Introduction to Computer Graphics

Marie-Paule Cani & Estelle Duveau

04/02 Introduction & projective rendering
11/02 Prodedural modeling, Interactive modeling with parametric surfaces

25/02 Introduction to OpenGL + lab: first steps & modeling
04/03 Implicit surfaces 1 + lecture/lab: transformations & hierarchies
11/03 Implicit surfaces 2 + Lights & materials in OpenGL
18/03 Textures, aliasing + Lab: Lights & materials in OpenGL
25/03 Textures in OpenGL: lecture + lab
01/04 Procedural & kinematic animation + lab: procedural anim
08/04 **Physically-based animation: particle systems** + lab: physics 1
22/04 Physically-based animation: collisions, control + lab: physics 2
29/04 Animating complex objects + Realistic rendering

06/05 Talks: results of cases studies

Descriptive vs Procedural animation

Descriptive animation

Describes a single motion

Ex: interpolating key-frames, direct and inverse kinematics

Procedural animation

Generates a familly of motion and deformations from

- Initial conditions
- Laws of motion
- Interaction (from other objects or the user)

Ex: physically-based models

Physically-based models

Laws of motion from mecanics

- Model + intial conditions + applied forces $\rightarrow$ motion

Advantage: a help for realism!

- useful when dynamics plays an important part
- easier for passive models!

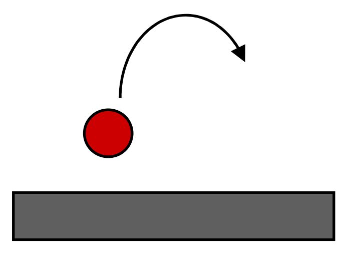
Examples :

- *Toy-Story 1, Shreck*

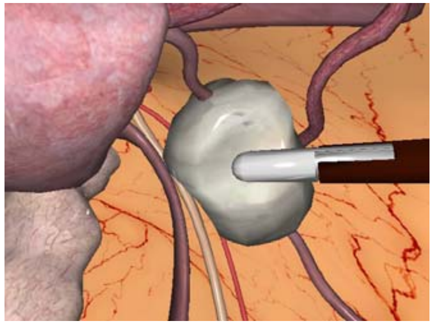
Physically-based models

Animation algorithm

- At each time step
  - for each object
    1. Compute new speed (use its law of motion)
    2. Compute new position
    3. Detect collisions – *not so simple!*
    4. Update geometry
    5. Update applied forces



Physically-based models
Which law do we need?

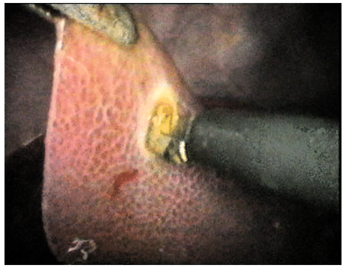


Physically-based models
Which law do we need?

- Physically-based particles
  - Gravels, sand, …
- Solids
- Articulated solids

Examples:

- Rolling ball?
- Lamps?
- Wire?




Physically-based models
Which law do we need?

Structured deformations

- Elasticity
  - Deformation function of constraints
  - Back to equilibrium
- Visco-elasticity
  - Speed of deformation
- Fractures

Ex : ball, flag, organ

Un-structured

- Neighbors change!

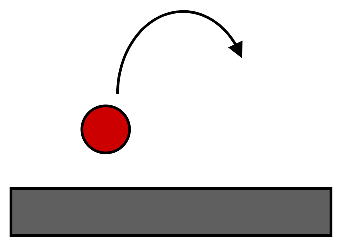
- Plasticity
  - Absorbs deformations
- Fluids
  - Navier-Stockes

Ex : modeling clay, liquids, smoke and clouds...

Some physical laws

Points

- Model [m, X, V]
- Law: $F = \sum forces = m\,a$

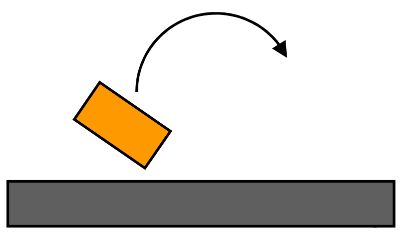


Solids

- Model [m, I inertia matrix, X, V, *rotation* ω, $\dot{\omega}$]
- Laws: $\sum F = m\,a$

$$\sum M = I\,\dot{\omega} + \omega \wedge I\,\omega$$

Difficulty: representation of orientations!

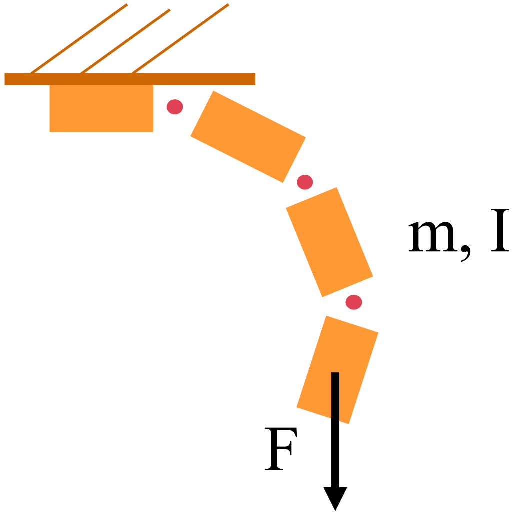


Some physical laws

Articulated solids

- Solid dynamics + constraints at joints!

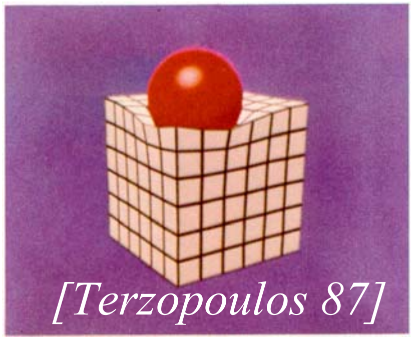
  (Lagrange multipliers..)



Deformable models

- Linear vs non-linear elasticity
- Navier stokes for fluids

  Eulerian vs Lagrangian discretization


[Terzopoulos 87]

Example: Visco-elastic models

- Cauchy : linear deformation law

- OK for small displacements
- but rotations produce forces!
- the object inflates!!

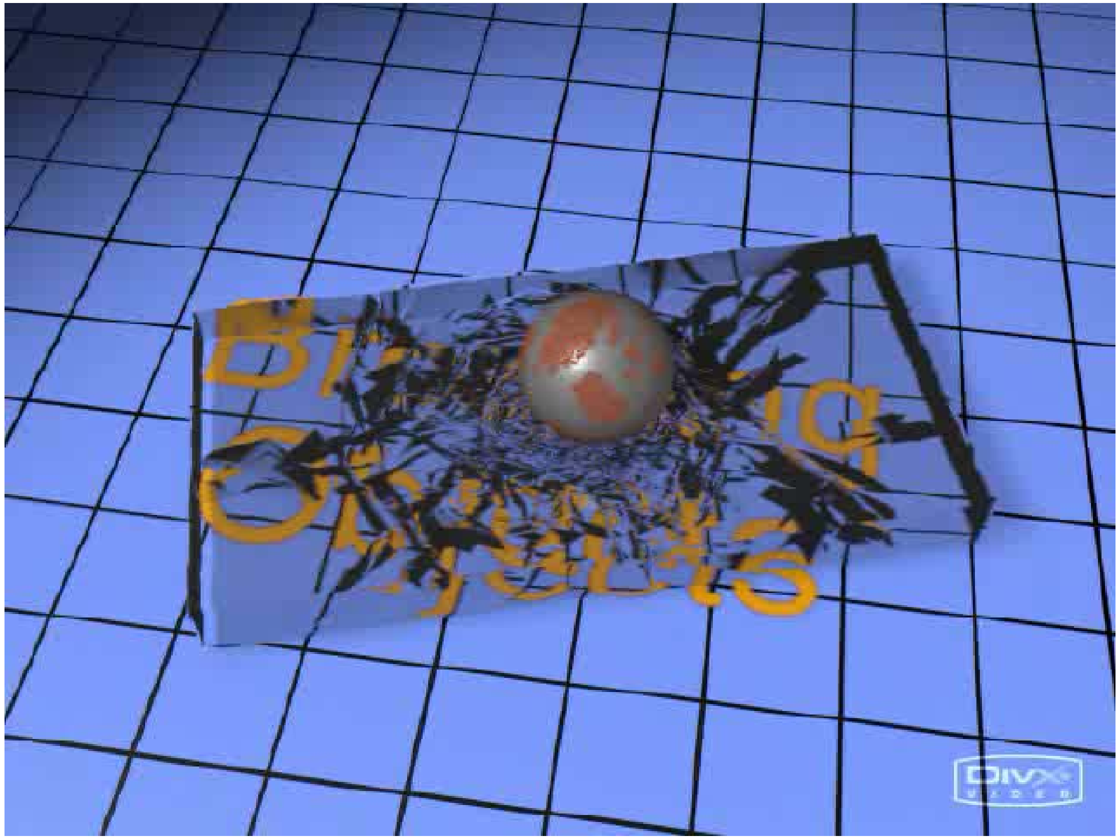
Apply Cauchy in local frames!

[Müller et al. 02, 04]

- similar to Green's non-linear tensor
- real-time

Example: Animating fratures

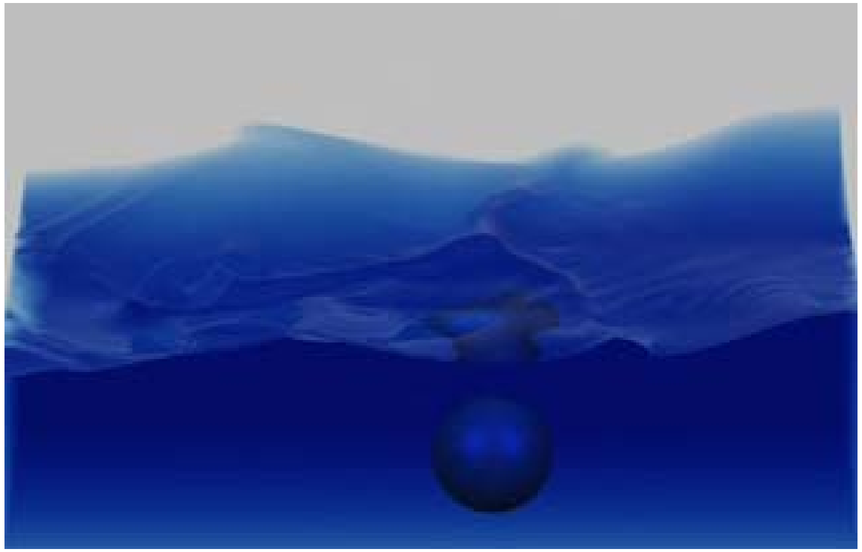
[James O'Brien]



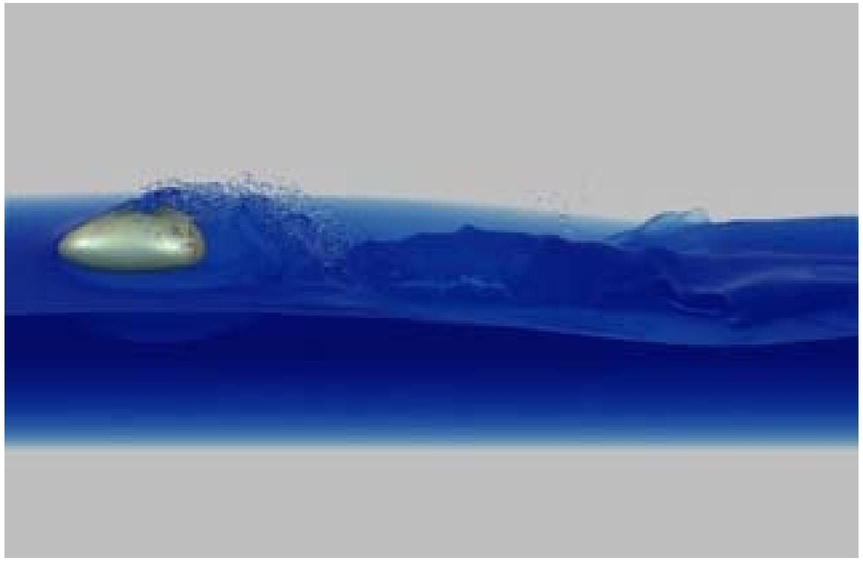
Example: liquids

Navier stockes +Eulerian grid +"level set" (implicit)

[Foster & Fedkiw 2001]



[Enright et al. 2002]



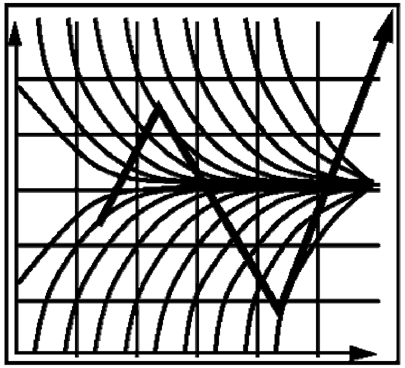
Be careful with integration!

$$\ddot{\mathbf{x}} = F(\dot{\mathbf{x}}, \mathbf{x}, t) = \mathbf{f} / m$$
$$\dot{\mathbf{x}} = \mathbf{v}$$

$$\dot{\mathbf{v}} = F(\mathbf{v}, \mathbf{x}, t)$$

- Explicit Euler

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \Delta t \, F(\mathbf{v}(t), \mathbf{x}(t), t)$$
$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \, \mathbf{v}(t)$$

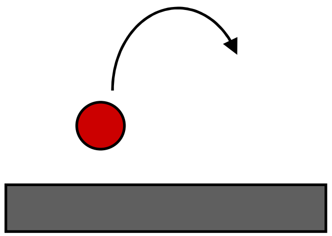


- Implicit Euler (much more stable!)

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \Delta t \, F(\mathbf{v}(t + \Delta t), \mathbf{x}(t + \Delta t), t)$$
$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \, \mathbf{v}(t + \Delta t)$$

Do it all with point-based physics?

- Model: Particles *[m, X, V]*
- Point-based physics $\sum$ *forces = m a*



Animation algorithm

– At each time step, for each particle

$$V(t+dt) = V(t) + \sum F(t)/m \;\; dt$$

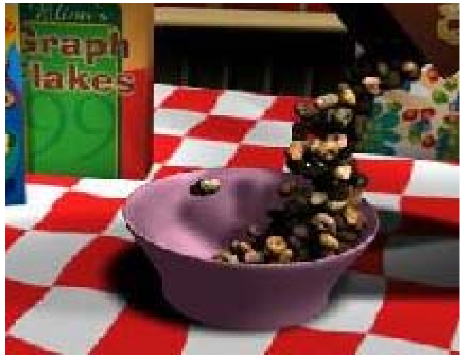
$$X(t+dt) = xX(t) + V(t)\, dt \quad \text{(if explicit Euler)}$$

- *Adapted forces ?*
- *Render with adapted geometry!*

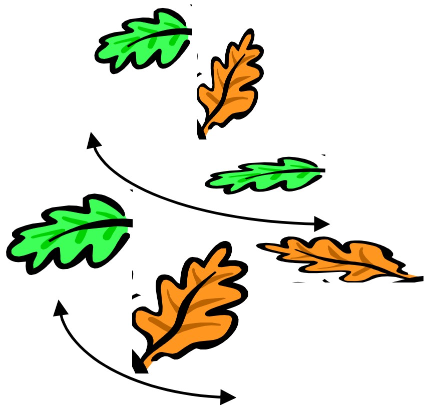
Do it all with point-based physics?
Lots of simple objects

"Physically-based particle systems"

- Example : gravels, cereals
  - Gravity
  - Spheres for collisions detection
  - Random individual geometry

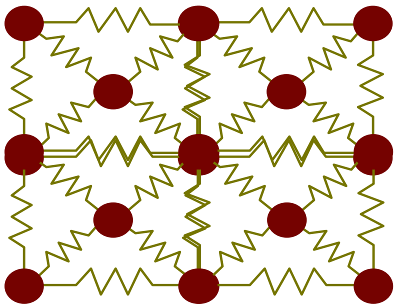


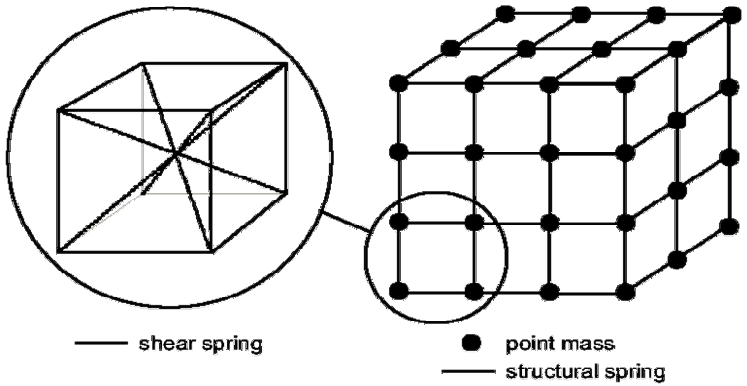
- Example : animating automn
  - Leaves = particle + local frame
  - Wind primitives
  - Gravity
  - Directional friction force



Do it all with point-based physics? Stuctured material

- 1D, 2D, 3D mass-spring networks





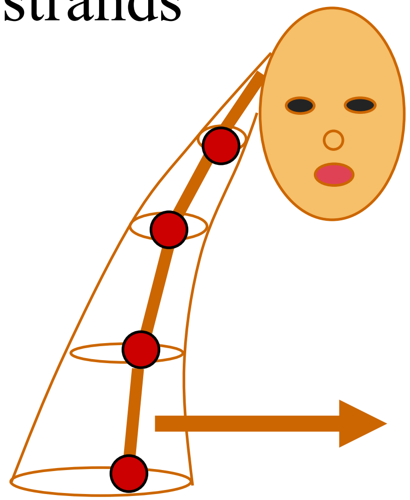
- Rigid and articulated solids?
  – Doable but very stiff! Too small time steps!

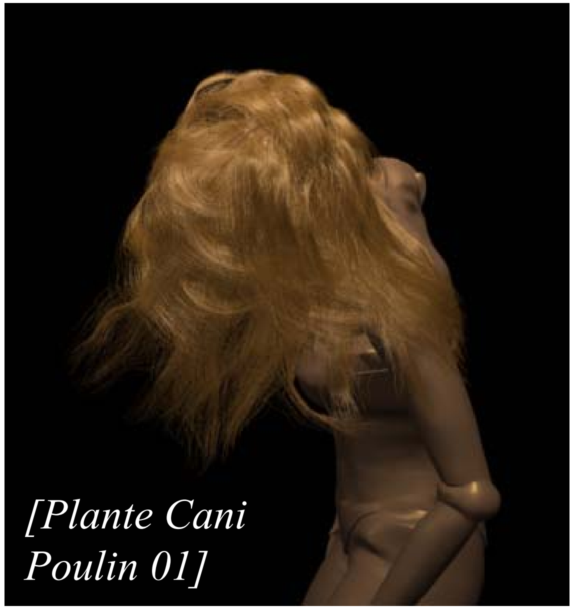
Do it all with point-based physics?
Stuctured objects

Example 1D case : hair animation

- Dynamics, inertia : wisp skeleton = chain of masses and springs
- Deformable wisp: add radial springs
- Anisotropic collisions
- Render individual hair strands

[Anjyo,1992]



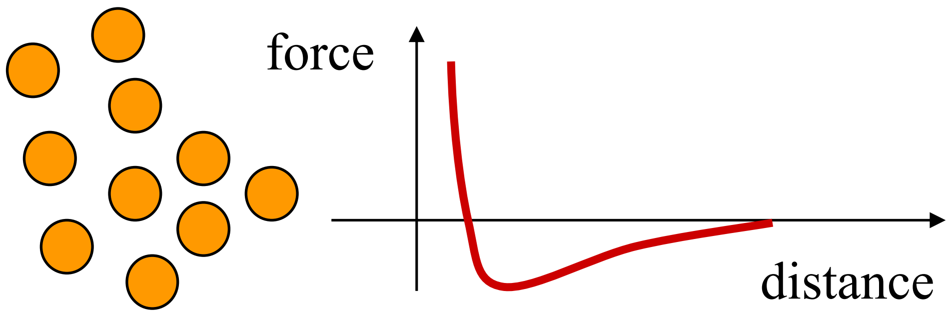


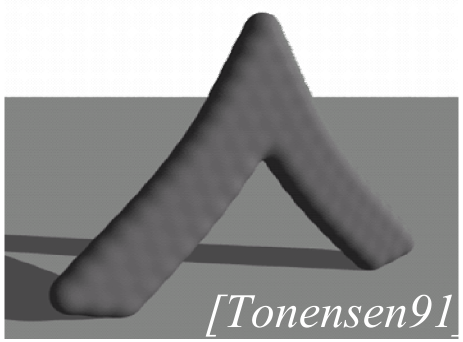
[Plante Cani Poulin 01]

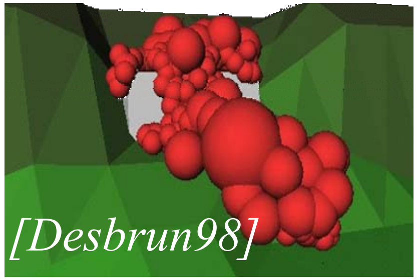
Do it all with point-based physics?
Stuctured objects

Do it all with point-based physics?
Unstructured objects

- Particle systems inspired from molecular dynamics
  - Lennard-Jones attraction/repulsion forces





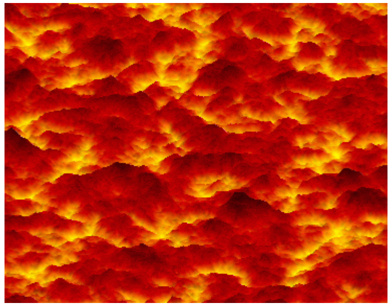


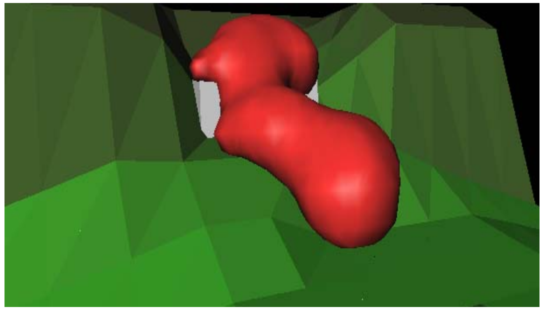
Do it all with point-based physics? Unstructured objects

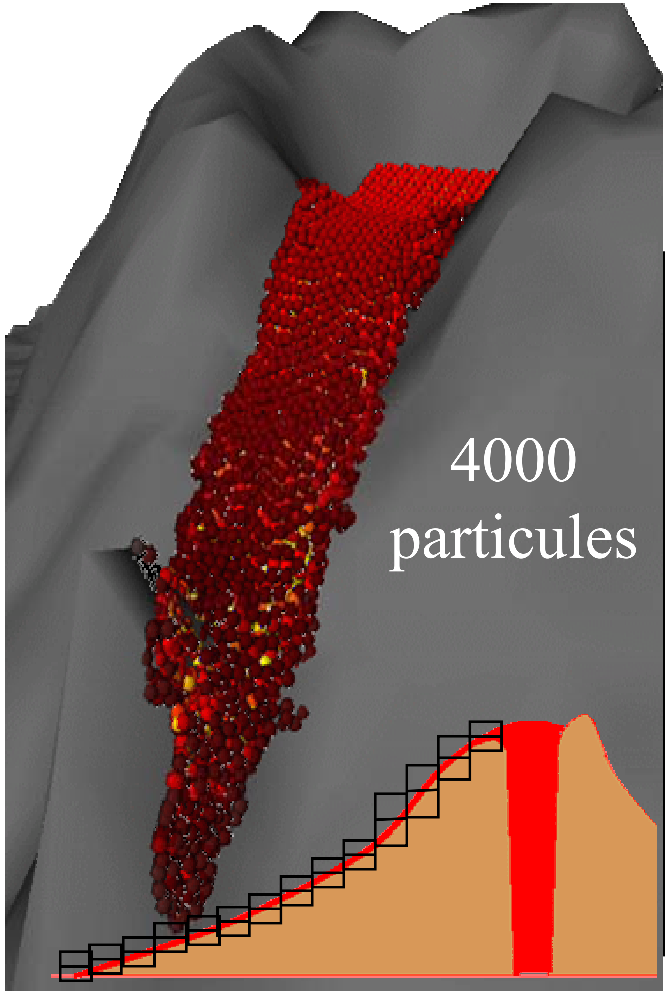
Example: Lava flow

- SPH particles + temperature
- Heat-diffusion equation
- Render with an implicit surface
- Generate crust texture

  Animated Perlin noise

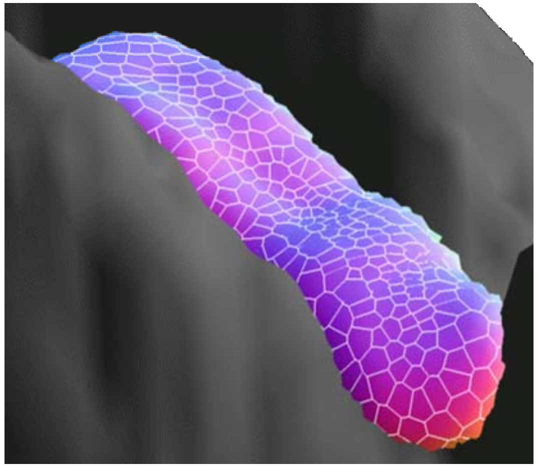


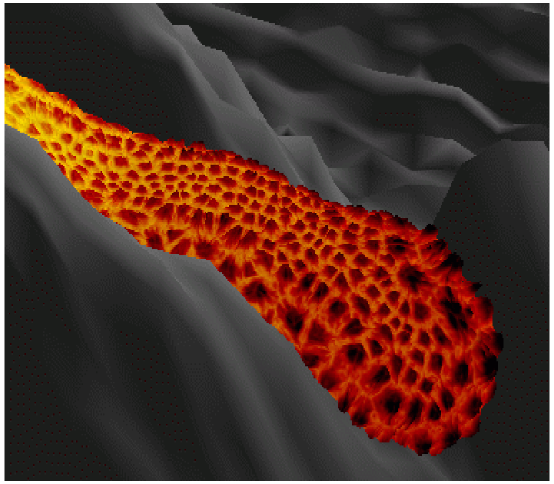


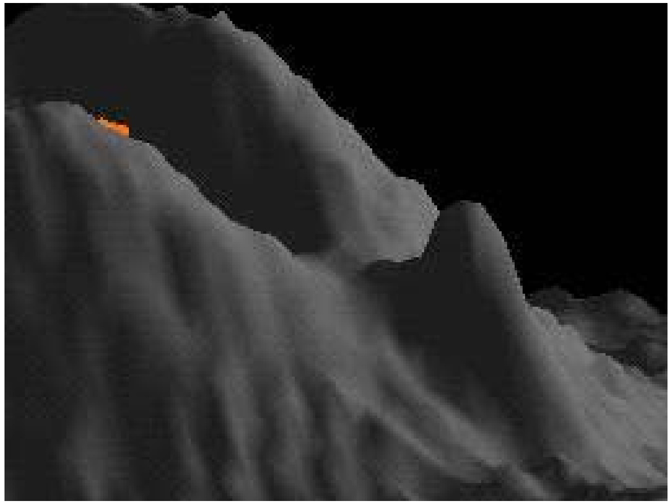


Do it all with point-based physics?
Unstructured objects

Lava flows: Coupling particles and crust *[Stora Agliati Cani Neyret 99]*



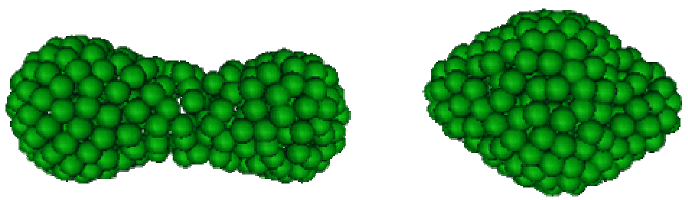


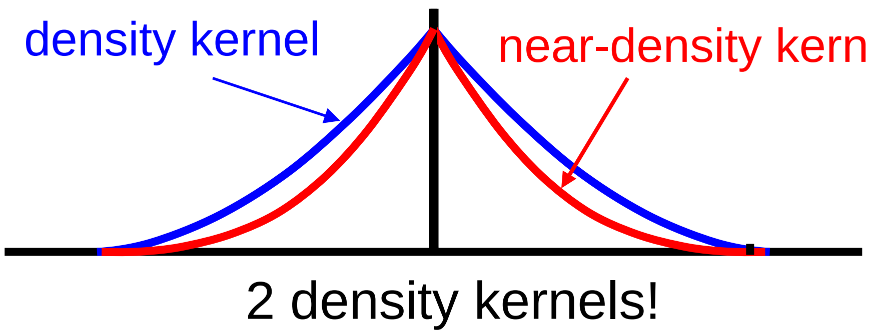


Do it all with point-based physics?
Unstructured objects

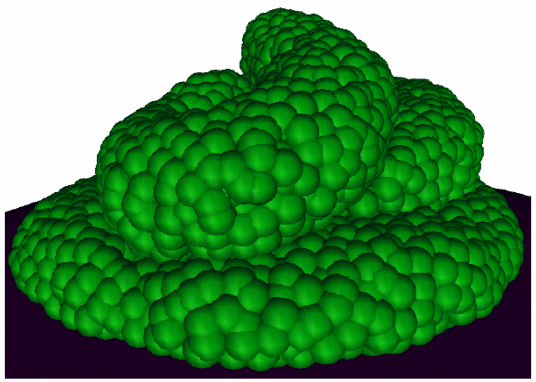
Viscous fluids in real-time [Clavet, Beaudoin, Poulin, SCA'2005]

- Particules SPH : $P = \rho - \rho_0$
- Surface tension effect

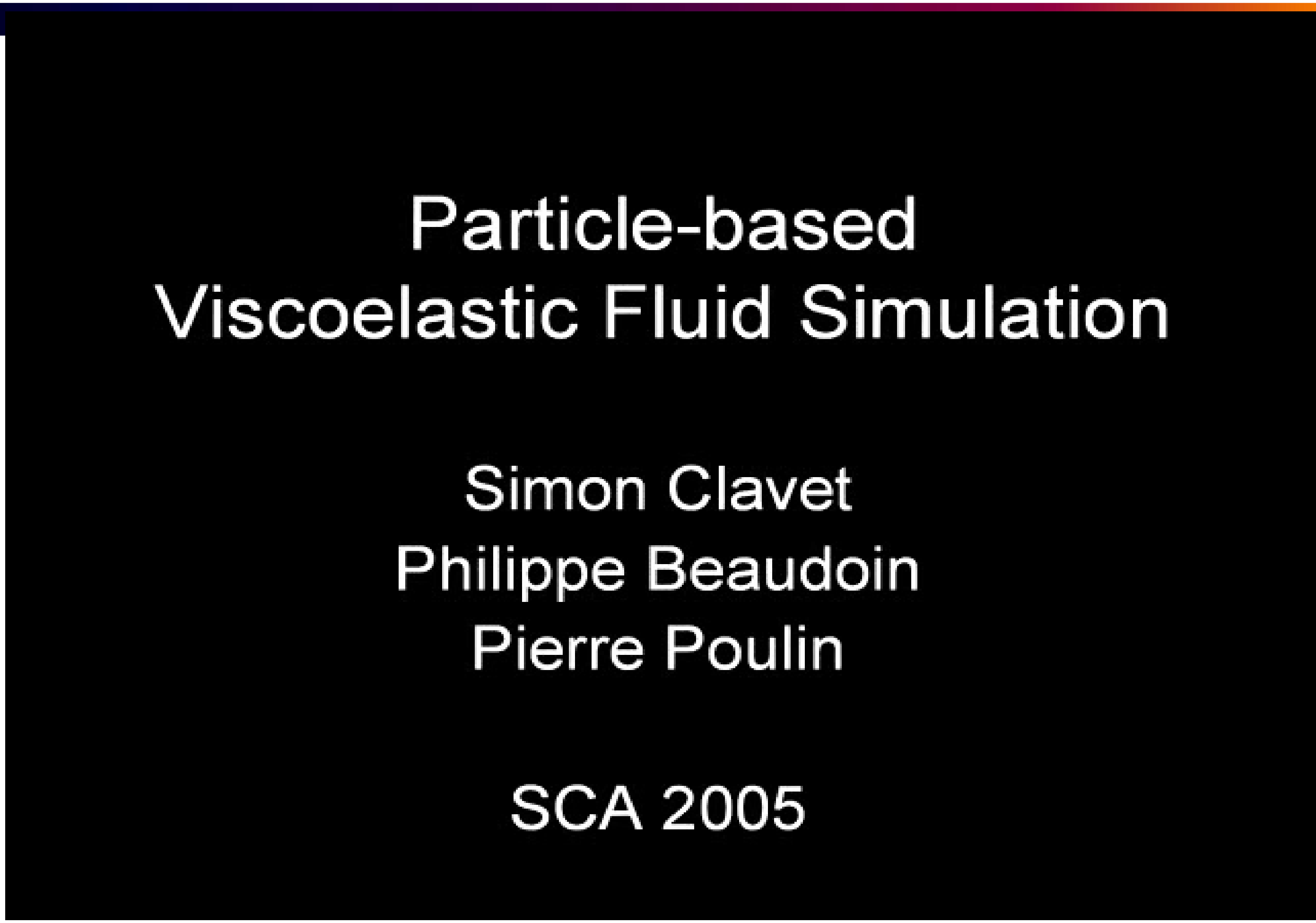




- Real-time for 1500 particles
- Plasticity: add/remove springs



Do it all with point-based physics?
Unstructured objects



Do it all with point-based physics?
Unstructured objects

Bi-phasic fluids

with vortex particles

[Coquerelle, Cottet, Cani 2006]