Simulation in Computer Graphics

Deformable Objects

Matthias Teschner

Computer Science Department
University of Freiburg

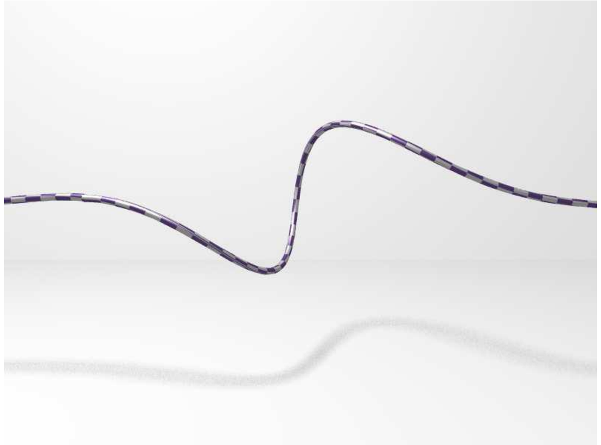
UNI FREIBURG

Outline

- introduction
- forces
- performance
- collision handling
- visualization

Motivation

- sets of particles are used to model time-dependent phenomena such as ropes, cloth, deformable objects
- forces between particles account for resistance to stretch, shear, bend, volume changes ...



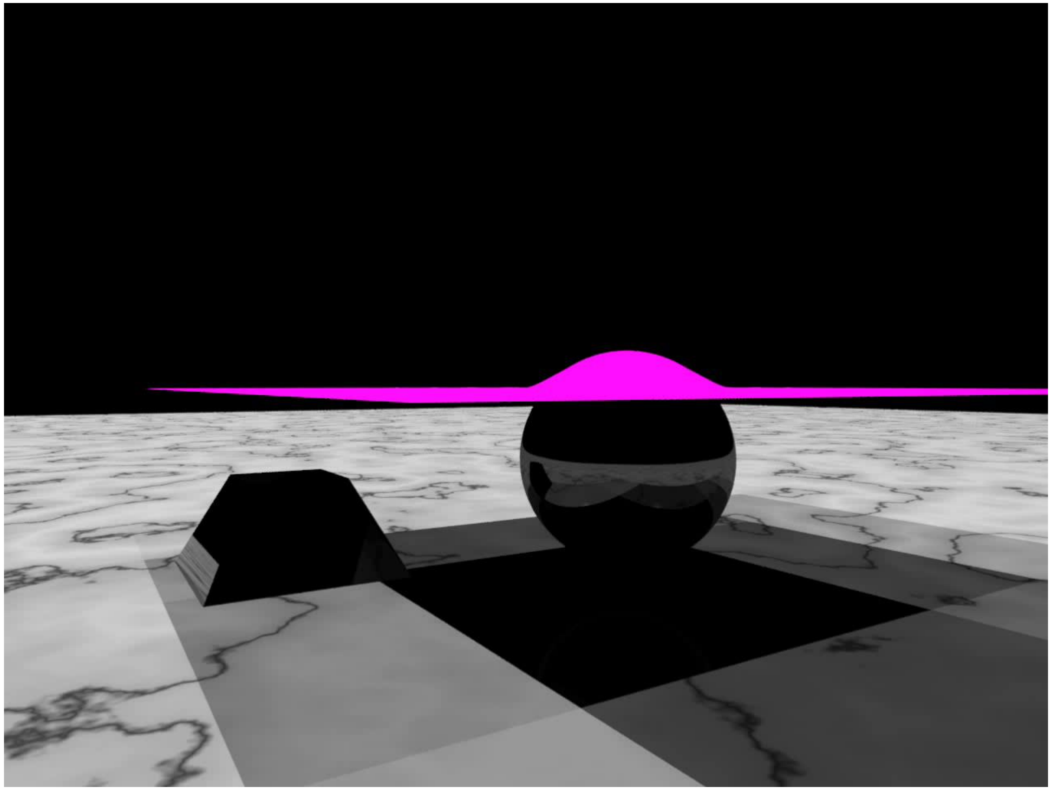
1D, 2D, and 3D mass-point systems, University of Freiburg

Example

- discretization of an object into **mass points**
- representation of **internal forces** between mass points, e. g. spring forces
- computation of the **dynamics**, positions and velocities at discrete time points

Applications

- entertainment technologies
  - cloth
  - facial expressions
- computational medicine
  - medical training
  - pre-operative surgical planning



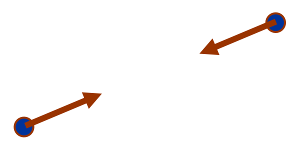
Bridson, Fedkiw, Anderson, Stanford University

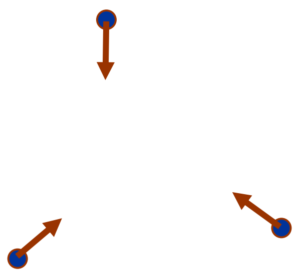
Outline

- introduction
- forces
  - examples
  - energy constraints
  - damping
  - plasticity and other effects
- performance
- collision handling
- visualization

Internal Forces

- internal forces are symmetric with respect to at least two mass points



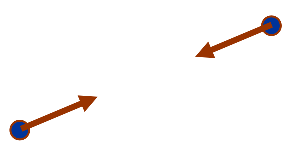


$$\mathbf{F}^{\text{int}}_{ij} + \mathbf{F}^{\text{int}}_{ji} = \mathbf{0} \quad \mathbf{F}^{\text{int}}_{ijk} + \mathbf{F}^{\text{int}}_{jki} + \mathbf{F}^{\text{int}}_{kij} = \mathbf{0}$$

- sum of internal forces is zero

$$\sum_i \sum_j \mathbf{F}^{\text{int}}_{ij} = \mathbf{0}$$

- internal forces cause no torque



internal forces

external forces

Internal Forces

- internal forces are defined for at least two points $\mathbf{F}_{ij}^{\text{int}} + \mathbf{F}_{ji}^{\text{int}} = \mathbf{0} \quad \Rightarrow \quad \mathbf{F}_{ii}^{\text{int}} = \mathbf{0}$
- internal forces do not influence the global dynamic behavior of a mass-point system (linear and angular momentum is preserved)
- spring force is an internal force
- external forces can change linear and angular velocity of a mass-point system
- is gravitational force an internal force?

Gravity

- objects with positions $\mathbf{x}_1, \mathbf{x}_2$ and masses m_1, m_2 attract each other with forces $\mathbf{F}_1, \mathbf{F}_2$

$$\mathbf{F}_1 = -\mathbf{F}_2 = G \frac{m_1 m_2}{|\mathbf{x}_2 - \mathbf{x}_1|^2} \frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}$$

- G is the gravitational constant

$$G \approx 6.67 \cdot 10^{-11} \mathrm{N\ m}^2 \mathrm{kg}^{-2}$$

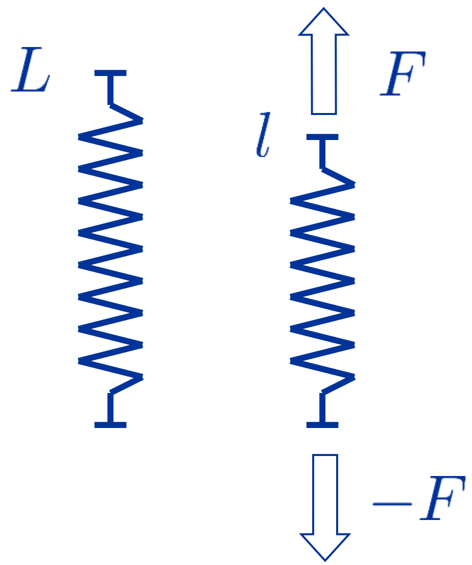
- internal force, if applied to both objects

Gravity

- on earth
  - gravity is dominated by earth
  - mass m_1 of the earth is constant
  - distance from surface to center is nearly constant
  - $\mathbf{g} = G \frac{m_1}{|\mathbf{x}_2 - \mathbf{x}_1|^2} \frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|} = \text{const}$
- $\mathbf{g}$ is an acceleration pointing towards the earth center
- in virtual environments, the direction depends on the coordinate system, e. g. $\mathbf{g} = 9.81 \cdot (0, 0, -1)^{\mathrm{T}} \mathrm{m\ s}^{-2}$
- force exerted to m_2 due to gravity: $\mathbf{F}_2 = m_2 \cdot \mathbf{g}$
- external force, if not applied to both objects

Elastic Spring

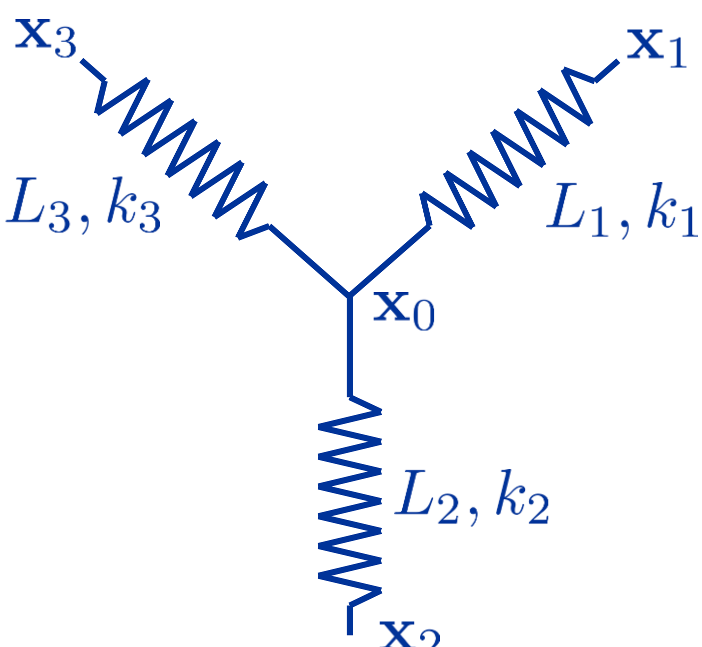
- k - spring stiffness
  L - initial spring length
  l - current spring length



- linear force-deformation relation (Hooke's law)
  $$F = k(L - l)$$
- simple mechanism for internal forces
- elasticity: ability of a spring to return to its initial configuration in the absence of forces

Mass-Spring System in 3D

- non-linear relation of forces and mass-point positions



$$\mathbf{F}_0 = \sum_i k_i(|\mathbf{x}_i - \mathbf{x}_0| - L_i)\frac{\mathbf{x}_i - \mathbf{x}_0}{|\mathbf{x}_i - \mathbf{x}_0|}$$

A Simple Deformable Object

- discretize the object into mass points
- define the connectivity
  (topology, adjacencies of mass points)
- set model parameters
  - point: mass, position, velocity
  - spring: stiffness, initial length
- compute forces: spring force, gravity
- update positions and velocities of all mass points
  with a numerical integration scheme, e. g.

$$\mathbf{v}_i^{t+h} = \mathbf{v}_i^t + h\frac{1}{m_i}\mathbf{F}_i^t \qquad \mathbf{x}_i^{t+h} = \mathbf{x}_i^t + h\mathbf{v}_i^{t+h}$$

Explicit Numerical Integration

- e.g., Heun
  - for all mass points compute $\mathbf{k}_{i,1}, \mathbf{l}_{i,1}$
  - for all mass points compute $\mathbf{k}_{i,2}, \mathbf{l}_{i,2}$
  - for all mass points compute $\mathbf{x}_i^{t+h}, \mathbf{v}_i^{t+h}$
- $\mathbf{l}_{i,1}$ depends on position $\mathbf{x}_i^t$ and all connected neighbors $\mathbf{x}_j^t$,
- $\mathbf{l}_{i,2}$ depends on the predicted position $\mathbf{x}_i^t + \mathbf{k}_{i,1}h$ and on the predicted positions of all neighbors $\mathbf{x}_j^t + \mathbf{k}_{j,1}h$

$$
\begin{aligned}
\mathbf{k}_{i,1} &= \dot{\mathbf{x}}^t \\
\mathbf{l}_{i,1} &= \dot{\mathbf{v}}_{\mathbf{x}_i^t,\ldots\mathbf{x}_j^t} \\
\mathbf{k}_{i,2} &= \dot{\mathbf{x}}_i^t + \mathbf{l}_{i,1}h \\
\mathbf{l}_{i,2} &= \dot{\mathbf{v}}_{\mathbf{x}_i^t+\mathbf{k}_{i,1}h,\ldots,\mathbf{x}_j^t+\mathbf{k}_{j,1}h} \\
\mathbf{x}_i^{t+h} &= \mathbf{x}_i^t + h\left(\frac{1}{2}\mathbf{k}_{i,1} + \frac{1}{2}\mathbf{k}_{i,2}\right) \\
\mathbf{v}_i^{t+h} &= \mathbf{v}_i^t + h\left(\frac{1}{2}\mathbf{l}_{i,1} + \frac{1}{2}\mathbf{l}_{i,2}\right)
\end{aligned}
$$

Implicit Numerical Integration

- e.g., implicit Euler

$$\mathbf{x}_i^{t+h} = \mathbf{x}_i^t + h\mathbf{v}_i^{t+h}$$
$$\mathbf{v}_i^{t+h} = \mathbf{v}_i^t + h\frac{1}{m}\mathbf{F}_i^{t+h}(\mathbf{x}_i^t, \ldots, \mathbf{x}_j^t)$$

- $\mathbf{x}^t = (\mathbf{x}_1^{\mathrm{T}}, \mathbf{x}_2^{\mathrm{T}}, \ldots, \mathbf{x}_n^{\mathrm{T}})^{\mathrm{T}}$

  $\mathbf{v}^t = (\mathbf{v}_1^{\mathrm{T}}, \mathbf{v}_2^{\mathrm{T}}, \ldots, \mathbf{v}_n^{\mathrm{T}})^{\mathrm{T}}$

  $\mathbf{F^t}(\mathbf{x}^t) = (\mathbf{F}_1^{\mathrm{T}}, \mathbf{F}_2^{\mathrm{T}}, \ldots, \mathbf{F}_n^{\mathrm{T}})^{\mathrm{T}}$

  $\mathbf{M} = diag(m_1, m_1, m_1, \ldots, m_n, m_n, m_n) \in \mathbb{R}^{3n\times 3n}$

- $\mathbf{M}\mathbf{v}^{t+h} = \mathbf{M}\mathbf{v}^t + h\mathbf{F}^{t+h}(\mathbf{x}^{t+h})$

  $\mathbf{M}\mathbf{v}^{t+h} = \mathbf{M}\mathbf{v}^t + h\mathbf{F}^{t+h}(\mathbf{x}^t + h\mathbf{v}^{t+h})$

- force linearization

  $\mathbf{M}\mathbf{v}^{t+h} = \mathbf{M}\mathbf{v}^t + h\mathbf{F}^t(\mathbf{x}^t) + h^2\frac{\partial \mathbf{F}^t}{\partial \mathbf{x}^t}\mathbf{v}^{t+h}$ $\quad \frac{\partial \mathbf{F}^t}{\partial \mathbf{x}^t} = \mathbf{J}^t \in \mathbb{R}^{3n\times 3n}$

  $\left(\mathbf{M} - h^2\mathbf{J}^t\right)\mathbf{v}^{t+h} = \mathbf{M}\mathbf{v}^t + h\mathbf{F}^t(\mathbf{x}^t)$

Implicit Numerical Integration

- in the Jacobian $\mathbf{J}^t$, a spring force between $\mathbf{x}_i^t$ and $\mathbf{x}_j^t$ is represented by four sub matrices
  $\mathbf{J}_{i,j}^t \in \mathbb{R}^{3\times 3},\ \mathbf{J}_{j,i}^t \in \mathbb{R}^{3\times 3},\ \mathbf{J}_{i,i}^t \in \mathbb{R}^{3\times 3},\ \mathbf{J}_{j,j}^t \in \mathbb{R}^{3\times 3}$
  that are accumulated at positions
  $(3i, 3j),\ (3j, 3i),\ (3i, 3i),\ (3j, 3j)$

$$\mathbf{J}_{i,i}^t = \frac{\partial \mathbf{F}_i^t}{\partial \mathbf{x}_i^t} \in \mathbb{R}^{3\times 3}$$

$$\mathbf{J}_{i,i}^t = \frac{\partial}{\partial \mathbf{x}_i^t} k_s \left((\mathbf{x}_j - \mathbf{x}_i) - L_s \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|} \right)$$

$$= k_s \left(-\mathbf{I} + \frac{L_s}{|\mathbf{x}_j - \mathbf{x}_i|} \left(\mathbf{I} - \frac{1}{|\mathbf{x}_j - \mathbf{x}_i|^2} (\mathbf{x}_j - \mathbf{x}_i)(\mathbf{x}_j - \mathbf{x}_i)^{\mathrm{T}} \right) \right)$$

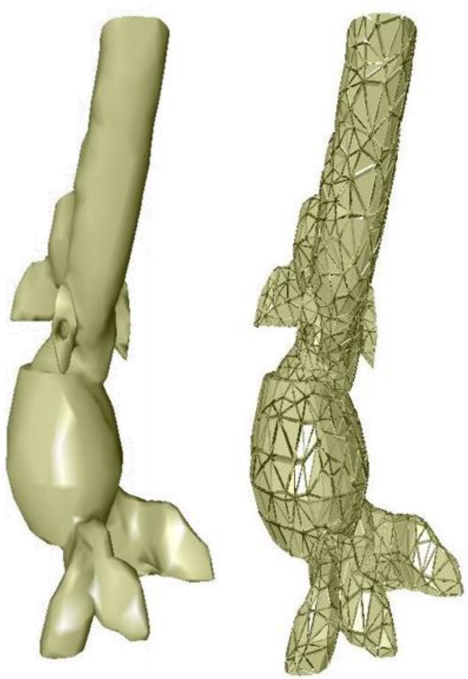
$$= -\mathbf{J}_{i,j}^t = \mathbf{J}_{j,j}^t = -\mathbf{J}_{j,i}^t$$

Outline

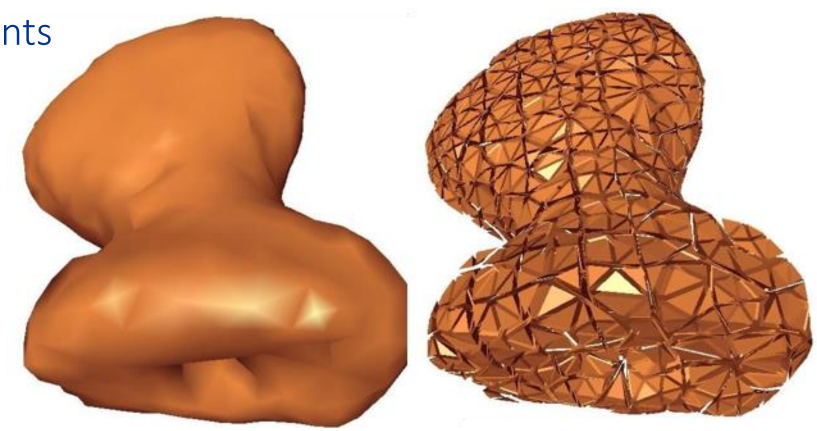
- introduction
- forces
  - examples
  - energy constraints
  - damping
  - plasticity and other effects
- performance
- collision handling
- visualization

Spatial Discretization

- deformable objects are commonly discretized into mass points and simplices
  - line segments in 1D, triangles in 2D, tetrahedrons in 3D

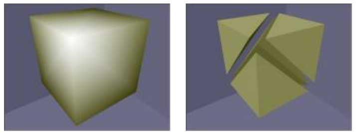


1349 mass points
4562 tetras
6888 springs



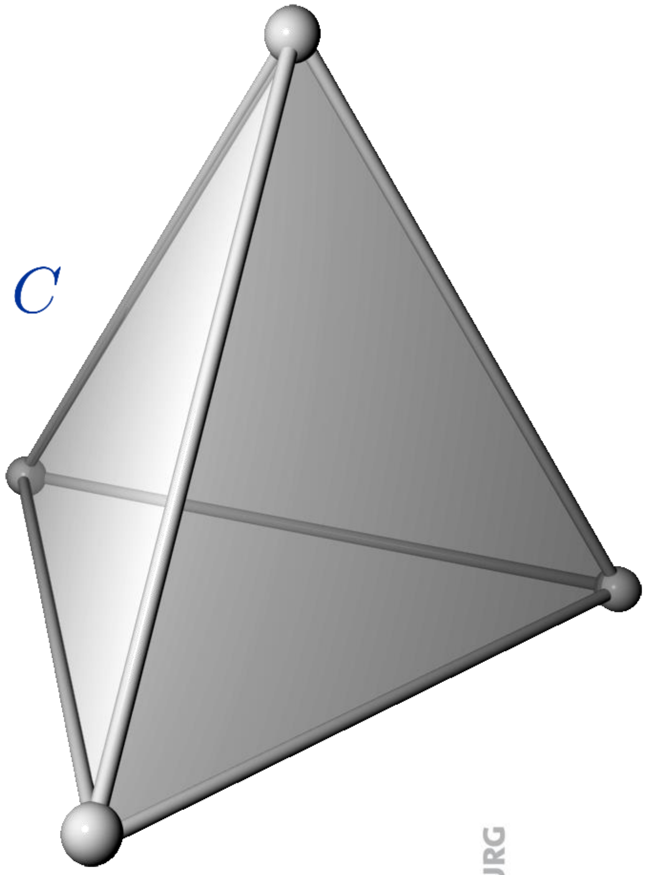
2949 mass points
10257 tetras
15713 springs

8 mass points
5 tetras



Generalized Springs

- can be used to preserve, e. g.,
  - a distance between two points
  - an area defined by three points
  - a volume defined by four points
- forces are derived from constraints C
- C depends on mass point positions
- $C(\mathbf{x}_1, \ldots, \mathbf{x}_n) = 0$
  iff the constraint is met,
  e. g. a current distance equals
  a goal distance, a current area
  equals a goal area, …
- motivation demo



Constraint Forces

- potential energy E based on constraint C

$$E(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \frac{1}{2} k C(\mathbf{x}_1, \ldots, \mathbf{x}_n)^2$$

- $E = 0$ iff the constraint is met
- $E > 0$ iff the constraint is not met

- force at mass point j based on the potential energy E

$$\begin{aligned} \mathbf{F}_j(\mathbf{x}_1, \ldots, \mathbf{x}_n) &= -\frac{\partial}{\partial \mathbf{x}_j} E(\mathbf{x}_1, \ldots, \mathbf{x}_n) \\ &= -kC(\mathbf{x}_1, \ldots, \mathbf{x}_n) \frac{\partial C(\mathbf{x}_1, \ldots, \mathbf{x}_n)}{\partial \mathbf{x}_j} \end{aligned}$$

Constraint Forces

- for a constraint C , the sum of constraint forces at all involved mass points is equal to zero
  $\sum_j \mathbf{F}_j(\mathbf{x}_1, \ldots, \mathbf{x}_n) = 0$

- linear and angular momentum of the system $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ are preserved
- constraint forces are internal forces (conservative forces)

Distance Preservation

- preserve distance L between $\mathbf{x}_1$ and $\mathbf{x}_2$

$$C_d(\mathbf{x}_1, \mathbf{x}_2) = |\mathbf{x}_1 - \mathbf{x}_2| - L$$

$$\mathbf{x}_1 = (x_1, y_1, z_1)^{\mathrm{T}} \quad \mathbf{x}_2 = (x_2, y_2, z_2)^{\mathrm{T}}$$

$$C_d(x_1, y_1, z_1, x_2, y_2, z_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} - L$$

$$\frac{\partial C_d}{\partial \mathbf{x}_1} = \begin{pmatrix} \frac{\partial C_d}{\partial x_1} \\ \frac{\partial C_d}{\partial y_1} \\ \frac{\partial C_d}{\partial z_1} \end{pmatrix} = \frac{1}{|\mathbf{x}_1 - \mathbf{x}_2|} \begin{pmatrix} x_1 - x_2 \\ y_1 - y_2 \\ z_1 - z_2 \end{pmatrix} = \frac{\mathbf{x}_1 - \mathbf{x}_2}{|\mathbf{x}_1 - \mathbf{x}_2|}$$

Distance Preservation

- forces $\mathbf{F}_d$ based on constraint C_d

$$\mathbf{F}_d(\mathbf{x}_1) = -k_d C_d \frac{\partial C_d}{\partial \mathbf{x}_1} = -k_d(|\mathbf{x}_1 - \mathbf{x}_2| - L)\frac{\|\mathbf{x}_1 - \mathbf{x}_2}{|\mathbf{x}_1 - \mathbf{x}_2|}$$

$$\mathbf{F}_d(\mathbf{x}_2) = -k_d C_d \frac{\partial C_d}{\partial \mathbf{x}_2} = k_d(|\mathbf{x}_1 - \mathbf{x}_2| - L)\frac{\|\mathbf{x}_1 - \mathbf{x}_2}{|\mathbf{x}_1 - \mathbf{x}_2|}$$

- $\mathbf{F}_d$ are spring forces with stiffness constant k_d

UNI FREIBURG

Area Preservation

- preserve area A of a triangle $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$
- edges $\mathbf{e}_1 = \mathbf{x}_3 - \mathbf{x}_1 \quad \mathbf{e}_2 = \mathbf{x}_3 - \mathbf{x}_2$
- constraint $C_a(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = \frac{1}{2}|\mathbf{e}_1 \times \mathbf{e}_2| - A$

$$\mathbf{t} = \mathbf{e}_1 \times \mathbf{e}_2 \qquad s = k_a \frac{C_a}{0.5|\mathbf{e}_1 \times \mathbf{e}_2|}$$

- forces $\mathbf{F}_a(\mathbf{x}_1) = s\mathbf{e}_2 \times \mathbf{t}$

$$\mathbf{F}_a(\mathbf{x}_2) = s\mathbf{t} \times \mathbf{e}_1$$

$$\mathbf{F}_a(\mathbf{x}_3) = s\mathbf{t} \times (\mathbf{e}_2 - \mathbf{e}_1)$$

Volume Preservation

- preserve volume V of a tetrahedron $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$
- edges $\mathbf{e}_1 = \mathbf{x}_2 - \mathbf{x}_1 \quad \mathbf{e}_2 = \mathbf{x}_3 - \mathbf{x}_1 \quad \mathbf{e}_3 = \mathbf{x}_4 - \mathbf{x}_1$
- constraint $C_v(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) = \frac{1}{6}\mathbf{e}_1(\mathbf{e}_2 \times \mathbf{e}_3) - V$
- forces

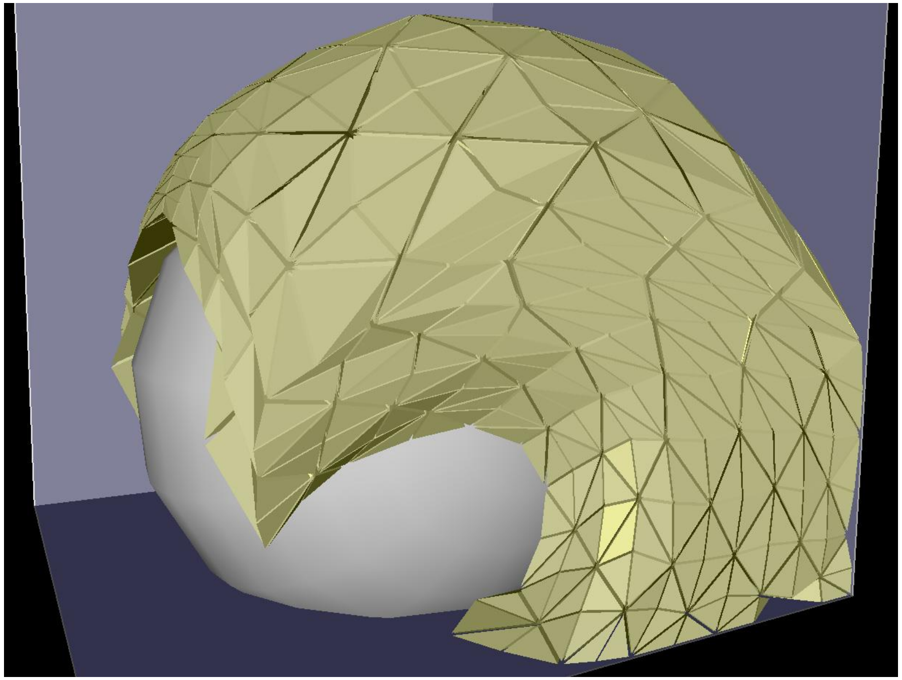
$$\mathbf{F}_v(\mathbf{x}_1) = k_v C_v(\mathbf{e}_2 - \mathbf{e}_1) \times (\mathbf{e}_3 - \mathbf{e}_1)$$

$$\mathbf{F}_v(\mathbf{x}_2) = k_v C_v \mathbf{e}_3 \times \mathbf{e}_2$$

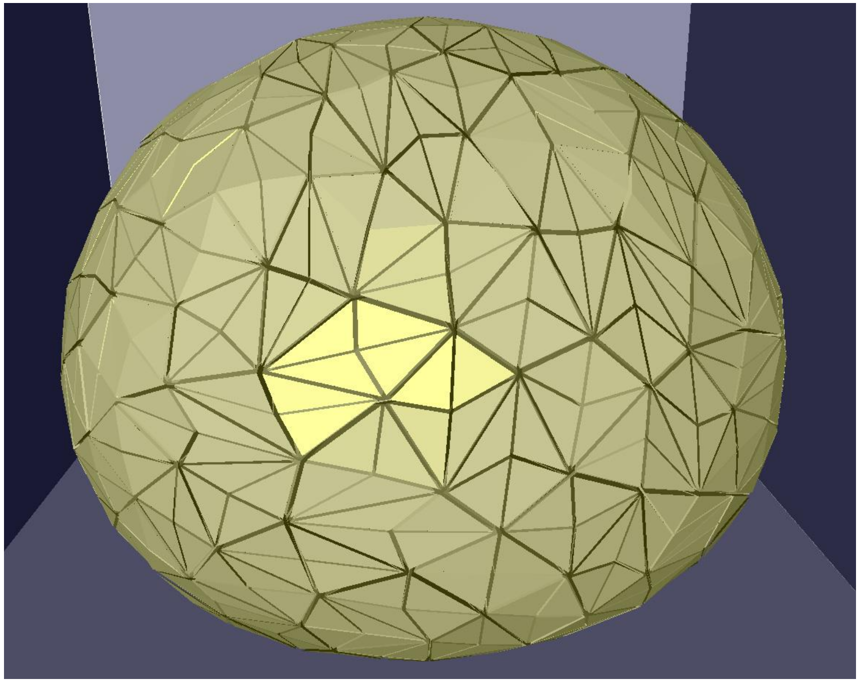
$$\mathbf{F}_v(\mathbf{x}_3) = k_v C_v \mathbf{e}_1 \times \mathbf{e}_3$$

$$\mathbf{F}_v(\mathbf{x}_4) = k_v C_v \mathbf{e}_2 \times \mathbf{e}_1$$

Demos



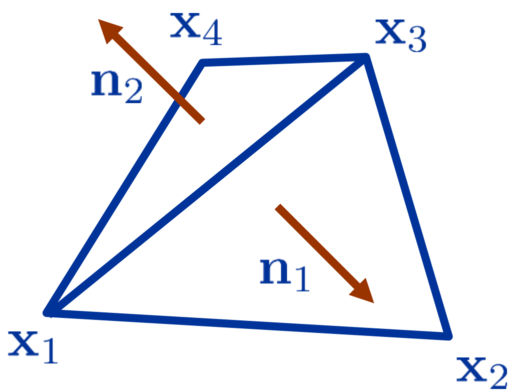
distance preservation vs.
volume preservation



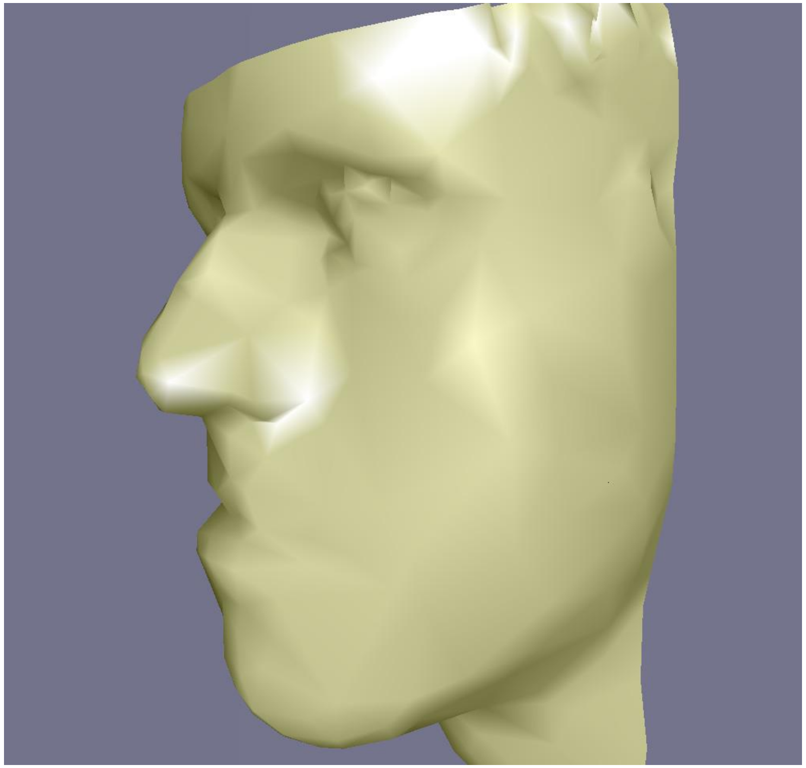
surface tension vs.
volume preservation

Demos

- volume preservation can be used to mimic curvature preservation at adjacent triangles



curvature can be preserved
by preserving the volume
of the virtual tetrahedron (x_1,x_2,x_3,x_4)



volume forces can mimic
bending forces

Constraint Forces - Summary

- powerful mechanism to preserve various characteristics (constraints)
- are internal forces, preserve linear and angular momentum
- are defined for sets of mass points
- can be combined, weighted with stiffness constants
- drawbacks
  - can be computationally expensive
  - non-intuitive parameters in case of combined constraints
  - can be redundant or competing

Outline

- introduction
- forces
  - examples
  - energy constraints
  - damping
  - plasticity and other effects
- performance
- collision handling
- visualization

Damping Forces

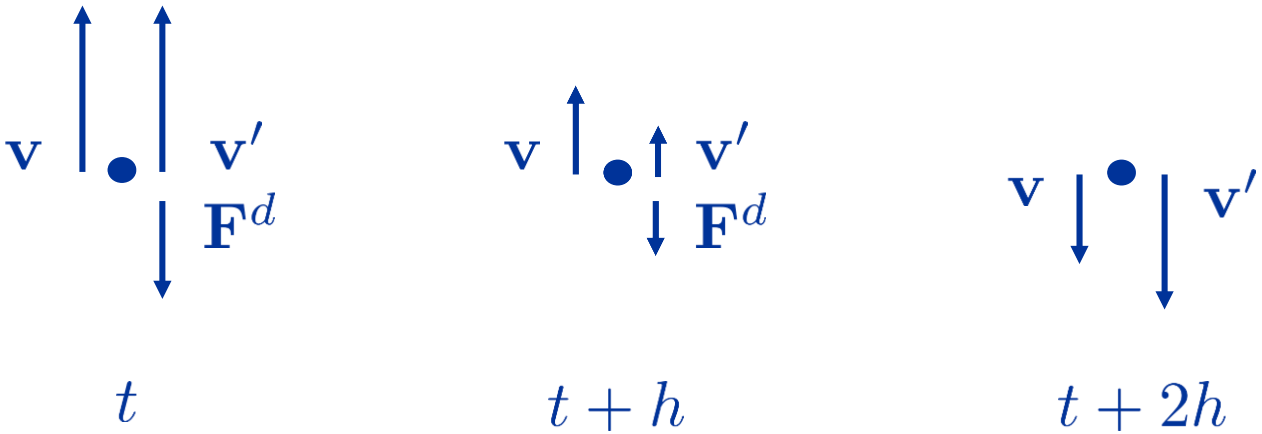
- are proportional to a velocity
- act in the direction opposite to a velocity
- model friction
- can improve the stability of a system
- should not slow down the movement of a system

Point Damping

- damping force according to the velocity of a mass point
- force is applied in opposite direction to the velocity
  $m\ddot{\mathbf{x}}_t = \mathbf{F}_t - \gamma\dot{\mathbf{x}}_t$
- force at a point is "used" for acceleration and damping
  $m\ddot{\mathbf{x}}_t + \gamma\dot{\mathbf{x}}_t = \mathbf{F}_t$
- e.g., mass point under gravity does not accelerate iff gravity and damping cancel out each other

Explicit Point Damping

- damping does not always damp
- $\mathbf{v}$ - velocity without damping
- $\mathbf{v}'$ - velocity with added damping
- $\mathbf{F}^d$ - damping force



Implicit Point Damping

- considering the current velocity for damping can cause problems

$$\mathbf{v}_{t+h} = \mathbf{v}_t + h\frac{1}{m}(\mathbf{F}_t - \gamma \mathbf{v}_t) \qquad = \mathbf{v}_t + h\frac{1}{m}(\mathbf{F}_t - \mathbf{F}_t^d)$$

- considering the velocity of the next time step reduces problems

$$\mathbf{v}_{t+h} = \mathbf{v}_t + h\frac{1}{m}(\mathbf{F}_t - \gamma \mathbf{v}_{t+h}) = \mathbf{v}_t + h\frac{1}{m}(\mathbf{F}_t - \mathbf{F}_{t+h}^d)$$

- can still be directly solvable for $\mathbf{v}_{t+h}$, e.g.,

$$\mathbf{v}_{t+h} = \frac{1}{1+\frac{h\gamma}{m}}\left(\mathbf{v}_t + h\frac{1}{m}\mathbf{F}_t\right)$$

Explicit Spring Damping

- damping force according to the relative velocity of adjacent mass points $\mathbf{x}_1$ and $\mathbf{x}_2$
- normalized direction
  $\hat{\mathbf{d}}_t = \frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}$
- difference of the magnitudes of velocities projected onto $\hat{\mathbf{d}}_t$
  (magnitude of the relative velocity)
  $m_t = \mathbf{v}_{2,t}\hat{\mathbf{d}}_t - \mathbf{v}_{1,t}\hat{\mathbf{d}}_t$
- damping forces
  $\mathbf{F}_{1,t} = \gamma m_t \hat{\mathbf{d}}_t \qquad \mathbf{F}_{2,t} = -\gamma m_t \hat{\mathbf{d}}_t$

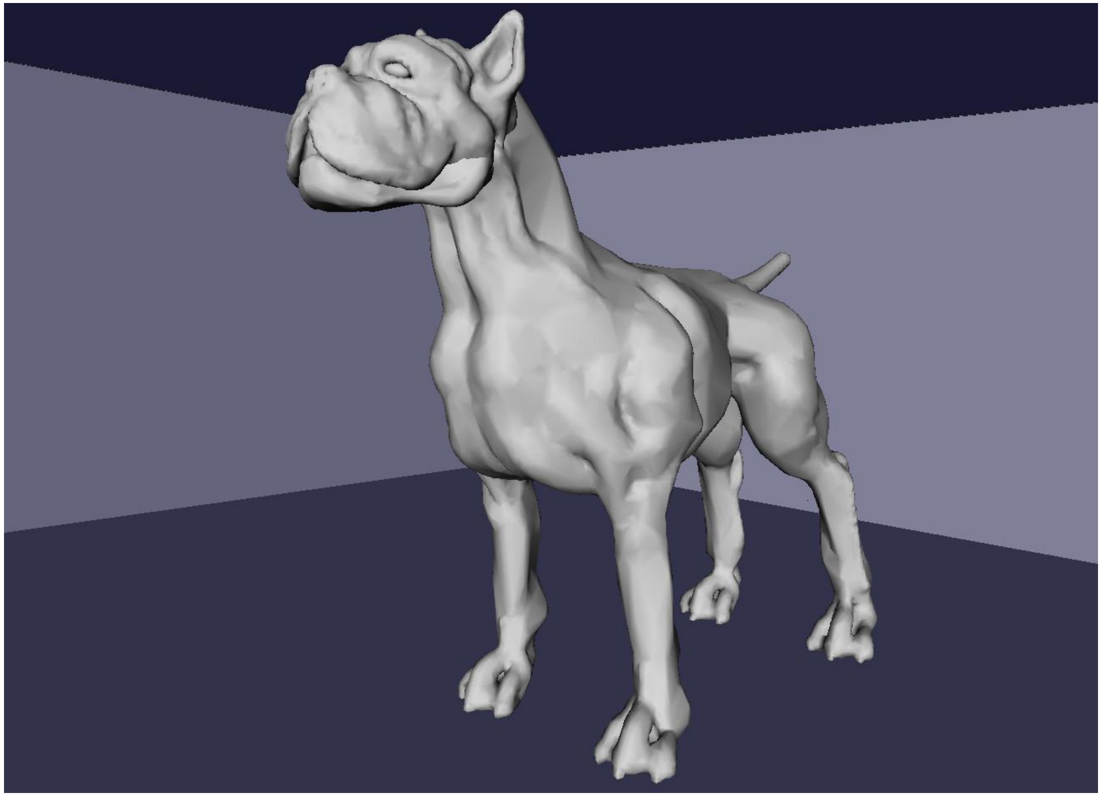
Implicit Spring Damping

- generally more robust

$$\mathbf{F}_{1,t} = \gamma m_{t+h} \hat{\mathbf{d}}_{t+h} \qquad \mathbf{F}_{2,t} = -\gamma m_{t+h} \hat{\mathbf{d}}_{t+h}$$

- implementation in two integration steps
  - first step predicts positions and velocities without damping
  - second step corrects predicted quantities with added damping
- implementation in one integration step
  - predict positions and velocities within the damping force computation, e. g. using Euler
  - prediction and actual integration can be done with different schemes

Demos



benefits and drawbacks of damping

Damping - Summary

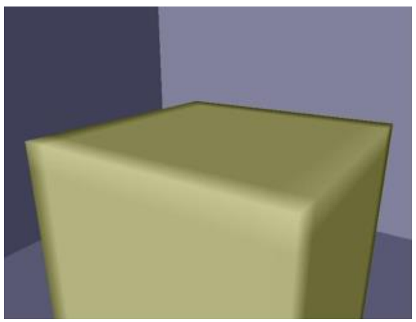
- point and spring damping influence the stability
- implicit forms are preferable due to time discretization
- reduces oscillations
- point damping affects the global object dynamics
- integration schemes can add artificial point damping (which cannot be controlled by the user)
- spring damping does not affect the global object dynamics

Outline

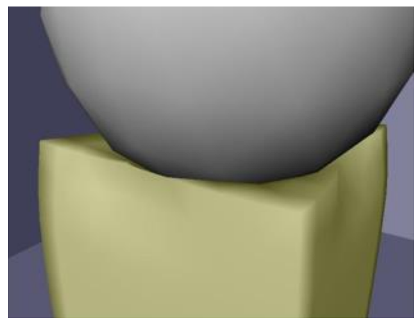
- introduction
- forces
  - examples
  - energy constraints
  - damping
  - plasticity and other effects
- performance
- collision handling
- visualization

Elasticity and Plasticity

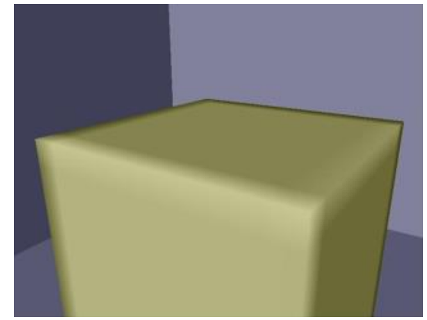
- elastic deformation is reversible
- plastic deformation is not reversible



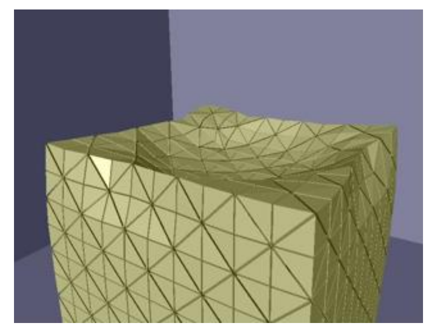
original shape



deformation
due to forces



elastic object
reversible deformation



plastic object
irreversible
deformation

Elasticity and Plasticity

- decomposition of deformation

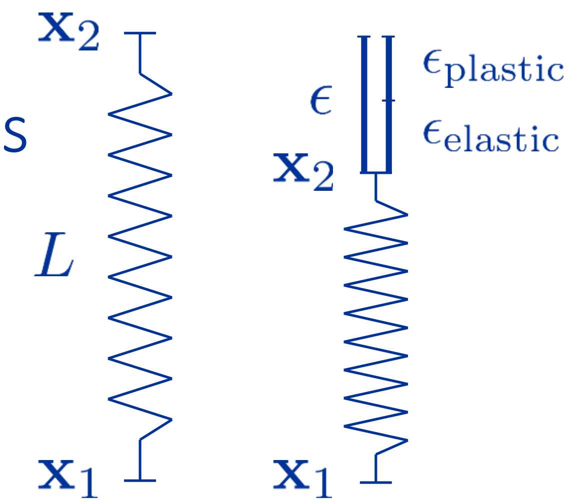
$$\epsilon = \epsilon_{\text{elastic}} + \epsilon_{\text{plastic}}$$

- decomposition of corresponding forces

$$F = k\epsilon = k\epsilon_{\text{elastic}} + k\epsilon_{\text{plastic}}$$

- only elastic forces are considered

$$F_{\text{elastic}} = k\epsilon - k\epsilon_{\text{plastic}} = k\epsilon_{\text{elastic}}$$



Implementation

- initialization

  $\epsilon_{\text{plastic}} = 0$

- update

  compute ϵ

  $\epsilon_{\text{elastic}} = \epsilon - \epsilon_{\text{plastic}}$

  if $\epsilon_{\text{elastic}} > \text{yield}$ then $\epsilon_{\text{plastic}} = \epsilon_{\text{plastic}} + \text{creep}\ \epsilon_{\text{elastic}}$

  if $\epsilon_{\text{plastic}} > \max$ then $\epsilon_{\text{plastic}} = \max$

- $\text{yield}, \text{creep}, \max$ are user-defined parameters

Elastic and Plastic Deformation

Resting Distance, Area, Volume

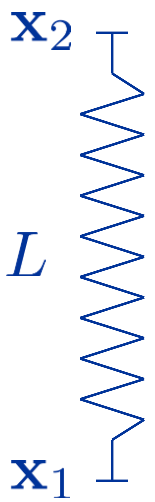
- plastic deformation corresponds to adjusting the resting distance between mass points
- principle can also be applied to other properties, e. g. area, volume
- adjustment of resting states causes internal forces
- can be used for effects such as contraction

Strain Limiting

- limited deformation
- geometric, position-based implementation

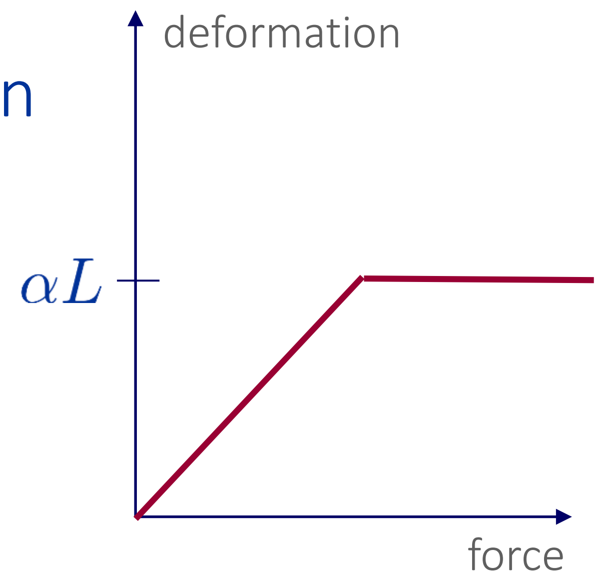
$$\text{if } |\mathbf{x}_1 - \mathbf{x}_2| > \alpha L \text{ then}$$

$$\begin{aligned}
\mathbf{x}_1 &= \mathbf{x}_1 + \frac{m_2}{m_1 + m_2}(|\mathbf{x}_2 - \mathbf{x}_1| - \alpha L)\frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|} \\
\mathbf{x}_2 &= \mathbf{x}_2 - \frac{m_1}{m_1 + m_2}(|\mathbf{x}_2 - \mathbf{x}_1| - \alpha L)\frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}
\end{aligned}$$



Strain Limiting

- implementation approximates a bi-phasic force-deformation relation
- position update can be performed after the integration step
- iterative implementation for mass-point systems
- preserves linear and angular momentum
  - corresponds to some internal forces



Forces - Summary

- external forces change the linear and angular momentum of a system, e.g. gravity, point damping
- internal forces can preserve characteristics, e.g. distances, areas, volumes
- damping forces improve the stability of a system
- resting length adjustments, symmetric position or momentum adjustments can mimic internal forces, e. g. for plasticity, stiff springs
- challenge:
  stable simulation of stiff, non-oscillating deformable objects without explicit or artificial point damping

Outline

- introduction
- forces
- performance
- collision handling
- visualization

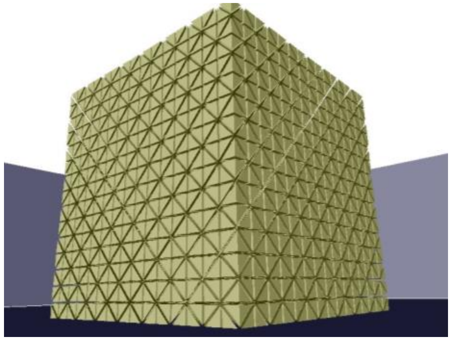
Performance

- criteria
  - system updates per second (frames per second)
  - simulation time step

- parameters
  - number of primitives (mass points, distances, volumes ...)
  - internal and external forces
  - numerical integration scheme
  - additional costs for, e.g., collision handling, rendering, ...

Performance - Example

- cube with 4096 mass points, 16875 tetrahedrons, 22320 springs, distance and volume forces, gravity, Pentium 4, 2GHz



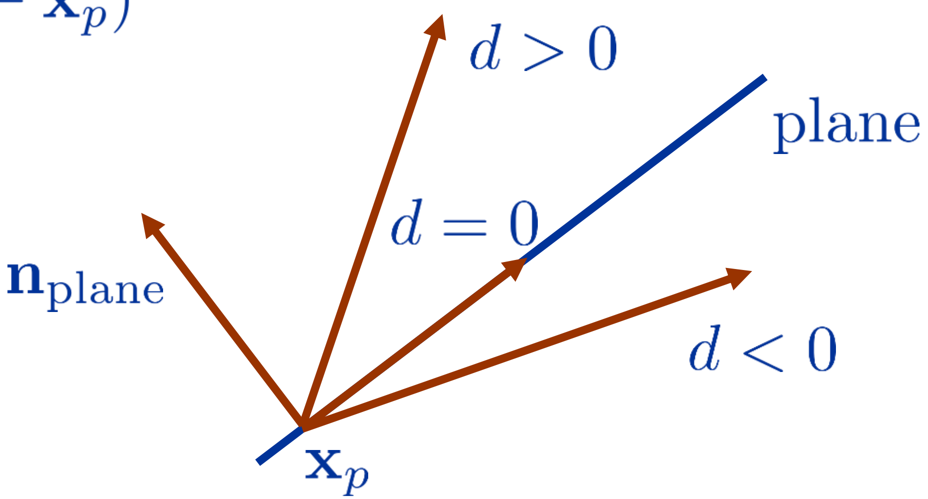
method	error order	time step [ms]	comp. time [ms]	ratio
expl. Euler	1	0.5	9.5	0.05
RK 2	2	3.8	18.9	0.20
impl. Euler	1	49.0	172.0	0.28
RK 4	4	17.0	50.0	0.34
Verlet	3	11.5	9.5	1.21

Outline

- introduction
- forces
- performance
- collision handling
- visualization

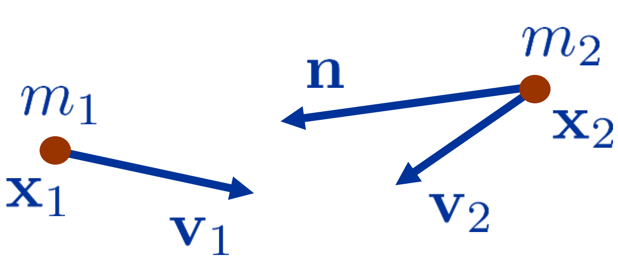
Plane Representation

- in 3D, a plane can be defined with a point $\mathbf{x}_p$ on the plane and a normalized plane normal $\mathbf{n}_{\text{plane}}$
- the plane is the set of points $\mathbf{x}$ with $\mathbf{n}_{\text{plane}} \cdot (\mathbf{x} - \mathbf{x}_p) = 0$
- for a point $\mathbf{x}$, the distance to the plane is $d = \mathbf{n}_{\text{plane}} \cdot (\mathbf{x} - \mathbf{x}_p)$



Collision Response

- if a collision is detected, i. e. $d < 0$,
  a collision impulse is computed that prevents the interpenetration of the mass point and the plane (wall)
- we first consider the case of a particle-particle collision with $\mathbf{n}$ being the normalized direction from $\mathbf{x}_2$ to $\mathbf{x}_1$



- the response scheme is later adapted to the particle-plane case

Coordinate Systems

- velocities $\mathbf{v}$ before the collision response and velocities $\mathbf{V}$ after the collision response are considered in the coordinate system defined by collision normal $\mathbf{n}$ and two orthogonal normalized tangent axes $\mathbf{t}$ and $\mathbf{k}$
- e. g. $$\begin{pmatrix} v_{1,n} \\ v_{1,t} \\ v_{1,k} \end{pmatrix} = \begin{pmatrix} n_x & n_y & n_z \\ t_x & t_y & t_z \\ k_x & k_y & k_z \end{pmatrix} \begin{pmatrix} v_{1,x} \\ v_{1,y} \\ v_{1,z} \end{pmatrix}$$
- the velocity $\mathbf{V}$ after the response is transformed back

$$\begin{pmatrix} V_{1,x} \\ V_{1,y} \\ V_{1,z} \end{pmatrix} = \begin{pmatrix} n_x & t_x & k_x \\ n_y & t_y & k_y \\ n_z & t_z & k_z \end{pmatrix} \begin{pmatrix} V_{1,n} \\ V_{1,t} \\ V_{1,k} \end{pmatrix}$$

Concept

- conservation of momentum

$$
\begin{aligned}
m_1 V_{1,n} - m_1 v_{1,n} &= P_n & \qquad m_2 V_{2,n} - m_2 v_{2,n} &= -P_n \\
m_1 V_{1,t} - m_1 v_{1,t} &= P_t & \qquad m_2 V_{2,t} - m_2 v_{2,t} &= -P_t \\
m_1 V_{1,k} - m_1 v_{1,k} &= P_k & \qquad m_2 V_{2,k} - m_2 v_{2,k} &= -P_k
\end{aligned}
$$

- coefficient of restitution, $e = 1$ elastic, $e = 0$ inelastic
  $V_{1,n} - V_{2,n} = -e(v_{1,n} - v_{2,n})$
- friction opposes sliding motion along $\mathbf{t}$ and $\mathbf{k}$
  $P_t = \mu P_n \qquad P_k = \mu P_n$

Linear System

- nine equations, nine unknowns

$$
\begin{pmatrix}
0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\
-\mu & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\mu & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & m_1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & m_1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & m_1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & m_2 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & m_2 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & m_2
\end{pmatrix}
\begin{pmatrix}
P_n \\ P_k \\ P_t \\ V_{1,n} \\ V_{1,t} \\ V_{1,k} \\ V_{2,n} \\ V_{2,t} \\ V_{2,k}
\end{pmatrix}
=
\begin{pmatrix}
-e(v_{1,n} - v_{2,n}) \\ 0 \\ 0 \\ m_1 v_{1,n} \\ m_1 v_{1,t} \\ m_1 v_{1,k} \\ m_2 v_{2,n} \\ m_2 v_{2,t} \\ m_2 v_{2,k}
\end{pmatrix}
$$

Linear System

- inverse

$$
\begin{pmatrix}
\frac{m_1 m_2}{m_1+m_2} & 0 & 0 & -\frac{m_2}{m_1+m_2} & 0 & 0 & \frac{m_1}{m_1+m_2} & 0 & 0 \\
\frac{m_1 m_2 \mu}{m_1+m_2} & 1 & 0 & -\frac{m_2 \mu}{m_1+m_2} & 0 & 0 & \frac{m_1 \mu}{m_1+m_2} & 0 & 0 \\
\frac{m_1 m_2 \mu}{m_1+m_2} & 0 & 1 & -\frac{m_2 \mu}{m_1+m_2} & 0 & 0 & \frac{m_1 \mu}{m_1+m_2} & 0 & 0 \\
\frac{m_2}{m_1+m_2} & 0 & 0 & \frac{1}{m_1+m_2} & 0 & 0 & \frac{1}{m_1+m_2} & 0 & 0 \\
\frac{m_2 \mu}{m_1+m_2} & \frac{1}{m_1} & 0 & -\frac{m_2 \mu}{m_1(m_1+m_2)} & \frac{1}{m_1} & 0 & \frac{\mu}{m_1+m_2} & 0 & 0 \\
\frac{m_2 \mu}{m_1+m_2} & 0 & \frac{1}{m_1} & -\frac{m_2 \mu}{m_1(m_1+m_2)} & 0 & \frac{1}{m_1} & \frac{\mu}{m_1+m_2} & 0 & 0 \\
-\frac{m_1}{m_1+m_2} & 0 & 0 & \frac{1}{m_1+m_2} & 0 & 0 & \frac{1}{m_1+m_2} & 0 & 0 \\
-\frac{m_1 \mu}{m_1+m_2} & -\frac{1}{m_2} & 0 & \frac{\mu}{m_1+m_2} & 0 & 0 & -\frac{m_1 \mu}{m_2(m_1+m_2)} & \frac{1}{m_2} & 0 \\
-\frac{m_1 \mu}{m_1+m_2} & 0 & -\frac{1}{m_2} & \frac{\mu}{m_1+m_2} & 0 & 0 & -\frac{m_1 \mu}{m_2(m_1+m_2)} & 0 & \frac{1}{m_2}
\end{pmatrix}
$$

Particle-Plane

- plane has infinite mass and does not move: $v_2 = V_2 = 0$
- columns 2, 3, 7, 8, 9 do not contribute to the solution
- to solve for the particle velocity V_1 after collision response, rows 4, 5, 6 have to be considered

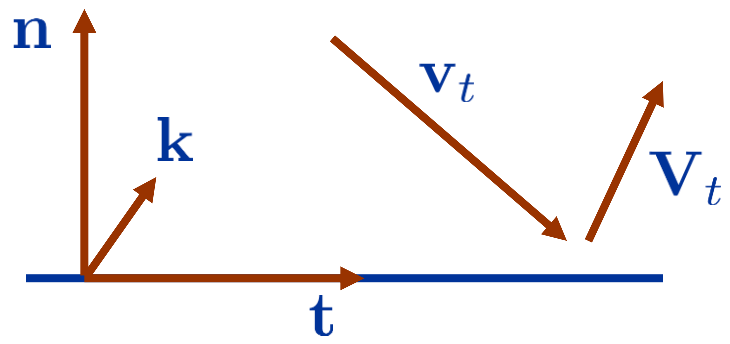
$$\begin{pmatrix} \frac{m_2}{m_1+m_2} & \frac{1}{m_1+m_2} & 0 & 0 \\ \frac{m_2\mu}{m_1+m_2} & -\frac{m_2\mu}{m_1(m_1+m_2)} & \frac{1}{m_1} & 0 \\ \frac{m_2\mu}{m_1+m_2} & -\frac{m_2\mu}{m_1(m_1+m_2)} & 0 & \frac{1}{m_1} \end{pmatrix} \begin{pmatrix} -ev_{1,n} \\ m_1 v_{1,n} \\ m_1 v_{1,t} \\ m_1 v_{1,k} \end{pmatrix} = \begin{pmatrix} V_{1,n} \\ V_{1,t} \\ V_{1,k} \end{pmatrix}$$

- plane has infinite mass

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ \mu & -\frac{\mu}{m_1} & \frac{1}{m_1} & 0 \\ \mu & -\frac{\mu}{m_1} & 0 & \frac{1}{m_1} \end{pmatrix} \begin{pmatrix} -ev_{1,n} \\ m_1 v_{1,n} \\ m_1 v_{1,t} \\ m_1 v_{1,k} \end{pmatrix} = \begin{pmatrix} V_{1,n} \\ V_{1,t} \\ V_{1,k} \end{pmatrix}$$

Implementation

$$\begin{aligned} V_{t,n} &= -e v_{t,n} \\ V_{t,t} &= v_{t,t} - \mu(e+1) v_{t,n} \\ V_{t,k} &= v_{t,k} - \mu(e+1) v_{t,n} \end{aligned}$$



- μ is difficult to handle
- $|V_{t,t}| \leq |v_{t,t}|$ and $\mathrm{sign}(V_{t,t}) = \mathrm{sign}(v_{t,t})$ should be guaranteed
- $V_{t,t} = \mu v_{t,t} \qquad V_{t,k} = \mu v_{t,k} \qquad 0 \leq \mu \leq 1$ is a useful simplification

Position Update

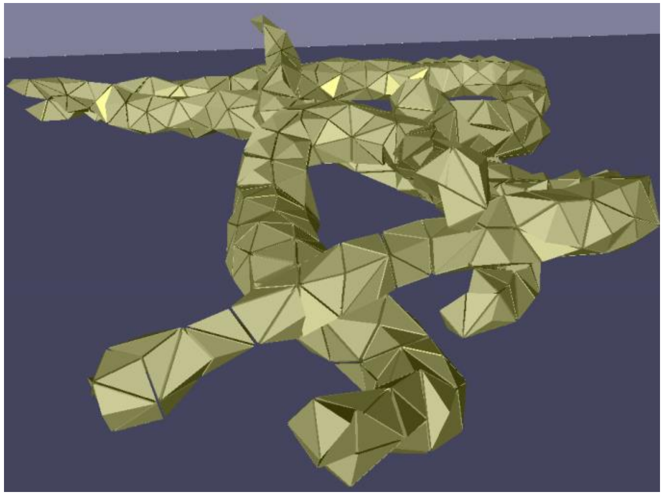
- the collision impulse updates the velocity
- however, the point is still in collision $(d < 0)$
- for low velocities, the position update in the following integration step may not be sufficient to resolve the collision
- therefore, the position should be updated as well, e.g. $\mathbf{x}_{t+h} = \mathbf{x}_t + d \cdot \mathbf{n}$ which projects the point onto the plane
- the position update is not physically-motivated, it just resolves problems due to discrete time steps

Outline

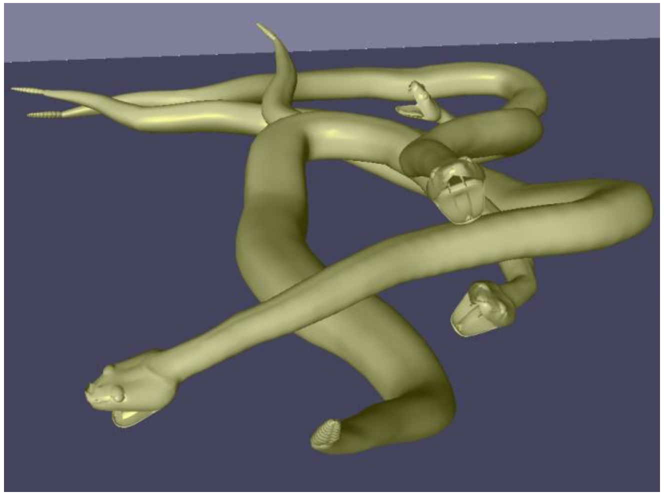
- introduction
- forces
- performance
- collision handling
- visualization

Concept

- geometric combination of
  - a low-resolution tetrahedral mesh for simulation and
  - a high-resolution triangular mesh for visualization
- coupling by Barycentric coordinates of a surface point with respect to a corresponding tetrahedron



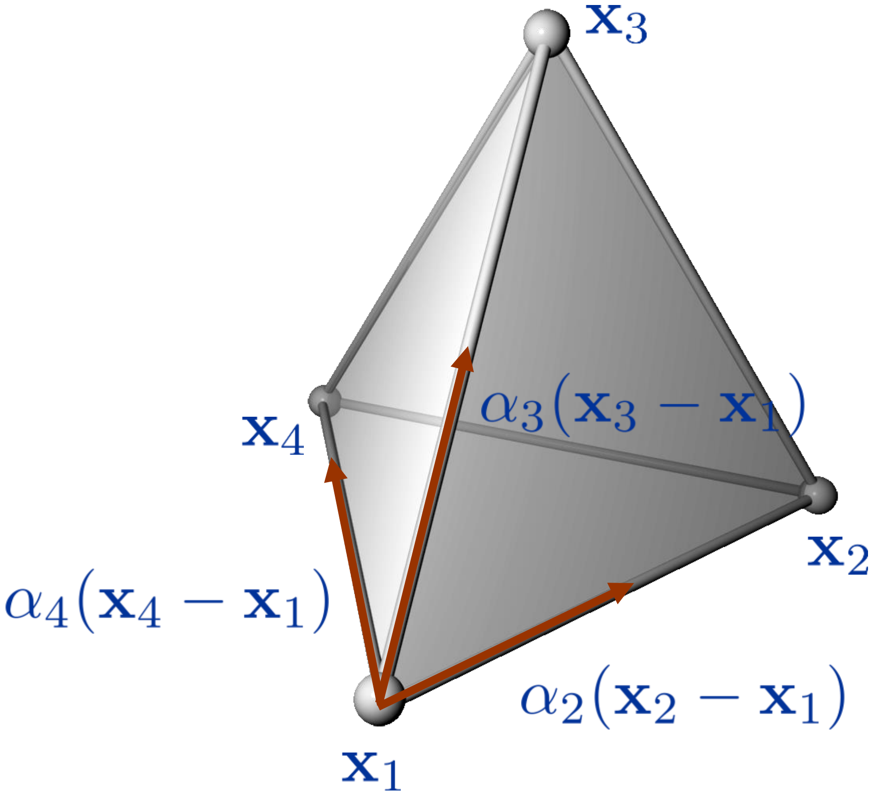
simulation



visualization

Surface-Volume Coupling

- a point $\mathbf{x}_s$ can be represented with the points of a tetrahedron



Barycentric Coordinates in 3D

- a point $\mathbf{x}_s$ can be represented using $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$

$$\mathbf{x}_s = \mathbf{x}_1 + \alpha_2(\mathbf{x}_2 - \mathbf{x}_1) + \alpha_3(\mathbf{x}_3 - \mathbf{x}_1) + \alpha_4(\mathbf{x}_4 - \mathbf{x}_1)$$

$$\mathbf{x}_s = (1 - \alpha_2 - \alpha_3 - \alpha_4)\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + \alpha_3\mathbf{x}_3 + \alpha_4\mathbf{x}_4$$

$$\mathbf{x}_s = \alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + \alpha_3\mathbf{x}_3 + \alpha_4\mathbf{x}_4 \qquad \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$$

- $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ are Barycentric coordinates of $\mathbf{x}_s$ with respect to $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$

Properties

- $0 < \alpha_i < 1$
  $\mathbf{x}_s$ is inside the convex combination of $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$, i.e. inside the tetrahedron

- $\alpha_i = 0 \vee \alpha_i = 1$
  $\mathbf{x}_s$ is on the surface of the tetrahedron

- $\alpha_i < 0 \vee \alpha_i > 1$
  $\mathbf{x}_s$ is outside the tetrahedron

Computation

$$\mathbf{x}_s = \mathbf{x}_1 + \alpha_2(\mathbf{x}_2 - \mathbf{x}_1) + \alpha_3(\mathbf{x}_3 - \mathbf{x}_1) + \alpha_4(\mathbf{x}_4 - \mathbf{x}_1)$$

- leads to the following system

$$\begin{pmatrix} (\mathbf{x}_2 - \mathbf{x}_1) & (\mathbf{x}_3 - \mathbf{x}_1) & (\mathbf{x}_4 - \mathbf{x}_1) \end{pmatrix} \begin{pmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} = \mathbf{x}_s - \mathbf{x}_1$$

- singular, if two edges of the tetrahedron are parallel !
- α_1 is computed as $\alpha_1 = 1 - \alpha_2 - \alpha_3 - \alpha 4$

Implementation

- data structure
  - for each point of the surface mesh,
    store Barycentric coords and the corresponding tetrahedron
- pre-processing
  - for each surface point, determine the closest tetrahedron of the simulation mesh (point of the surface mesh should be located inside a tetrahedron)
  - for each surface point, compute its Barycentric coords with respect to the corresponding tetrahedron
- in each simulation step
  - for each surface point, compute its position from its Barycentric coords and the positions of the mass points of the corresponding tetrahedron
- demo

Summary Mass-Point Systems

- forces, e. g. gravity, energy constraints, damping, plasticity
- numerical integration schemes (see particle systems)
- collision handling for planes
- visualization, combination of low-resolution simulation meshes with high-resolution visualization meshes

UNI FREIBURG

References

- Ian Millington,
  "Game Physics Engine Development",
  Elsevier Amsterdam, ISBN-13: 978-0-12-369471-3, 2007.
- David H. Eberly,
  "Game Physics",
  Elsevier Amsterdam, ISBN 1-55860-740-4, 2003.
- Murilo G. Coutinho,
  "Dynamic Simulations of Multibody Systems",
  Springer Berlin, ISBN 0-387-95192-X, 2001.
- Andrew Witkin, Kurt Fleischer, Alan Barr,
  "Energy Constraints on Parameterized Models",
  SIGGRAPH '87, ACM New York, pp. 225-232, 1987.
- M. Teschner, B. Heidelberger, M. Mueller, M. Gross,
  "**A Versatile and Robust Model for Geometrically Complex Deformable Solids**", *Proc. Computer Graphics International CGI'04*, Crete, Greece, pp. 312-319, June 16-19, 2004.