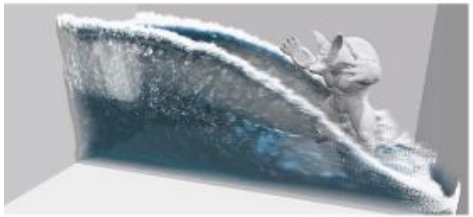
Physics Simulation for 3D graphics and games

Jorg Peters

Understand and simulate physical laws computationally

Science



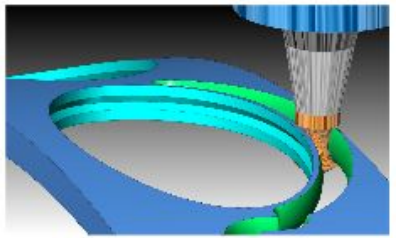
Position Based Fluids [Macklin and Müller 2013]

Games

BeamNG

Movies

Industry



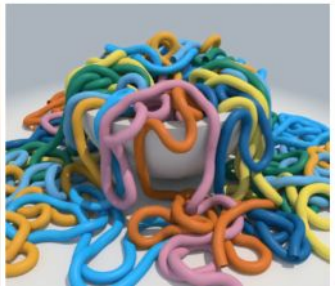
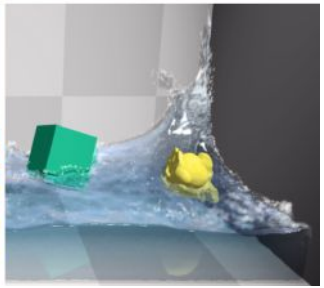
ModuleWorks GmbH

For example: Position-based dynamics

[2015 EG Bender Muller, Macklin]

Symplectic Euler:

$$\mathbf{v}_i(t_0+\Delta t) = \mathbf{v}_i(t_0) + \Delta t \frac{1}{m_i}\mathbf{f}_i(t_0)$$
$$\mathbf{x}_i(t_0+\Delta t) = \mathbf{x}_i(t_0) + \Delta t\, \mathbf{v}_i(t_0+\Delta t).$$

Algorithm 1 Position-based dynamics

1: **for all** vertices i **do**
2: initialize $\mathbf{x}_i = \mathbf{x}_i^0$, $\mathbf{v}_i = \mathbf{v}_i^0$, $w_i = 1/m_i$
3: **end for**
4: **loop**
5: **for all** vertices i **do** $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t w_i \mathbf{f}_{\text{ext}}(\mathbf{x}_i)$
6: **for all** vertices i **do** $\mathbf{p}_i \leftarrow \mathbf{x}_i + \Delta t \mathbf{v}_i$
7: **for all** vertices i **do** genCollConstraints($\mathbf{x}_i \rightarrow \mathbf{p}_i$)
8: **loop** solverIteration **times**
9: projectConstraints($C_1, \ldots, C_{M+M_{\text{Coll}}}, \mathbf{p}_1, \ldots, \mathbf{p}_N$)
10: **end loop**
11: **for all** vertices i **do**
12: $\mathbf{v}_i \leftarrow (\mathbf{p}_i - \mathbf{x}_i)/\Delta t$
13: $\mathbf{x}_i \leftarrow \mathbf{p}_i$
14: **end for**
15: velocityUpdate($\mathbf{v}_1, \ldots, \mathbf{v}_N$)
16: **end loop**

Interaction through forces:

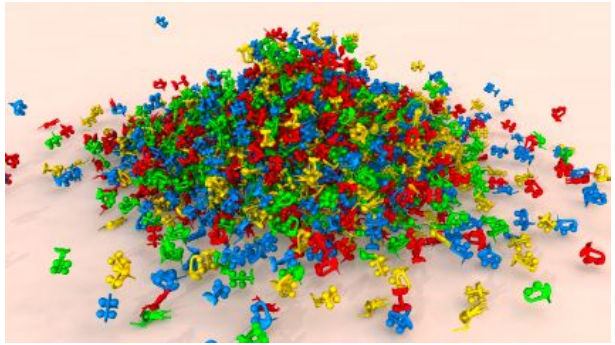
- Gravity, magnetism
- Friction, collision
- Chemical ties

Interaction through forces:

Kinematics → motion (x,x',x")
Dynamics → forces and masses (inertia, collision, …)

Control dynamics to achieve kinematic effect:

- Direct: skeleton, game controller
- Indirect: collision, soft deformation



[Deul el al., Position-Based Rigid Body Dynamics, 2014]

Difference to physics class:

- Visually convincing vs discovery
- Robustness (stability, convergence) vs correctness
- Efficiency (real-time) vs accuracy

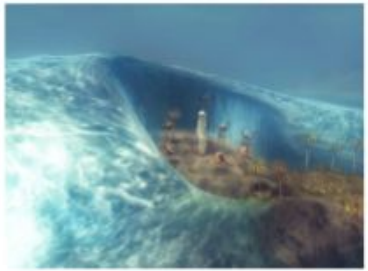
Our Method
10 iteration, 50 ms

Exact Solution
Newton's Method, 13 s

[Liu *et al.*, Fast Simulation of Mass-Spring Systems]

3 models:
Physical
Collission
Visual

- Objects as collections of points (molecules).
- Rigid bodies.
  - Deform as a single piece.
- Soft bodies.
  - Each point deforms locally in a continuum.
- Detachable bodies.
  - Several objects either stick together or break apart.



MATH

Numerical Computation needs

Linear Algebra: matrices ...

Calculus: differentiation, integration, ode, vectors

LECTURES

- Physics Engines
- (Calculus & Algebra)
- ODE & Rigid-body physics
- Collision (space partition)
- Fluids
- Elasticity (soft body)

IMPLEMENTATION & PRACTICE

- Two smaller projects
- One big project + presentation (recent paper)

- Groups of 4
- probably a choice of platform:
  Physx, SOFA, C++

- One test