-animation/cloth/cloth-paper.pdf

M. Kauppila. "Implementing the implicit Euler method for mass-spring systems"

- available at: http://hugi.scene.org/online/hugi28/

Michael Hauth: "Numerical Techniques for Cloth Simulation" Siggraph 2003 Course Notes (The math required for cloth)

  - http://www.gris.uni-tuebingen.de/people/staff/mhauth/tutorials/Vis03/SIG2003Tut29INumerics.pdf

ASSIGNMENT 6 MEETINGS NEXT WEEK

Monday 16/03/2015	
14:00	Xinwei Xiong
14:15	Brendan O'Connor
14:30	Hao Guan
14:45	Tony Cullen
15:00	Yafei Qu
15:15	Fan Li
15:30	Saloni Sharma
15:45	Sarah Noonan

Thursday 19/03/2015	
10:00	Jeremiah Dunne
10:15	Giovanni Campo
10:30	Huangxiang Wang
10:45	Patrick O'Halloran

TO DO

Check Baraff Notes PBM for integration

Check "Implicit Euler Method for Mass-springs" and see if we can make a slide on this http://hugi.scene.org/online/hugi28/hugi%2028%20-%20coding%20corner%20uttumuttu%20implementing%20the%20implicit%20euler%20method%20for%20mass-spring%20systems.htm

- Some equations in: http://web.cse.ohio-state.edu/~whmin/courses/cse788-2011-fall/results/KoChih_Wang/finalReport.htm

Abstract Slide 22 – see f there are other examples?

Decide on Required Reading