

CS 277 - Experimental Haptics  
Lecture 9

# Deformable Models (I)



# Outline

- Introduction
- Spring Models
- Computing Dynamics in the Haptics Loop
- Filling Sphere Approach for Elastic Models
- Computing Collision Detection in Real Time
- Demonstrations

# Outline

- **Introduction**
- Spring Models
- Computing Dynamics in the Haptics Loop
- Filling Sphere Approach for Elastic Models
- Computing Collision Detection in Real Time
- Demonstrations

# overview

## simulators

### running offline

- CAD design tools
- movie rendering
- building a look-up-table
- games
- model tuning and calibration

### running real-time

- interactive games
- flight simulator
- haptic simulator
- real robot controller
- weather forecast

- haptics
- robot control

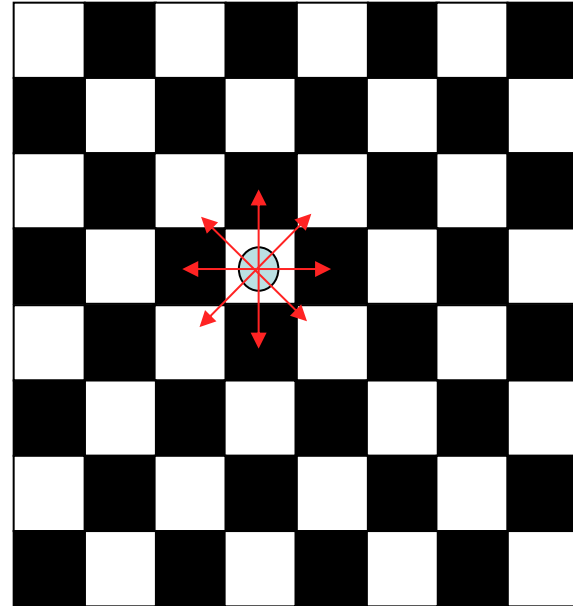
- graphic animation
- games

- CAD tools
- apps

- industrial simulations
- movie renderings

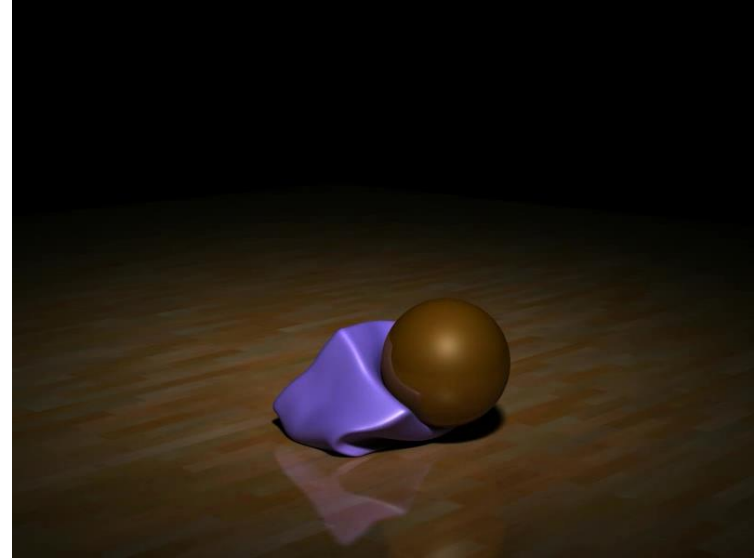


# Discrete Events



In discrete-event simulation, the operation of a system is represented as a chronological sequence of events. Each event occurs at an instant in time and marks a change of state in the system.

# Haptic Simulation



- To make objects look, behave and feel realistic when forces are applied.
- To provide visual and force feedback to the user in real time.

# Haptic Simulation

- **Input**
  - Object model (non-deformed geometry)
  - Forces
    - Over whole volume (e.g., gravity)
    - Over the surface (e.g., pressure, drag)
    - Concentrated loads (e.g., poking with haptic device)
- **Output:**
  - New, deformed geometry
    - Static equilibrium
    - At each time step (dynamic)
  - If using meshes, just need node displacements
  - Usually assumes invariant topology (e.g., no cutting)

# Haptic Simulation

- **Input**

- Object model (non-deformed geometry)
- Forces
  - Over whole volume (e.g., gravity)
  - Over the surface (e.g., pressure, drag)
  - Concentrated loads (e.g., poking with haptic device)

- **Output:**

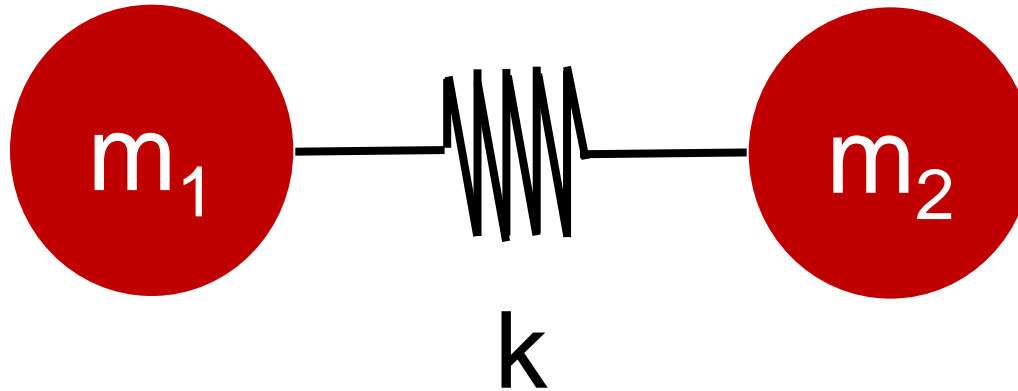
- New, deformed geometry
  - Static equilibrium
  - At each time step (dynamic)
- If using meshes, just need node displacements
- Usually assumes invariant topology (e.g., no cutting)



# Outline

- Introduction
- **Spring Models**
- Computing Dynamics in the Haptics Loop
- Filling Sphere Approach for Elastic Models
- Computing Collision Detection in Real Time
- Demonstrations

# Spring Models



- Discretize the object into a collection of  $N$  nodes interconnected with springs
- For each node  $i$ , just using Newton's  $F=ma$  law, with some velocity-dependent damping,

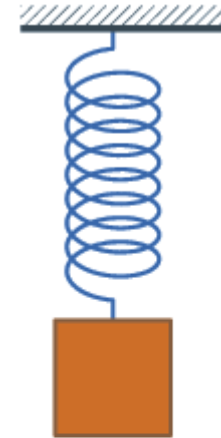
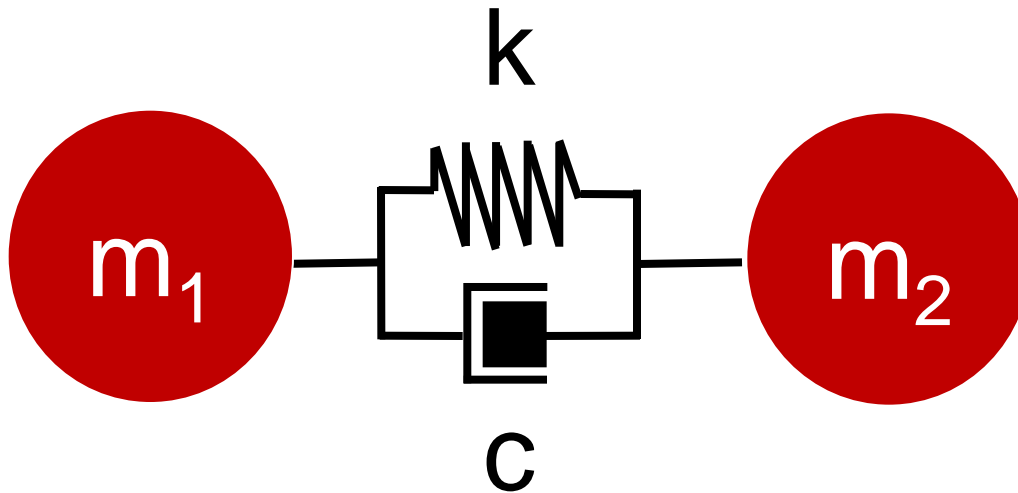
$$m_i \ddot{\mathbf{x}}_i = -\gamma_i \dot{\mathbf{x}}_i + \sum_j \mathbf{g}_{ij} + \mathbf{f}_i$$

**damping** (points to  $-\gamma_i \dot{\mathbf{x}}_i$ )

**gravity** (points to  $\mathbf{g}_{ij}$ )

**external forces (i.e. springs)** (points to  $\mathbf{f}_i$ )

# Spring Models



$$F_d = -cv = -c\dot{x} = -c\frac{dx}{dt}$$

$$F_s = -kx$$

# Implementation

(1) For each mass node, compute all external forces

- springs interconnecting mass nodes
- reaction forces between colliding mass nodes
- gravitational forces

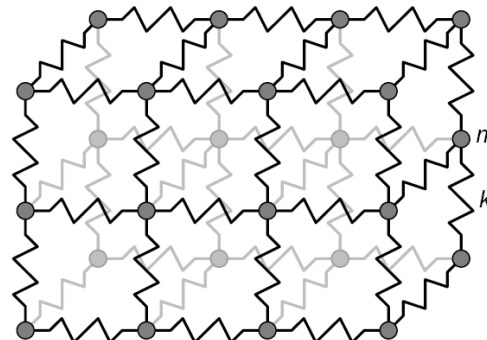
(2) Compute Acceleration

$$a = F / m$$

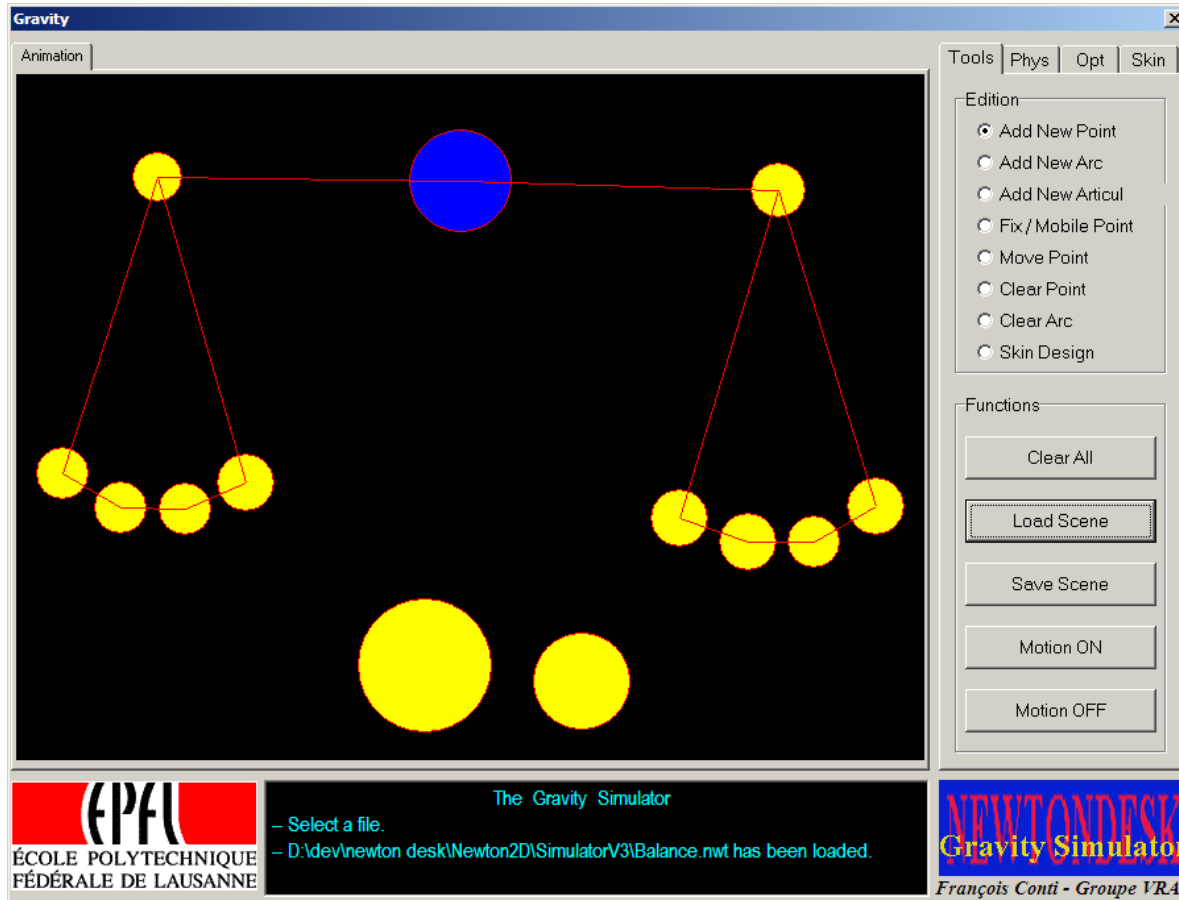
(3) Update Velocity and Position through integration over time **dt**

$$v = v_{\text{prev}} + a \, dt$$

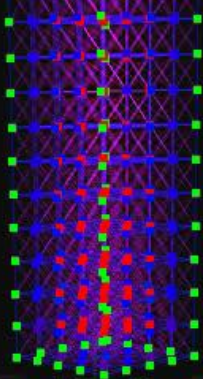
$$x = x_{\text{prev}} + v \, dt + \frac{1}{2} a \, dt^2$$



# Demonstration



# Haptic Simulation



Behavior Model

**:: SOFA ::**  
Smoothed Particle  
Hydrodynamics (SPH)  
Fluid Simulation

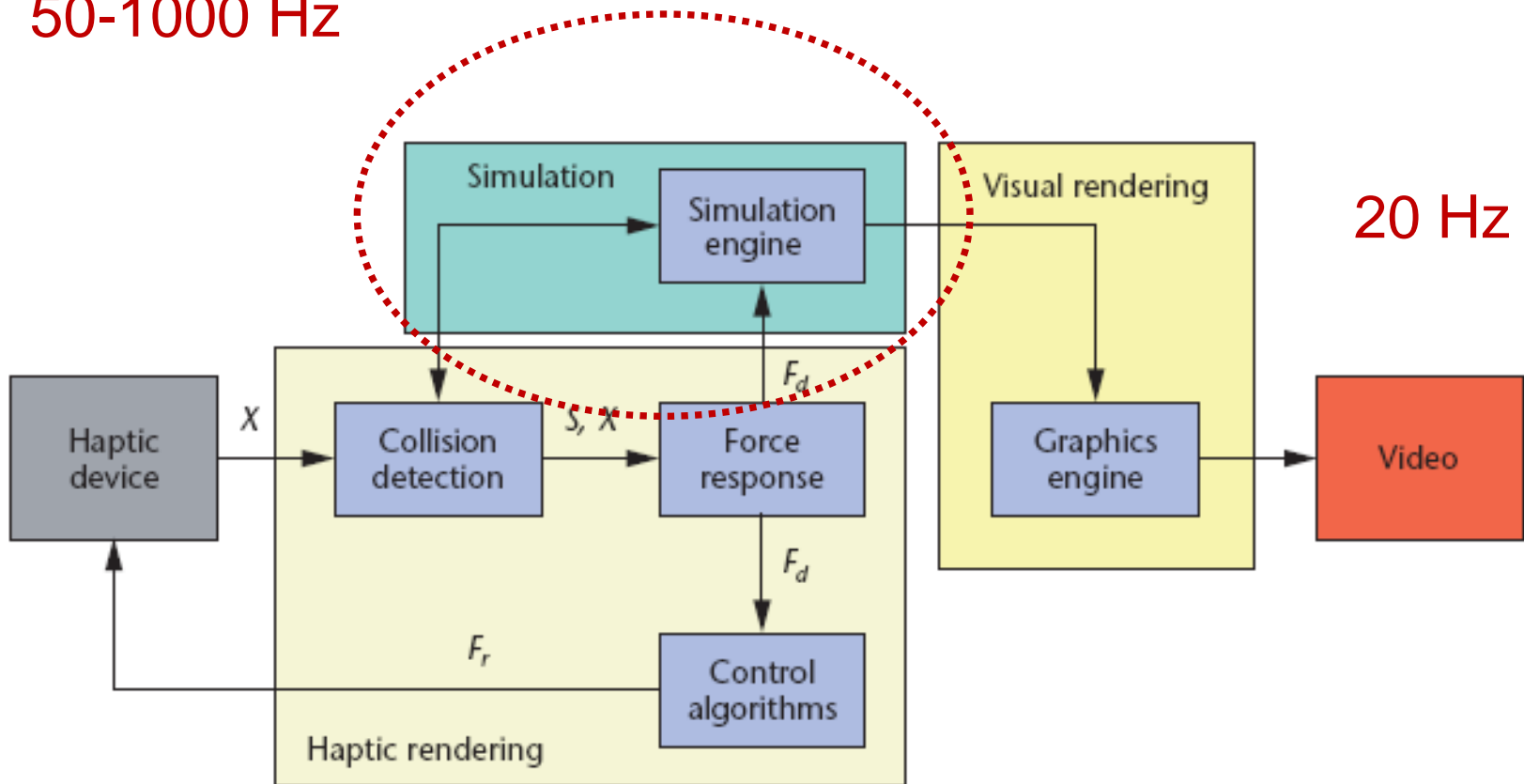
The image shows a 3D visualization of a haptic simulation. It features a grid of particles, with some particles highlighted in blue, green, and red. The text 'Behavior Model' is positioned to the right of the grid. Below the grid, the text 'SOFA' is displayed in a large, stylized font. Underneath 'SOFA', the text 'Smoothed Particle Hydrodynamics (SPH) Fluid Simulation' is written in a smaller font.

# Outline

- Introduction
- Spring Models
- **Computing Dynamics in the Haptics Loop**
- Filling Sphere Approach for Elastic Models
- Computing Collision Detection in Real Time
- Demonstrations

# Computing Dynamics

50-1000 Hz

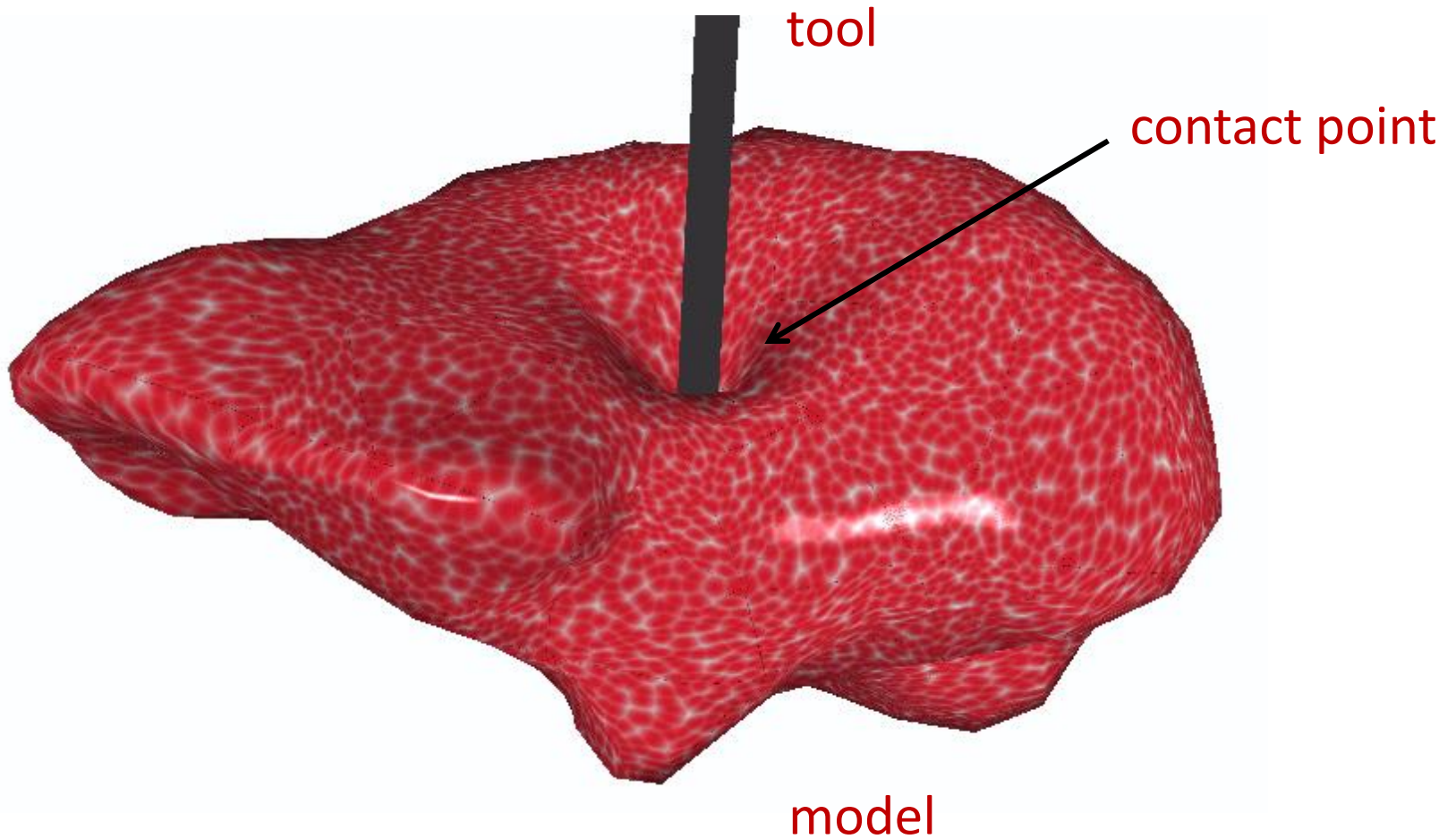


20 Hz

1000 Hz



# Deformable Mesh



# Outline

- Introduction
- Spring Models
- Computing Dynamics in the Haptics Loop
- **Filling Sphere Approach for Elastic Models**
- Computing Collision Detection in Real Time
- Demonstrations

# Filling Sphere Approach (2D)



**Gravity**

Animation

Tools | Phys | Opt | Skin

Edition

- Add New Point
- Add New Arc
- Add New Articul
- Fix / Mobile Point
- Move Point
- Clear Point
- Clear Arc
- Skin Design

Functions

Clear All

Load Scene

Save Scene

Motion ON

Motion OFF

**EPFL**  
ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE

- Move the selected point (2) with mouse.
- Move the selected point (0) with mouse.
- Move the selected point (0) with mouse.

**NEWTONDESK**  
Gravity Simulator  
François Conti - Groupe VRAI

# Outline

- Introduction
- Spring Models
- Computing Dynamics in the Haptics Loop
- Filling Sphere Approach for Elastic Models
- **Computing Collision Detection in Real Time**
- Demonstrations

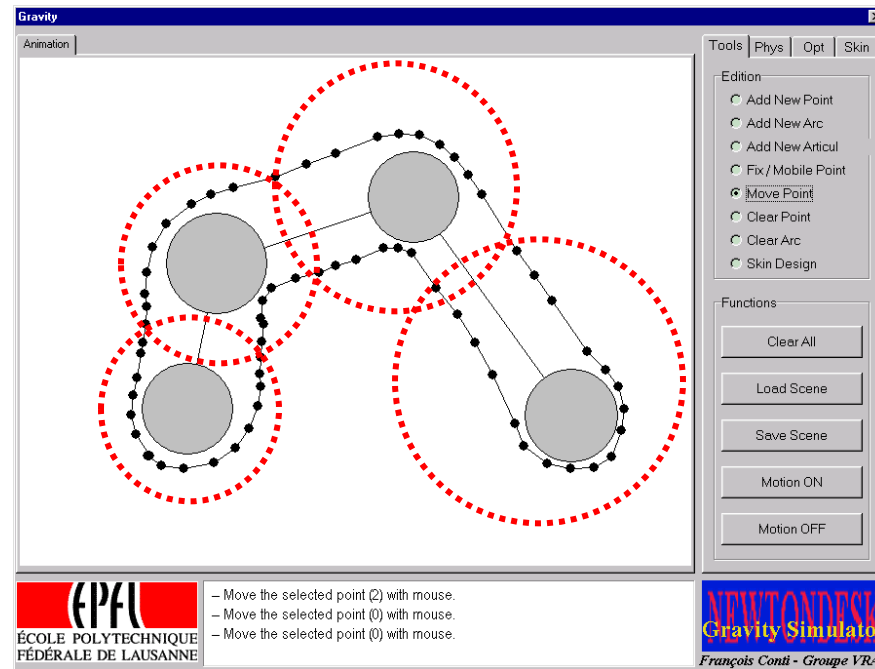
# Collision Detection

Collision detection with **deformable meshes** are difficult to achieve in real time due to the constant change of their geometry (constant update of the collision detection model)

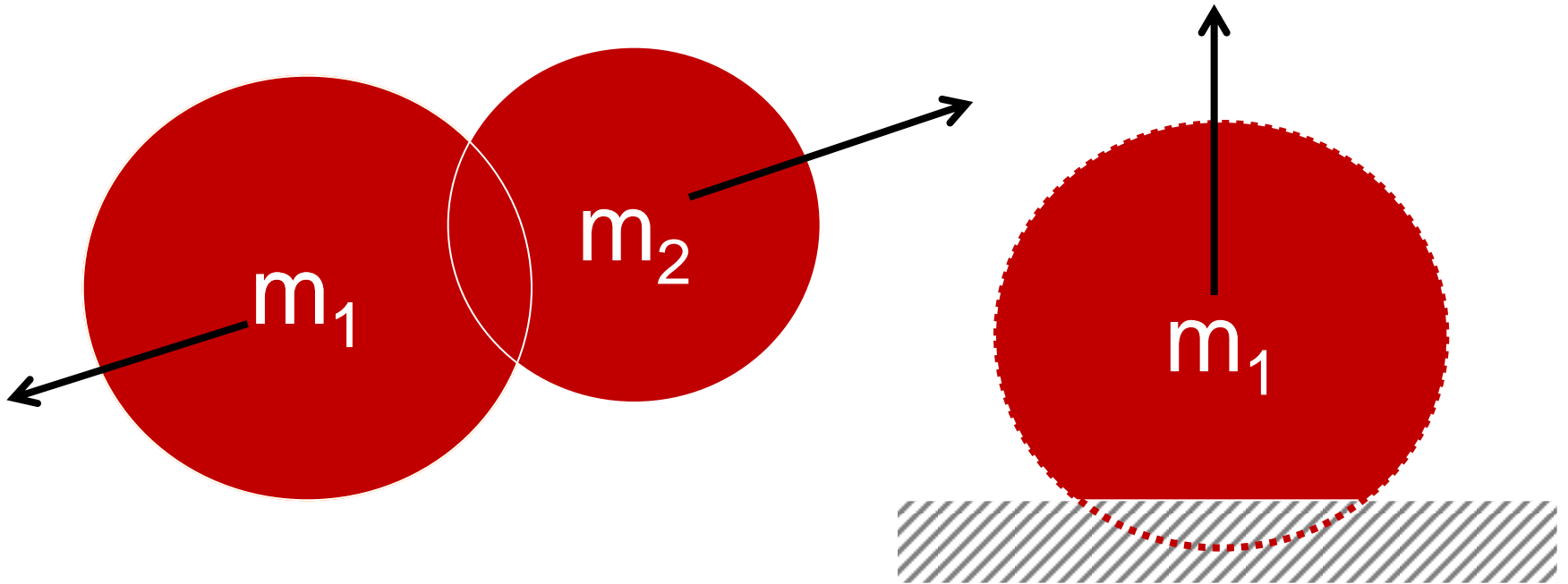
- How can we compute a collision between a segment and deformable mesh?
- How can we compute collisions between deformable meshes?

# Collision Segment-Mesh

1. Collision detection is first performed between the input segment and the **collision spheres** composing the skeleton of the model.
2. Collision between the segment and the triangles are then searched locally



# Collision Mesh-Mesh



Reaction forces are computed between mass nodes

$$F_r = -k x$$

# Outline

- Introduction
- Spring Models
- Computing Dynamics in the Haptics Loop
- Filling Sphere Approach for Elastic Models
- Computing Collision Detection in Real Time
- **Demonstrations**