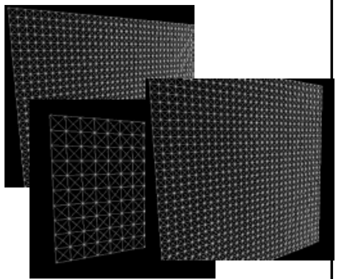
ETH Zurich

Modeling Dynamic Deformation with Mass Points and Springs



Matthias Teschner
Seminar – Wintersemester 02/03

Deformable Objects

ETH Zurich

1. Discretization of an object into **mass points**
2. Representation of forces between mass points with **springs**
3. Computation of the dynamics

➡ deformable mass-spring system

Teschner

Outline

ETH Zurich

Motivation
Model Components
- Mass Points
- Springs
- Forces

Computation of the Dynamic Behavior
- Explicit Numerical Integration
- Implicit Numerical Integration
- Higher-Order Numerical Integration

Stability and Performance Aspects
- Performance
- Time and Space Adaptive Sampling
- Damping
- Force-Deformation Relationship
- Model Topology

Virtual Clothing

ETH Zurich

- Simulation of cloth based on deformable surfaces (polygonal mesh)
- Realistic simulation of cloth with different fabrics such as wool, cotton or silk for garment design

Thalmann

Baraff

Strasser

Virtual Clothing

ETH Zurich

Thalmann
MIRALab

Virtual Clothing

ETH Zurich

Choi
Seoul National University

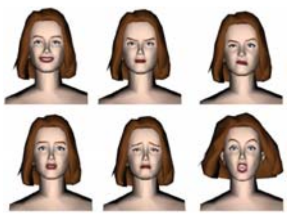
Facial Animation

ETH Zurich

- Simulation of facial expressions based on
  - deformable surfaces
  - deformable volumes
  - muscles
- Animation of face models from speech and mimic parameters

Thalmann

Terzopoulos



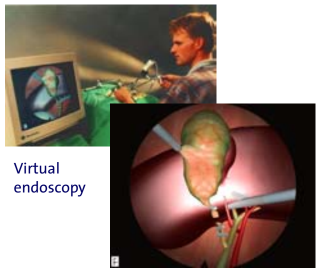
Medical Simulation

ETH Zurich

- Simulation of deformable soft tissue
- Surgical planning
- Medical training
- Swiss NCCR Co-Me **www.co-me.ch**

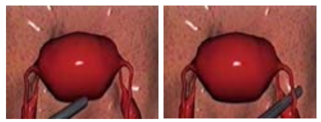
Teschner

Prediction of the surgical outcome in craniofacial surgery



Virtual endoscopy

Kuehnapfel



Laparoscopic (endoscopic) simulation

Szekely

Artificial Life

ETH Zurich

- Simulation of systems consisting of
  - sensors for perception
  - behavioral rules
  - physically-based representation of motor functions

Terzopoulos

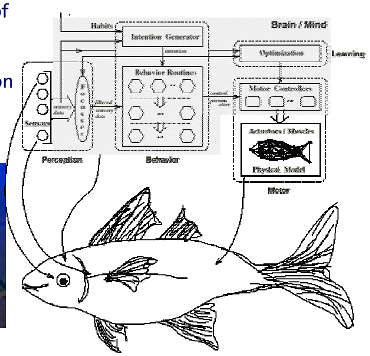
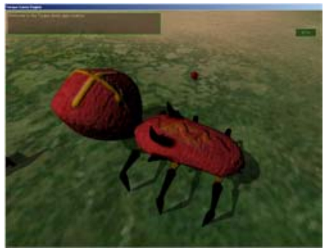


Figure 1: Control and information flow in artificial fish.

Games

ETH Zurich

- Deformable structures are challenging for computer games



Niederberger
Gross

Mueller
Novodex

Outline

ETH Zurich

Motivation
Model Components
- Mass Points
- Springs
- Forces

Computation of the Dynamic Behavior
- Explicit Numerical Integration
- Implicit Numerical Integration
- Higher-Order Numerical Integration

Stability and Performance Aspects
- Performance
- Time and Space Adaptive Sampling
- Damping
- Force-Deformation Relationship
- Model Topology

Mass Points

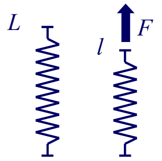
ETH Zurich

```
class POINT
{
   public:
      float mass;
      float position[3];

      . . .
}
```

Elastic Springs

ETH Zurich



Spring stiffness is characterized by k
Initial spring length L
Current spring length l

Deformation linear proportional to force:

$$F = k(L - l)$$ Hooke's law

Elasticity: Ability of a spring to return to its initial form when the deforming force is removed.

Elastic Springs

ETH Zurich

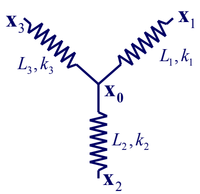
```
class SPRING
{
  public:
    POINT *point1;
    POINT *point2;
    float  stiffness;     // k
    float  initialLength; // L
    float  currentLength; // l

    . . .
}
```


Forces at Mass Points

ETH Zurich

Internal forces $\mathbf{F}^{\text{int}}$



$$\mathbf{F}_0^{\text{int}} = \sum_{i|i\in\{1,2,3\}} k_i\left(|\mathbf{x}_i - \mathbf{x}_0| - L_i\right)\frac{\mathbf{x}_i - \mathbf{x}_0}{|\mathbf{x}_i - \mathbf{x}_0|}$$

External forces $\mathbf{F}^{\text{ext}}$

- Gravity
- All forces, which are not caused by springs

Resulting force at point i

$$\mathbf{F}_i = \mathbf{F}_i^{\text{int}} + \mathbf{F}_i^{\text{ext}}$$

Mass Points

ETH Zurich

```
class POINT
{
   public:
      float mass;
      float position[3];

      float velocity[3];
      float force[3];
      int *adjacentSpring;  // List of adjacent springs

      . . .
}
```

Model - Summary

ETH Zurich

- **Discretization** of an object into mass points
- Definition of the **connectivity** (topology, adjacencies of mass points)
- **Model parameters**:
  - **Points**: mass, initial position, velocity
  - **Springs**: stiffness, initial length
  - Definition of **external forces** (gravity)

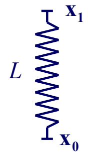
Outline

ETH Zurich

Static Equilibrium

ETH Zurich

System of mass points $\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1)^T$



$$\mathbf{F}_0^{\text{int}} = k \frac{(|\mathbf{x}_1 - \mathbf{x}_0| - L)}{|\mathbf{x}_1 - \mathbf{x}_0|}(\mathbf{x}_1 - \mathbf{x}_0)$$

$$-\mathbf{KX} = -\begin{pmatrix} k \frac{(|\mathbf{x}_1 - \mathbf{x}_0| - L)}{|\mathbf{x}_1 - \mathbf{x}_0|} & -k \frac{(|\mathbf{x}_1 - \mathbf{x}_0| - L)}{|\mathbf{x}_1 - \mathbf{x}_0|} \\ -k \frac{(|\mathbf{x}_1 - \mathbf{x}_0| - L)}{|\mathbf{x}_1 - \mathbf{x}_0|} & k \frac{(|\mathbf{x}_1 - \mathbf{x}_0| - L)}{|\mathbf{x}_1 - \mathbf{x}_0|} \end{pmatrix} \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{pmatrix}$$

Static Equilibrium: $\mathbf{F}^{\text{int}} = -\mathbf{KX} = \mathbf{F}^{\text{ext}}$

From Statics to Dynamics

ETH Zurich

Consideration of **dynamics**

$-\mathbf{KX} = \mathbf{F}^{\text{ext}}$ ➡ $0 = \mathbf{KX}(t) + \mathbf{F}^{\text{ext}}(t)$

Motion equation for a system of mass points

$$\mathbf{M}\frac{d^2\mathbf{X}(t)}{dt^2} = \mathbf{KX}(t) + \mathbf{F}^{\text{ext}}(t)$$

Incorporation of **damping**

$$\mathbf{M}\frac{d^2\mathbf{X}(t)}{dt^2} + \mathbf{D}\frac{d\mathbf{X}(t)}{dt} = \mathbf{KX}(t) + \mathbf{F}^{\text{ext}}(t)$$

Mass-Point Dynamics

ETH Zurich

Motion equation for mass point i at time point t

mass — position — damping coefficient — force at mass point

$$m_i \frac{d^2\mathbf{x}_i(t)}{dt^2} + \gamma \frac{d\mathbf{x}_i(t)}{dt} = \mathbf{F}_i$$

acceleration force — damping force linear proportional to velocity (Stokes friction)

- second-order differential equation
- Newton's equation or motion equation in Langrange form
- force **F** is used for acceleration and damping
- simplified form m**a** = **F** if damping is neglected

Rewriting the Problem

ETH Zurich

Reduction of an second-order differential equation to two **coupled** first-order differential equations.

$$m_i \frac{d^2\mathbf{x}_i(t)}{dt^2} + \gamma \frac{d\mathbf{x}_i(t)}{dt} = \mathbf{F}_i$$

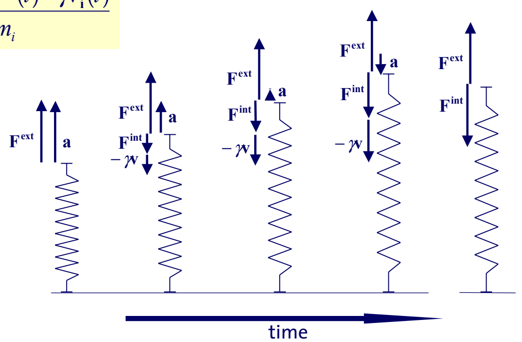
$$\frac{d\mathbf{x}_i(t)}{dt} = \mathbf{v}_i(t) \qquad \frac{d\mathbf{v}_i(t)}{dt} = \frac{\mathbf{F}_i(t) - \gamma\mathbf{v}_i(t)}{m_i}$$

velocity — acceleration

Example

ETH Zurich

$$\mathbf{a}(t) = \frac{\mathbf{F}_i^{int}(t) + \mathbf{F}_i^{ext}(t) - \gamma\mathbf{v}_i(t)}{m_i}$$



Problem

ETH Zurich

We have: Initial position **x**
Initial velocity **v**
Derivative of position **x** with respect to time.
Derivative of velocity **v** with respect to time.

$$\frac{d\mathbf{x}_i(t)}{dt} = \mathbf{v}_i(t) \qquad \frac{d\mathbf{v}_i(t)}{dt} = \frac{\mathbf{F}_i(t) - \gamma\mathbf{v}_i(t)}{m_i}$$

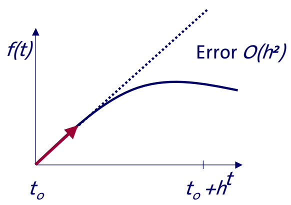
Goal: Computation of position **x** over time.

Explicit Numerical Integration

ETH Zurich

Euler Method

Leonard Euler:
1707 (Basel) – 1783 (Petersburg)



- Initial value $f(t_o)$
- Compute the derivative at t_o
- Move from t_o to $t_o + h$ using the derivative at t_o

M Teschner - Deformable Modeling 25

Euler Method

ETH Zurich

$$\mathbf{x}'(t) = \mathbf{v}(t) \qquad \mathbf{v}'(t) = \frac{\mathbf{F}(t) - \gamma\mathbf{v}(t)}{m}$$

Start with initial values	$\mathbf{x}(t_0) = \mathbf{x}_0 \quad \mathbf{v}(t_0) = \mathbf{v}_0$
Compute	$\mathbf{v}'(t_0) \quad \mathbf{x}'(t_0)$
Assume	$\mathbf{v}'(t) = \mathbf{v}'(t_0) \quad \mathbf{x}'(t) = \mathbf{x}'(t_0) \quad t_0 \le t \le t_0 + h$
Compute	$\mathbf{x}(t_0 + h) = \mathbf{x}(t_0) + h\mathbf{x}'(t_0) = \mathbf{x}(t_0) + h\mathbf{v}(t_0)$
Compute	$\mathbf{v}(t_0 + h) = \mathbf{v}(t_0) + h\mathbf{v}'(t_0) = \mathbf{v}(t_0) + h\frac{\mathbf{F}(t_0) - \gamma\mathbf{v}(t_0)}{m}$

(**F** *(t)* is computed from **x** *(t)* and external forces!)

M Teschner - Deformable Modeling 26

Mass Point

ETH Zurich

```
class POINT
{
   public:
      float mass;
      float position[3];
      float velocity[3];
      float force[3];
      float damping;
      int *adjacentSpring;

      void internalForces();
      void addExternalForces(float extForce[3]);
      void velocityEuler(float timeStep);
      void positionEuler(float timeStep);
}
```

M Teschner - Deformable Modeling 27

Coupled Differential Equations

ETH Zurich

$$\mathbf{x}'(t) = \mathbf{v}(t) \qquad \mathbf{v}'(t) = \frac{\mathbf{F}(t) - \gamma\mathbf{v}(t)}{m}$$

Euler step from t_o to $t_o + h$

$$\mathbf{x}(t_0 + h) = \mathbf{x}(t_0) + h\mathbf{v}(t_0) \qquad \mathbf{v}(t_0 + h) = \mathbf{v}(t_0) + h\frac{\mathbf{F}(t_0) - \gamma\mathbf{v}(t_0)}{m}$$

$$\mathbf{x}(t_0 + 2h) = \mathbf{x}(t_0 + h) + h\mathbf{v}(t_0 + h)$$

Next Euler step from $t_o + h$ to $t_o + 2h$

$$\mathbf{v}(t_0 + 2h) = \mathbf{v}(t_0 + h) + h\frac{\mathbf{F}(t_0 + h) - \gamma\mathbf{v}(t_0 + h)}{m}$$

M Teschner - Deformable Modeling 28

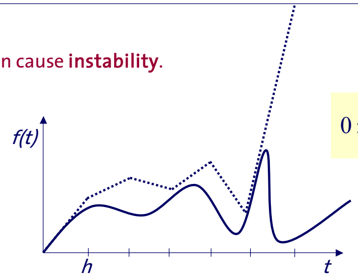
Problems

ETH Zurich

Numerical integration is **inaccurate**.

$$\underbrace{f(t+h) = f(t) + f'(t)h}_{\text{Euler step}} + \underbrace{O(h^2)}_{\text{Error}} \quad \text{(Taylor series)}$$

Inaccuracy can cause **instability**.



$$0 \le e < \frac{h^2}{2} \cdot f''(t+h)$$

Error

Avoiding Instability ?

ETH Zurich

- No general solution to avoid instability for complex mass-spring systems.
- A smaller time step increases the chance for stability.
- A larger time step speeds up the simulation.
- Parameters and topology of the mass-spring system, and external forces influence the stability of a system.

Mass-Point Dynamics - Summary

ETH Zurich

Motion equation for damped movement

$$\frac{d\mathbf{x}_i(t)}{dt} = \mathbf{v}_i(t) \qquad \frac{d\mathbf{v}_i(t)}{dt} = \frac{\mathbf{F}_i(t) - \gamma \mathbf{v}_i(t)}{m_i}$$

Initial values for velocity **v** and position **x**

Numerical integration of **v** and **x** through time

Euler method to get from time t_o to t_o+h

$$\mathbf{x}(t_0 + h) = \mathbf{x}(t_0) + h\mathbf{v}(t_0)$$

$$\mathbf{v}(t_0 + h) = \mathbf{v}(t_0) + h\frac{\mathbf{F}(t_0) - \gamma \mathbf{v}(t_0)}{m}$$

Accuracy, stability, and performance are dependent on time step *h*.
Larger time step is desired for better performance.
Smaller time step can be required for stability.

Improving the Accuracy 1 – Leap-Frog

ETH Zurich

$$\mathbf{v}(t+h/2) = \mathbf{v}(t-h/2) + h \cdot \mathbf{a}(t)$$

$$\mathbf{x}(t+h) = \mathbf{x}(t) + h \cdot \mathbf{v}(t+h/2)$$

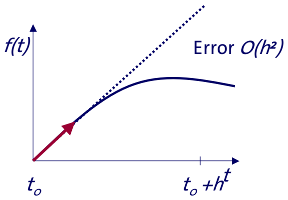
Error *O(h³)*
time step *h is* significantly larger compared to expl. Euler

Euler	Leap-Frog
	initV() // v(o) = v(o) – h/2a(o)
...	*...*
addForces(); //F(t)	*addForces(h); //F(t)*
*positionEuler(h); //x=x(t+h)=x(t)+**hv(t)***	*velocityEuler(h); //v=v(t+h)=v(t)+ha(t)*
velocityEuler(h); //v=v(t+h)=v(t)+ha(t)	*positionEuler(h); //x=x(t+h)=x(t)+**hv(t+h)***
...	*...*

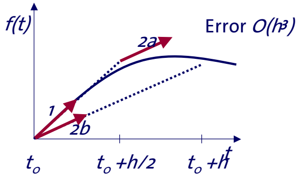
Improving the Accuracy 2

ETH Zurich

Euler Method



Runge-Kutta Method



Euler Method:

- Compute the derivative at t_o
- **Move from t_o to $t_o +h$ using the derivative at t_o**

Runge-Kutta Method:

- Compute the derivative at t_o
- Move to $t_o +h/2$
- Compute the derivative at $t_o +h/2$
- **Move from t_o to $t_o +h$ using the derivative at $t_o +h/2$**

Second-Order Runge-Kutta Method (Midpoint Method)

ETH Zurich

Carl Runge:
1856 (Bremen) – 1927 (Goettingen)
Wilhelm Kutta:
1867 (Pitschen) – 1944 (Fuerstenfeldbruck)

$$\mathbf{x}'(t) = \mathbf{v}(t) \quad \mathbf{a}(\mathbf{x}(t), \mathbf{v}(t)) = \frac{\mathbf{F}(t) - \gamma \mathbf{v}(t)}{m}$$

Compute **v** at t	$\mathbf{k}_1 = \mathbf{v}(t)$
Compute **a** at t	$\mathbf{l}_1 = \mathbf{a}(\mathbf{x}(t), \mathbf{v}(t))$
Compute **v** at $t+h/2$ (the midpoint)	$\mathbf{k}_2 = \mathbf{v}(t) + \mathbf{l}_1 \frac{h}{2}$
Compute **a** at $t+h/2$ with **x** and **v** at $t+h/2$	$\mathbf{l}_2 = \mathbf{a}\left(\mathbf{x}(t) + \mathbf{k}_1 \frac{h}{2}, \mathbf{k}_2\right)$
Compute **x** at $t+h$ with its velocity at $t+h/2$	$\mathbf{x}(t+h) = \mathbf{x}(t) + h\mathbf{k}_2$
Compute **v** at $t+h$ with the acceleration at $t+h/2$	$\mathbf{v}(t+h) = \mathbf{v}(t) + h\mathbf{l}_2$

Comparison

ETH Zurich

Euler Method	Runge-Kutta Method
• **One** computation of the derivative per time step • Error $O(h^2)$	• **Two** computations of the derivative per time step • Error $O(h^3)$ • Allows larger time steps

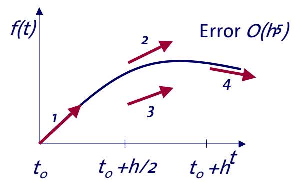
$$0 \le e < \frac{h^n}{n} \cdot f^{(n)}(x)$$

- Accuracy of Runge-Kutta per time step is higher.
- Computational complexity similar if time step for Runge-Kutta is twice the time step for Euler.
- Does Runge-Kutta really allow faster simulations of mass-spring systems?

Fourth-Order Runge-Kutta Method

ETH Zurich

- Compute the derivative at t_o (1)
- Move from t_o to $t_o +h/2$ using the derivative at t_o (1)
- Compute the derivative at $t_o +h/2$ (2)
- Move from t_o to $t_o +h/2$ using the derivative at $t_o +h/2$ (2)
- Compute the derivative at $t_o +h/2$ (3)
- Move from t_o to $t_o +h$ using the derivative at $t_o +h/2$ (3)
- Compute the derivative at $t_o +h$ (4)
- Compute a weighted average of all derivates (1) – (4) and use this value to move from t_o to $t_o +h$



Fourth-Order Runge-Kutta Method

ETH Zurich

$$\mathbf{k}_1 = \mathbf{v}(t)$$
$$\mathbf{l}_1 = \mathbf{a}(\mathbf{x}(t), \mathbf{v}(t))$$
$$\mathbf{k}_2 = \mathbf{v}(t) + \mathbf{l}_1 \frac{h}{2}$$
$$\mathbf{l}_2 = \mathbf{a}\left(\mathbf{x}(t) + \mathbf{k}_1 \frac{h}{2}, \mathbf{k}_2\right)$$
$$\mathbf{k}_3 = \mathbf{v}(t) + \mathbf{l}_2 \frac{h}{2}$$
$$\mathbf{l}_3 = \mathbf{a}\left(\mathbf{x}(t) + \mathbf{k}_2 \frac{h}{2}, \mathbf{k}_3\right)$$
$$\mathbf{k}_4 = \mathbf{v}(t) + \mathbf{l}_3 h$$
$$\mathbf{l}_4 = \mathbf{a}(\mathbf{x}(t) + \mathbf{k}_3 h, \mathbf{k}_4)$$

$$\mathbf{x}(t+h) = \mathbf{x}(t) + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$
$$\mathbf{v}(t+h) = \mathbf{v}(t) + \frac{h}{6}(\mathbf{l}_1 + 2\mathbf{l}_2 + 2\mathbf{l}_3 + \mathbf{l}_4)$$

- Four computations of the derivative per time step
- Error *$O(h^5)$*
- Allows even larger time steps

Implementation

ETH Zurich

Euler Method
- **Straightforward**
- Compute spring forces
- Add external forces
- Update positions
- Update velocities

Runge-Kutta Method
- Compute spring forces
- Add external forces
- Compute **auxiliary** positions and velocities
  - requires additional copies of data
  - once for second-order
  - three times for fourth-order
- Update positions
- Update velocities

Implicit Integration – Theta Scheme

ETH Zurich

$$\mathbf{x}(t+1) = \mathbf{x}(t) + h\big((1-\theta)\cdot\mathbf{v}(t) + \theta\cdot\mathbf{v}(t+1)\big)$$
$$m\cdot\mathbf{v}(t+1) = m\cdot\mathbf{v}(t) + h\big((1-\theta)\cdot\mathbf{F}(t) + \theta\cdot\mathbf{F}(t+1)\big)$$

$\theta = 0:$ *exlicit Euler*

$\theta = 1:$ *implicit Euler*

$\theta = 0.5:$ *Crank Nicolson*

Theta Scheme - Implementation

ETH Zurich

rewriting the problem

$$m\cdot\mathbf{v}(t+h) = m\cdot\mathbf{v}(t) + h\cdot\frac{\mathbf{F}(\mathbf{x}(t)) + \mathbf{F}(\mathbf{x}(t) + h\cdot\frac{\mathbf{v}(t)+\mathbf{v}(t+h)}{2})}{2}$$

linearization of force

$$\mathbf{F}(\mathbf{x}(t) + h\cdot\frac{\mathbf{v}(t)+\mathbf{v}(t+h)}{2}) \approx \mathbf{F}(\mathbf{x}(t)) + \left.\frac{\partial\mathbf{F}(\mathbf{x})}{\partial\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}(t)}\cdot h\cdot\frac{\mathbf{v}(t)+\mathbf{v}(t+h)}{2}$$

explicit form for **v**(*t+h*)

$$\left[m - \frac{h^2}{2}\cdot\left.\frac{\partial\mathbf{F}(\mathbf{x})}{\partial\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}(t)}\right]\cdot\mathbf{v}(t+h) \approx m\cdot\mathbf{v}(t) + h\cdot\mathbf{F}(\mathbf{x}(t)) + \frac{h^2\cdot\mathbf{v}(t)}{2}\cdot\left.\frac{\partial\mathbf{F}(\mathbf{x})}{\partial\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}(t)}$$

Theta Scheme – Conjugate Gradient

ETH Zurich

- linear system: $\mathbf{A} \cdot \mathbf{v} = \mathbf{b}$
- gradient of a function $\nabla f(\mathbf{v}) = \mathbf{A} \cdot \mathbf{v} - \mathbf{b}$

  with $\nabla f(\mathbf{v}) = 0$
- iterative solution for v with initial value v_0

  Iteration: $\mathbf{v}_{k+1} = \mathbf{v}_k + \alpha \cdot w(\nabla f(\mathbf{v}_k))$

Higher-Order Numerical Integration

ETH Zurich

Integration methods for first-order ODE's	Integration methods for Newton's motion equation
Euler Heun Runge Kutta	Verlet Velocity Verlet Beeman
commonly used in Computer Graphics applications	commonly used in molecular dynamics

Verlet Integration

ETH Zurich

$$\mathbf{x}(t+h) = \mathbf{x}(t) + h\mathbf{x}'(t) + \frac{h^2}{2}\mathbf{a}(t) + \frac{h^3}{6}\mathbf{x}'''(t) + O(h^4)$$

+

$$\mathbf{x}(t-h) = \mathbf{x}(t) - h\mathbf{x}'(t) + \frac{h^2}{2}\mathbf{a}(t) - \frac{h^3}{6}\mathbf{x}'''(t) + O(h^4)$$

=

$$\mathbf{x}(t+h) = 2\mathbf{x}(t) + \mathbf{x}(t-h) + h^2\mathbf{a}(t) + O(h^4)$$

$$v(t) = \frac{\mathbf{x}(t+h) - \mathbf{x}(t-h)}{2h} + O(h^2)$$

Other Integration Techniques

ETH Zurich

Predictor-Corrector Methods

- Predicts a value from previous derivatives

$$f(t_0+h) = f(t_0) + \frac{h}{a_1+a_2+a_3}\left(a_1 f'(t_0) + a_2 f'(t_0-h) + a_3 f'(t_0-2h)\right)$$

- Corrects the value using the derivative from the predicted value (implicit)

$$f(t_0+h) = f(t_0) + \frac{h}{b_1+b_2+b_3}\left(b_1 f'(t_0+h) + b_2 f'(t_0) + b_3 f'(t_0-h)\right)$$

Burlisch-Stoer Methods

- Polynomial function extrapolation based on midpoint method steps
- High accuracy with minimal computational effort
- Bad for non-smooth functions
- Not very promising, but has been used for mass-spring models

Numerical Integration - Summary

ETH Zurich

Motion equation for mass point
- second-order differential equation
- coupled system of first-order differential equation
- derivatives of velocity **v** and position **x** are described

Numerical integration
- known initial values at a certain time *t* for **v** and **x**
- approximative integration of **v** and **x** through time
- time step *h*

Integration techniques
- Euler, Leap-Frog
- Runge-Kutta
- Crank-Nicolson
- Verlet, velocity Verlet, Beeman
- predictor-corrector
- methods differ in accuracy and computational complexity
- size of time step h is trade-off between performance and robustness

Outline

ETH Zurich

Motivation
Model Components
- Mass Points
- Springs
- Forces

Computation of the Dynamic Behavior
- Explicit Numerical Integration
- Implicit Numerical Integration
- Higher-Order Numerical Integration

Stability and Performance Aspects
- Performance
- Time and Space Adaptive Sampling
- Damping
- Force-Deformation Relationship
- Model Topology

How to Measure Performance ?

ETH Zurich

It is difficult !

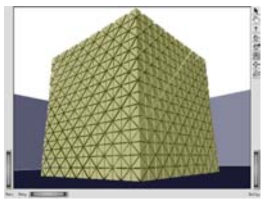
- Frames per second
  (System updates per real-life second)
  - Commonly used
  - How many mass points?
  - How many springs?
  - Computational expenses for external forces?
  - Other expenses like collision handling and rendering?
  - Which numerical integration technique?
  - What time step?

- Example: 0.1 – 1 s for 10,000 polygons per iteration using various integration techniques [Volino/Thalmann 2001] cloth simulation

Comparison of Integration Methods

ETH Zurich

- Synthetic object under gravity
- 22320 springs (16875 tetras)



Method	time step [ms]	comp. time [ms]	ratio
expl. Euler	0.5	9.3	0.05
Heun	2.9	27.5	0.1
RK2	3.8	18.9	0.2
impl. Euler	49.0	172.0	0.28
RK4	17.0	50.0	0.34
Verlet	11.5	8.5	1.35

Intel Pentium 4, 2GHz

Performance - Example

ETH Zurich

Uterus

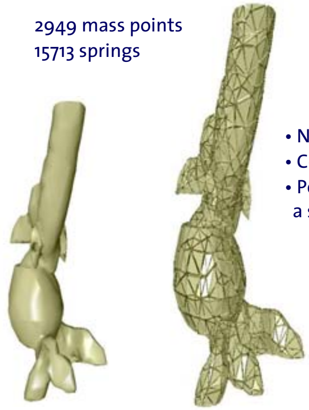
1349 mass points
6888 springs

- Numerical integration time step **5 ms**
- Computational time **1.8 ms**
- Performed tasks in a simulation environment per second
    - 450 integrations a 1.8 ms
    - 450 collision handlings a 0.2 ms
    - 23 visualizations a 1.7 ms

Performance - Example

ETH Zurich

2949 mass points
15713 springs

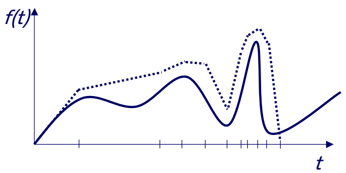


Vessel

- Numerical integration time step **1.9 ms**
- Computational time **6.1 ms**
- Performed tasks in a simulation environment per second
    - 135 integrations a 6.1 ms
    - 135 collision handlings a 1.1 ms
    - 7 visualizations a 3.7 ms

Adaptive Time Steps

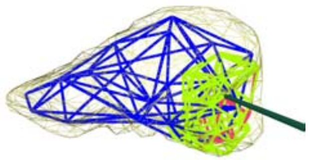
ETH Zurich



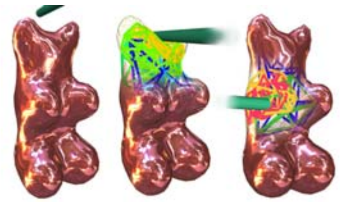
- Minimizing computational expenses by adaptively choosing time steps
- Small time steps if forces at mass points change significantly
- Larger time steps if forces do not change much
- Computational overhead for tracking force changes
- Example:
    - Free falling sphere does not require small time steps.
    - Any collision would require smaller time steps.
    - Early detection of force changes necessary.

Space Adaptive Sampling

ETH Zurich



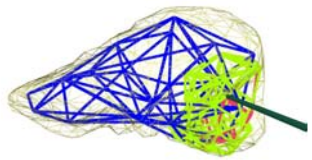
Debunne



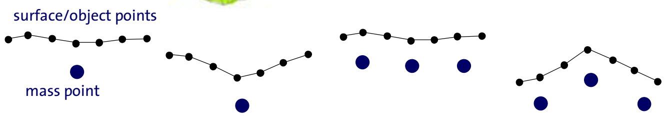
- Hierarchical model representation (coarse sampling to fine sampling)
- Different numbers of mass points for different modeling accuracies
- Allows to focus the computational load on interesting regions
- Problem: How to define regions of interest
- Problem: Relation between simulated mass points and object points

Space Adaptive Sampling

ETH Zurich



- Combination of physically-based and geometric deformation
- Different degrees of freedom for deformation



- Definition of rules to switch between level of details in certain areas
- Update of the relation between mass points and surface points.

Space Adaptive Sampling

ETH Zurich

Debunne
Desbrun
Coti
Barr

Damping

ETH Zurich

Damping is a force usually linear proportional to $\mathbf{v}^n$ in opposite direction

- Coulomb $F \sim \mathbf{v}^0$ — not applied for mass-spring systems
- Stokes $F \sim \mathbf{v}$ — uniform damping, **preferred**
- Newton $F \sim \mathbf{v}^2$ — non-uniform damping

- Models the real fact, that friction costs energy
- Smoothens variations in velocities and positions of mass points
- Reduces instability risk

It is not intended to bound velocities and positions!
Not the magnitude, but variations influence the stability!

Point Damping

ETH Zurich

- Force at a point is used for acceleration and damping

$$m_i \frac{d^2 \mathbf{x}_i(t)}{dt^2} + \gamma \frac{d\mathbf{x}_i(t)}{dt} = \mathbf{F}_i$$

damping force

- Damping force is applied in opposite direction to the velocity of the point

$$\frac{d\mathbf{v}_i(t)}{dt} = \frac{\mathbf{F}_i(t) - \gamma \mathbf{v}_i(t)}{m_i}$$

- Example: Falling mass point has a maximum speed, when gravity and damping are in an equilibrium

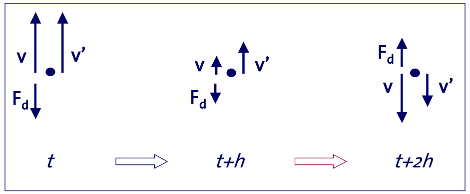
Point Damping Problem

ETH Zurich

$$\mathbf{v}(t_0+h)=\mathbf{v}(t_0)+h\frac{\mathbf{F}(t_0)-\gamma\mathbf{v}(t_0)}{m}$$

acceleration instead of damping from t+h to t+2h due to changing velocity direction

F_d - damping force
v' - velocity without damping
v - velocity with damping

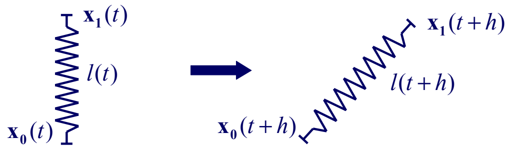


Solution: consideration of the velocity at the next time step

$$\mathbf{v}(t_0+h)=\mathbf{v}(t_0)+h\frac{\mathbf{F}(t_0)-\gamma\mathbf{v}(t_0+h)}{m}$$

Spring Damping

ETH Zurich



- Damping force proportional to the velocity of the spring

$$\mathbf{F}_1(t)=-\gamma\frac{l(t+h)-l(t)}{h}\frac{(\mathbf{x}_1(t)-\mathbf{x}_0(t))}{|\mathbf{x}_1(t)-\mathbf{x}_0(t)|}$$

- Damps relative movements of mass points
- Reduces internal oscillations of the mass-spring model
- Requires computation of auxiliary point positions
- Problem: Which force direction is correct?

Simplified Spring Damping

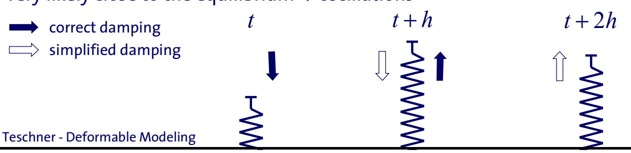
ETH Zurich

- Consider spring velocity from the previous time step to reduce computational complexity

$$\mathbf{F}_1(t)=-\gamma\frac{l(t+h)-l(t)}{h}\frac{(\mathbf{x}_1(t)-\mathbf{x}_0(t))}{|\mathbf{x}_1(t)-\mathbf{x}_0(t)|}$$

$$\mathbf{F}_1(t)=-\gamma\frac{l(t)-l(t-h)}{h}\frac{(\mathbf{x}_1(t)-\mathbf{x}_0(t))}{|\mathbf{x}_1(t)-\mathbf{x}_0(t)|}$$

- Problem (erroneous when direction of velocity changes):
- very likely close to the equilibrium -> oscillations



Damping - Summary

ETH Zurich

Spring damping has more influence on the stabilitiy of the numerical integration process than point damping.

It is essential to damp relative movement of mass points with respect to each other.

Euler integration does not provide information about changing directions of velocities within the next time step. Erroneous damping can happen!

Erroneous damping is problematic close to equlibriums and in case of large spring constants.

Integration techniques which use predictions of the next time step have the potential to detect changing directions of velocities. Erroneous damping can be avoided.

Force-Deformation Relationships

ETH Zurich

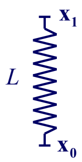
- Definition of forces which work against deformation
- No deformation -> no force

$$\mathbf{F}_0^{\text{int}} = k\left(|\mathbf{x}_1 - \mathbf{x}_0| - L\right)\frac{\mathbf{x}_1 - \mathbf{x}_0}{|\mathbf{x}_1 - \mathbf{x}_0|}$$

$$\mathbf{F}_0^{\text{int}} = k\left(|\mathbf{x}_1 - \mathbf{x}_0|^2 - L^2\right)\frac{\mathbf{x}_1 - \mathbf{x}_0}{|\mathbf{x}_1 - \mathbf{x}_0|^2}$$

$$\mathbf{F}_0^{\text{int}} = k\left(\frac{|\mathbf{x}_1 - \mathbf{x}_0| - L}{L}\right)^n\frac{\mathbf{x}_1 - \mathbf{x}_0}{|\mathbf{x}_1 - \mathbf{x}_0|}$$

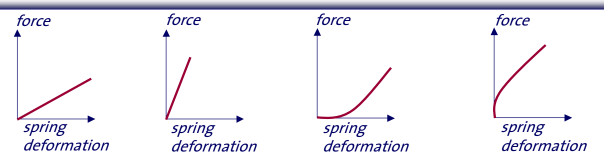
n odd ! Why ?



- Various computational complexity
- Influence on the function, which is integrated

Spring Properties and Stability

ETH Zurich



soft	stiff	oscillating	non-oscillating
stable	problematic	stable	don't even think of using it

- Behavior of force-deformation relation strongly influences stability.
- It's difficult to simulate stiff structures.
- It's difficult to represent non-oscillating deformable structures.
- Human soft tissue
  - Large forces should be handled.
  - No oscillations

Model Properties and Stability

ETH Zurich

- Two different sets of parameters (mass and stiffness) for the same topology
- Higher stiffness difficult to handle

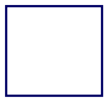
Teschner

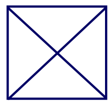
Topology and Stability –2D

ETH Zurich

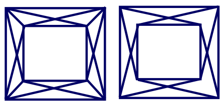
- Stable model topologies with respect to deformation

not stable





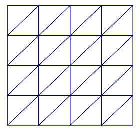
stable but not general



stable

can be generated automatically by copying the surface to an inner layer and connecting both – **layered model**

- Design problem

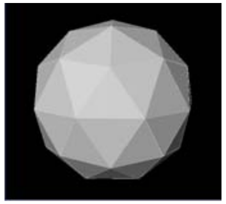
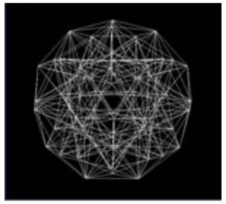


much more resistant in ↗ direction than in ↘ direction.

Topology and Stability –3D

ETH Zurich

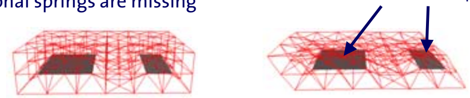
- Triangulated sphere surface
- Edges represented by springs
- Stable sphere model represented with two layers of springs
- Inner layer is a copy of the surface
- Both layers are connected by springs
- Stable, but many springs, many forces, computational expensive

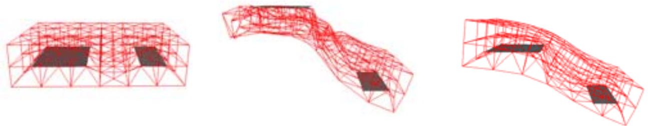
Topology - Ambiguity Problem

ETH Zurich

- Unappropriate topology
- No force penalty for shearing
- Diagonal springs are missing



- Appropriate topology
- However, self-collision problem

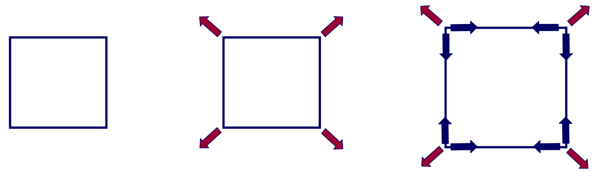


original equilibrium 1 equilibrium 2

Topology and Stability –2D

ETH Zurich

- Generating **pre-strained** models for stability



relaxed, not stable — additional forces — new stable? equilibrium

Problems:

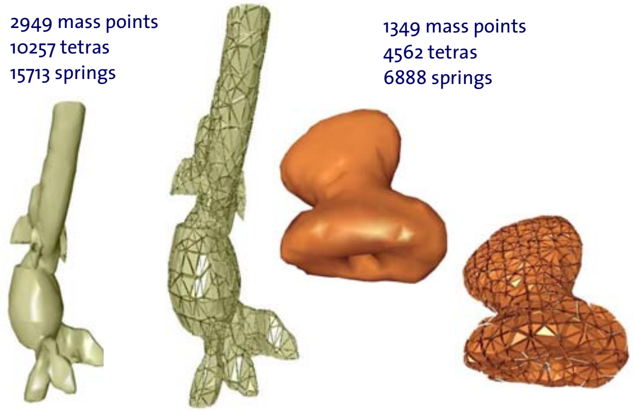
- New equilibrium does not represent given model geometry.
- Initiated forces are dependent on surface changes and object transformation.
- Initiated forces could be bound to surface normals.

Volumetric Models

ETH Zurich

2949 mass points
10257 tetras
15713 springs

1349 mass points
4562 tetras
6888 springs



Performance Aspects - Summary

ETH Zurich

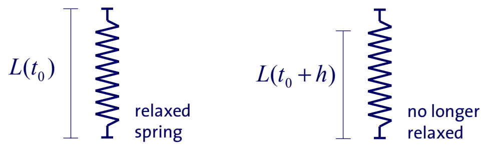
- Performance is difficult to assess!
- Varying time steps
- Model hierarchies, coarse to fine point sampling
- Point and spring damping.
- Force-deformation relationship. Soft and oscillating models are robust.
- Topology. How to connect mass points in an appropriate way?
  Influence on the stability of the numerical integration
- Topology. How to avoid too many springs?
  Influence on the performance of the numerical integration

Other Forces

ETH Zurich

- Induction of internal forces by falsifying the initial spring length

$$L(t) = const \longrightarrow L(t) = f(t)$$



- Simulation of muscle contraction (or fishes)
- *L(t)* has to be smooth for stability

Examples for External Forces

ETH Zurich

- Gravity
- Wind

Mass points can be part
of a mass-spring system, but
their velocities and positions
are not necessarily updated.

Movie

Movie

Teschner

Penalty Forces

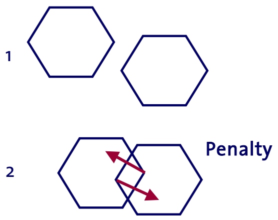
ETH Zurich

- Penalty forces are additional momentum conserving external forces at mass points.
- Penetration of a mass point into another object can cause a penalty force to resolve the collision.
- Volume variation of basic volumetric elements can cause forces to preserve volume.
- Surface changes can cause penalty forces at mass points to preserve certain characteristics of the surface.
- Proper relation of penalty, internal, and other external forces has to be considered.
- Penalty force functions should be appropriate for numerical integration.

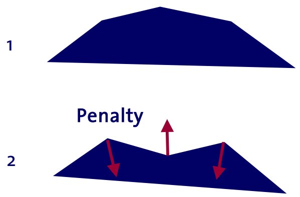
Penalty Forces

ETH Zurich

Collision



Surface changes



Dynamic Deformation - Summary

ETH Zurich

- **Discretization** of an object into mass points
- Definition of the **connectivity** (topology, adjacencies of mass points, layers)
- **Model parameters**:
  - **Points**: masses, initial positions, initial velocities
  - **Springs**: force-deformation relationship and its parameters (stiffness)
  - **Damping**: points and/or springs (Stokes and/or Newton)
  - **External and penalty forces**: definition of theses forces over time
  - **Restrictions**: fixed points, restricted movement

Dynamic Deformation - Summary

ETH Zurich

- **Numerical integration technique:**
  - Euler, Leap-Frog
  - Runge Kutta
  - Crank-Nicolson
  - Verlet, Beeman
  - Predictor-corrector methods
- **Consider performance aspects!**
- **Find an appropriate time step to solve your specific problem !**

Outline

ETH Zurich

Motivation
Model Components
- Mass Points
- Springs
- Forces

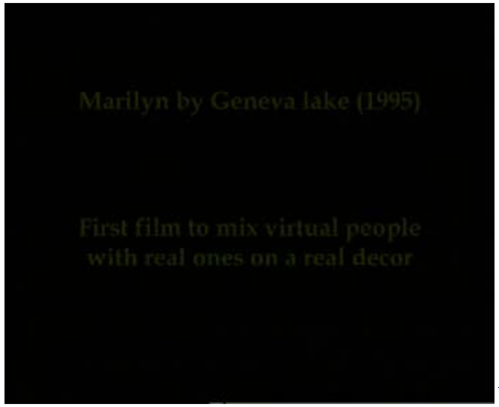
Computation of the Dynamic Behavior
- Explicit Numerical Integration
- Implicit Numerical Integration
- Higher-Order Numerical Integration

Stability and Performance Aspects
- Performance
- Time and Space Adaptive Sampling
- Damping
- Force-Deformation Relationship
- Model Topology

Movies

Dynamic Deformation – Cloth

ETH Zurich



Thalmann

Dynamic Deformation - Cloth

ETH Zurich

- Animated body (CuriousLabs' Poser)
- Deformable cloth modeling

Strasser

Dynamic Deformation – Cloth

ETH Zurich

- SGI's OpenInventor
- Deformation and collision handling in **realtime**!

Meyer
Desbrun
Barr

Surgical Simulation

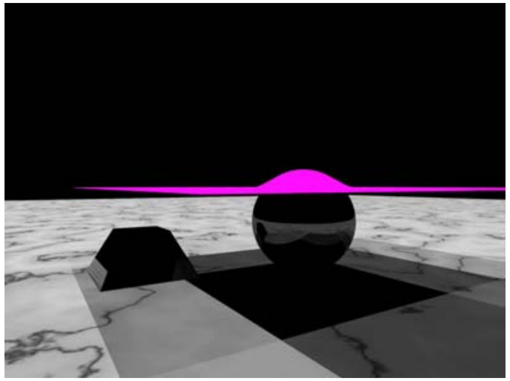
ETH Zurich

- 3D deformable structure
- deformation + collision handling in **realtime**!

Bielser
Gross

Cloth Simulation

ETH Zurich



Fedkiw
Stanford University

Hand Simulation

ETH Zurich

Articulated bone structure

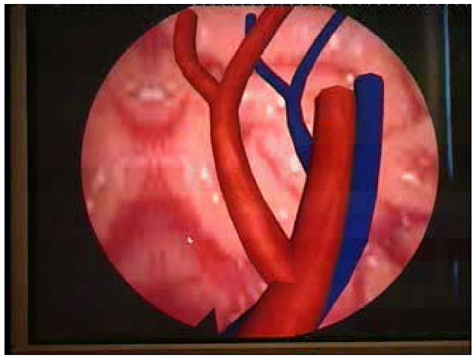
Sibille / Teschner / Latombe

Generic skin model

Deformable pulp space

Arterial Pulse Simulation

ETH Zurich



Kuehnapfel

Synthetic Faces

ETH Zurich

- Tracking of real faces, parameter estimation
- Animation of deformable, textured 3D models

Eisert / Girod

Fish Life

computer graphics lab
ETH Zurich

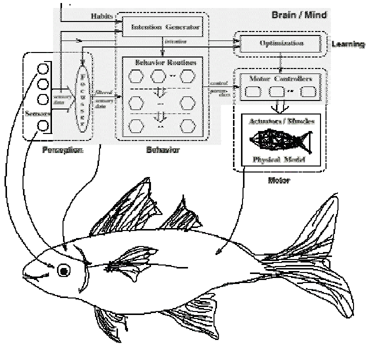


Figure 1: Control and information flow in artificial fish.

Terzopoulos

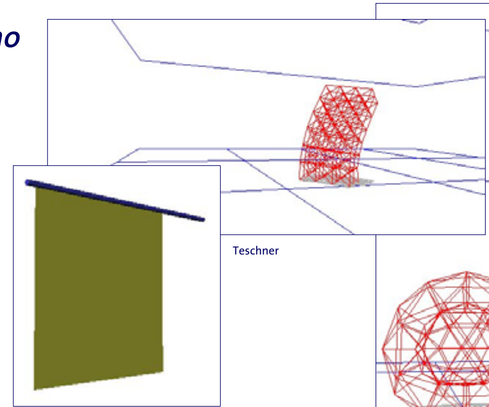
A few References

computer graphics lab
ETH Zurich

- D. Terzopoulos, J. Platt, A. Barr, K. Fleischer, “Elastically Deformable Models,” *SIGGRAPH’87, ACM Computer Graphics*, vol. 21, no. 4, pp. 205-214, 1987.
- D. Terzopoulos, K. Fleischer, “Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture,” *SIGGRAPH’88, ACM Computer Graphics*, vol. 22, no. 4, pp. 269-278, 1988.
- Y. Lee, D. Terzopoulos, K. Waters, “Realistic Modeling for Facial Animation,” *SIGGRAPH’95, ACM Computer Graphics*, vol. 29, no. 4, pp. 52-62, 1995.
- D. Baraff, A. Witkin, “Large Steps in Cloth Simulation,” *SIGGRAPH’98, ACM Computer Graphics*, pp. 43-54, 1998.
- M. Desbrun, P. Schroeder, A. Barr, “Interactive Animation of Structured Deformable Objects,” *Graphics Interface ’99,* Kingston, Canada, 1999.
- G. Debunne, M. Desbrun, M. Cani, A. Barr, “Dynamic Real-Time Deformations using Space and Time Adaptive Sampling,” *SIGGRAPHC’01, ACM Computer Graphics*, pp. 31-36, 2001.
- H. R. Schwarz, “Numerische Mathematik,” B. G. Teubner Stuttgart, ISBN 3-519-32960-3, 1997.

computer graphics lab
ETH Zurich

Demo



Teschner