CS-C3100 Computer Graphics
Jaakko Lehtinen

9.1 Intro to Dynamical Simulation

Futuremark Corp., used with permission

In This Video

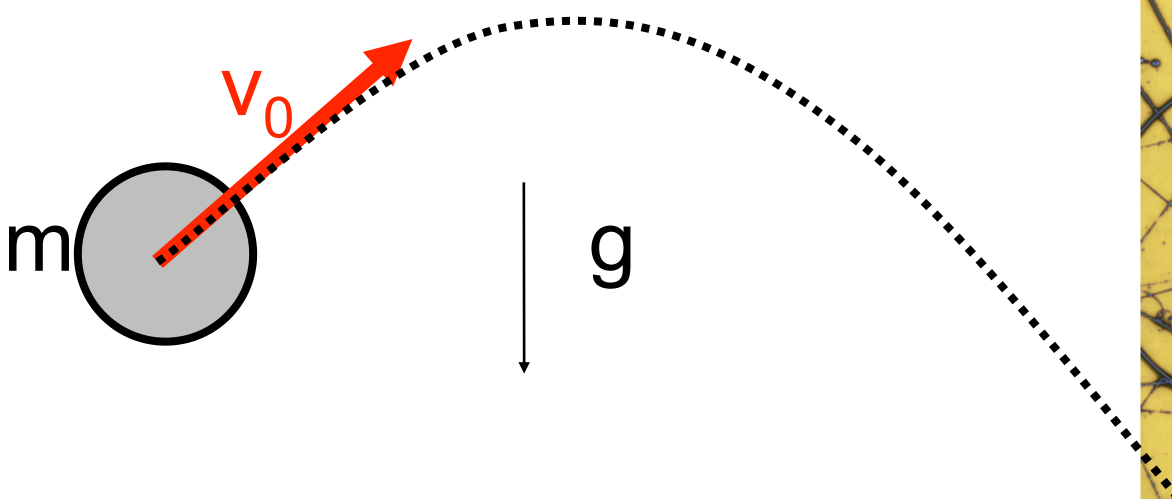
- A tour of physically-inspired animation techniques

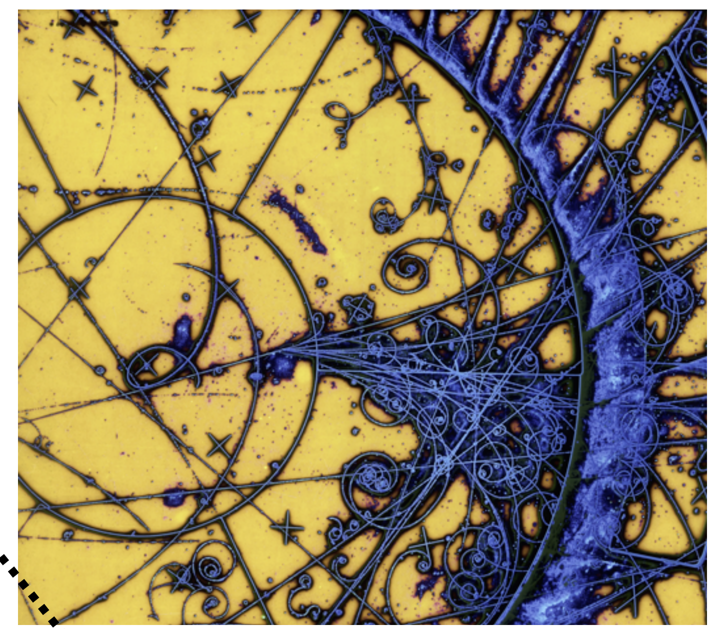
Types of Animation

- Keyframing
- Procedural
- **Physically-based**
  - Particle Systems
    - Smoke, water, fire, sparks, etc.
    - Usually heuristic as opposed to simulation, but not always
    - Mass-Spring Models (Cloth)
  - *Continuum Mechanics (fluids, etc.), finite elements*
    - *Not in this class*
  - *Rigid body simulation*
    - *Not in this class*

Types of Animation: Physically-Based

- Assign physical properties to objects
  - Masses, forces, etc.
- Also procedural forces (like wind)
- Simulate physics by solving equations of motion
  - Rigid bodies, fluids, plastic deformation, etc.
- Realistic but difficult to control



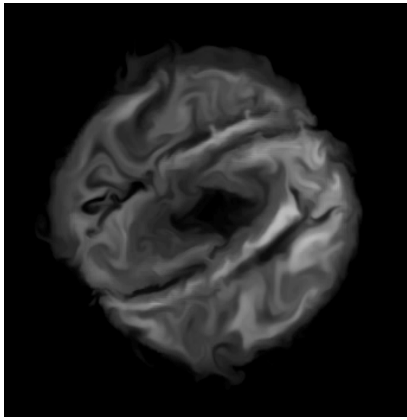


Control (Buster Keaton)

Steamboat Bill, Jr. (1928)

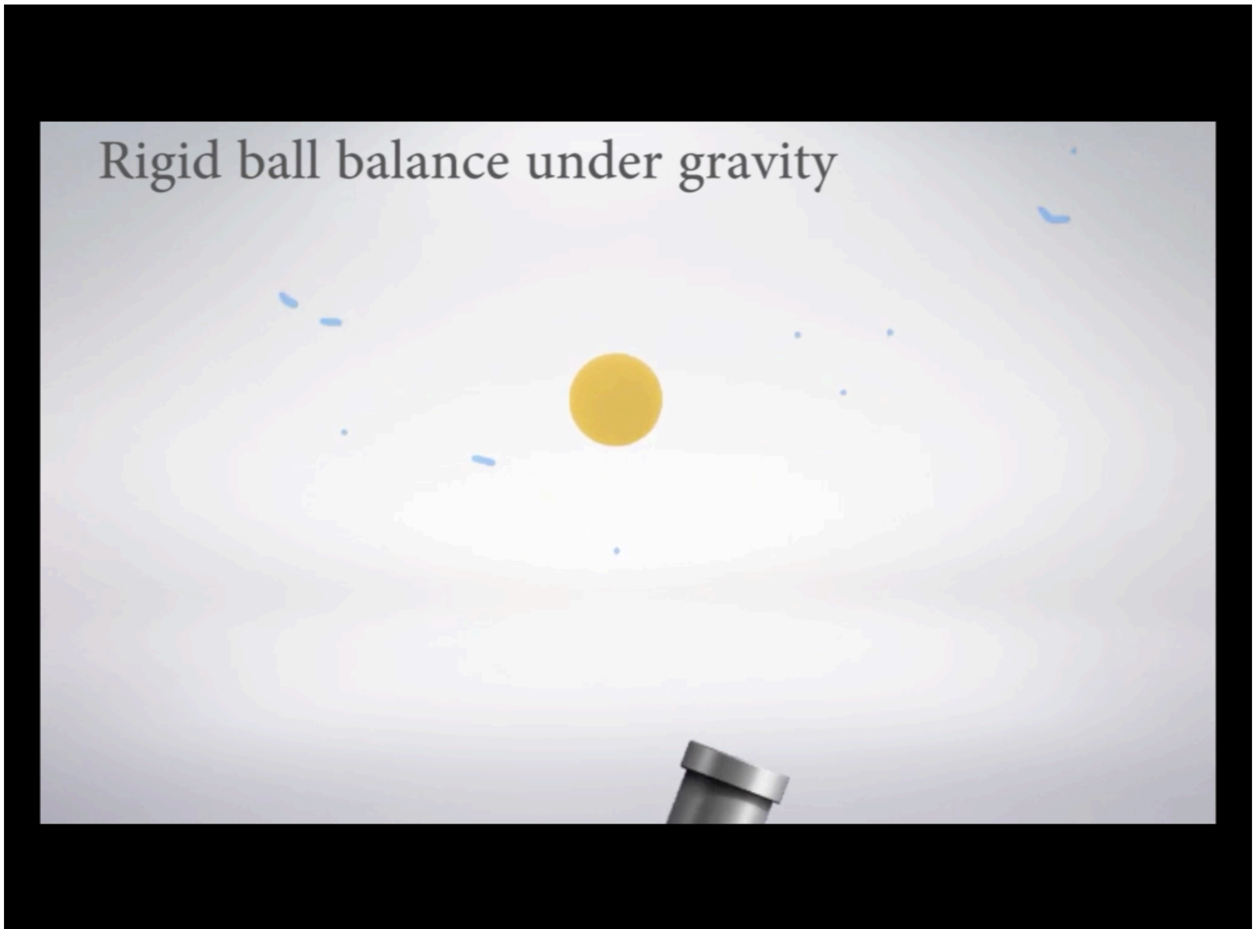
"Directable Simulation"

- Aim to produce simulations that obey user-specified constraints
  - Add as-small-as-possible fictitious forces to simulation to nudge it towards artist/TD needs
- Lots of cool stuff!
- See e.g. http://www.cs.huji.ac.il/labs/cglab/projects/tdsmoke/



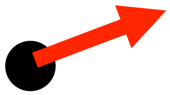
Modern: Let the AI Drive

- Ma et al., SIGGRAHP 2018, Fluid directed rigid body control using deep reinforcement learning



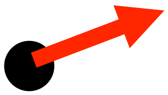
Types of dynamics

- Point



Types of dynamics

- Point

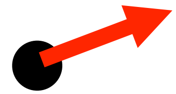


- Rigid body

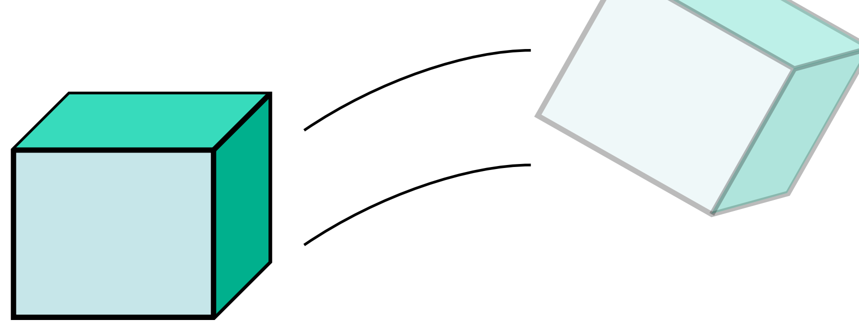
Animation by Mark Carlson

Types of dynamics

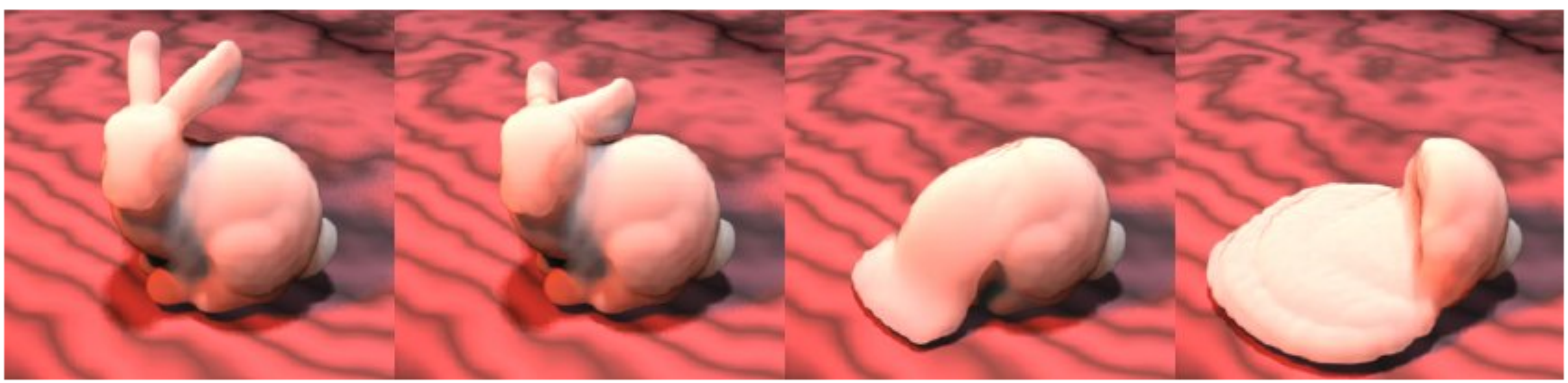
- Point



- Rigid body



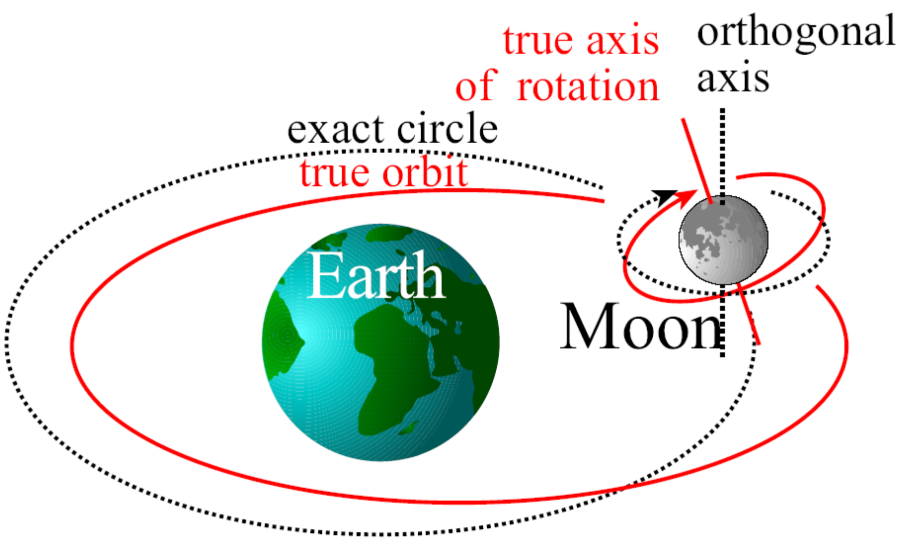
- Deformable body
(include clothes, fluids, smoke, etc.)

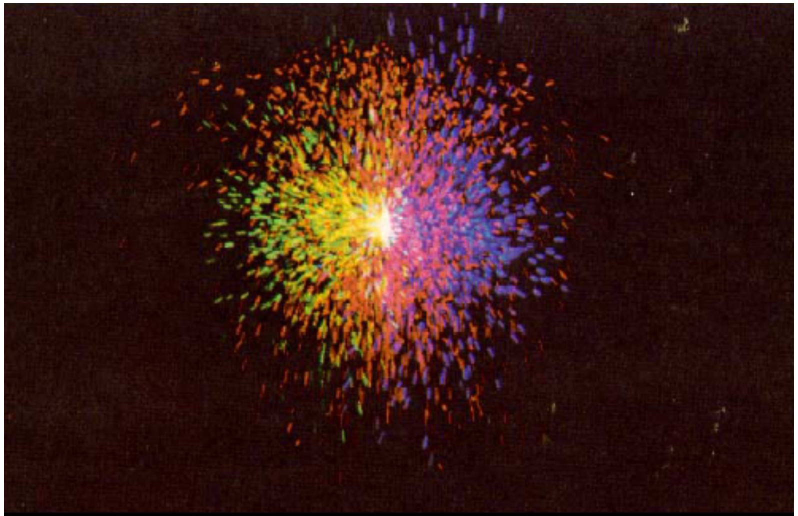


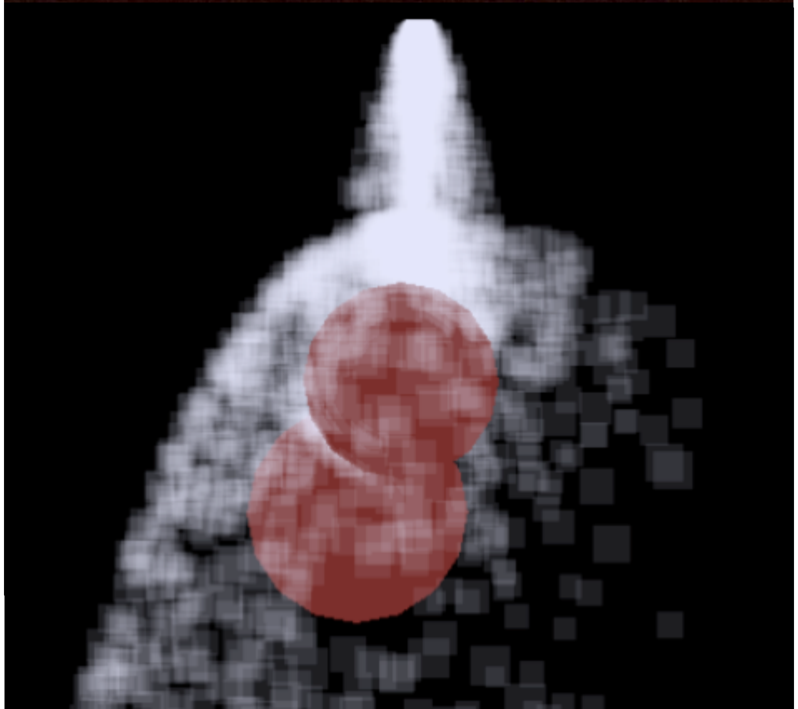
Mark Carlson

We Focus on Point Dynamics

- Lots of points!
- "Particle systems"
  - Borderline between procedural and physically-based

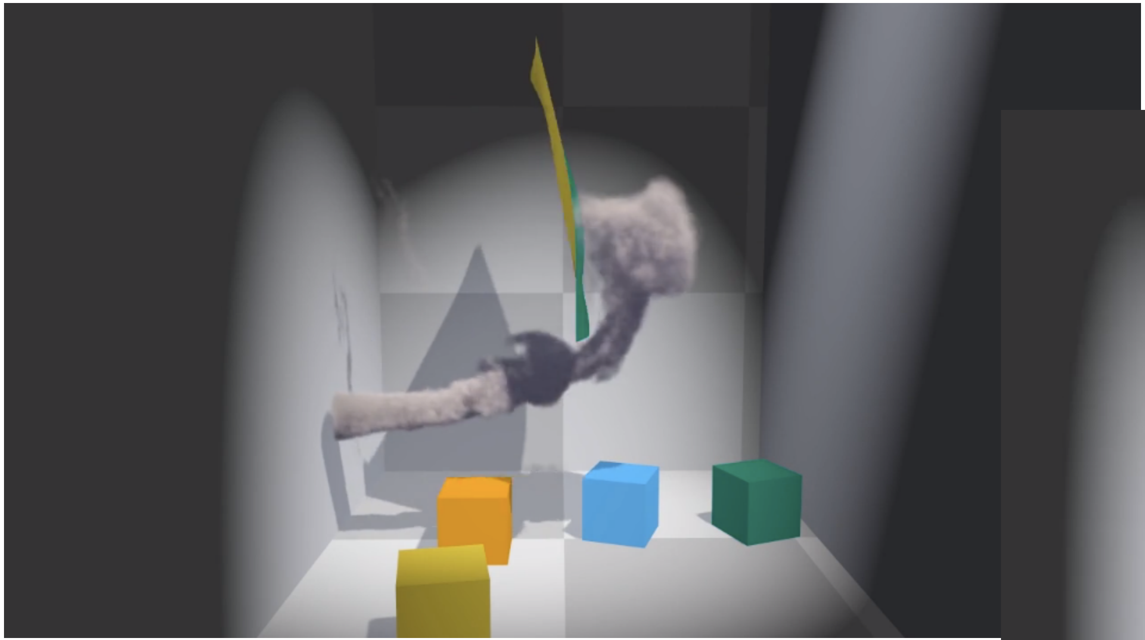


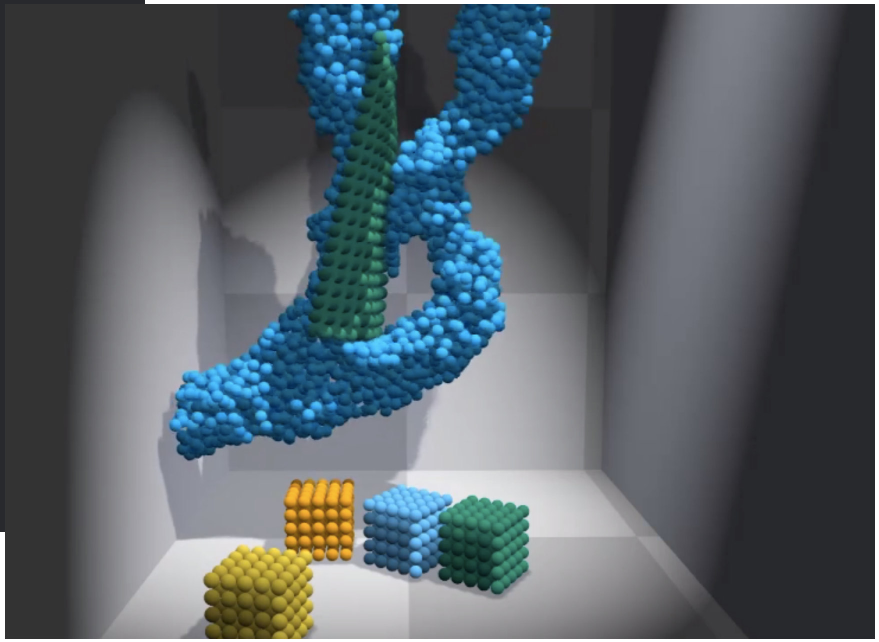


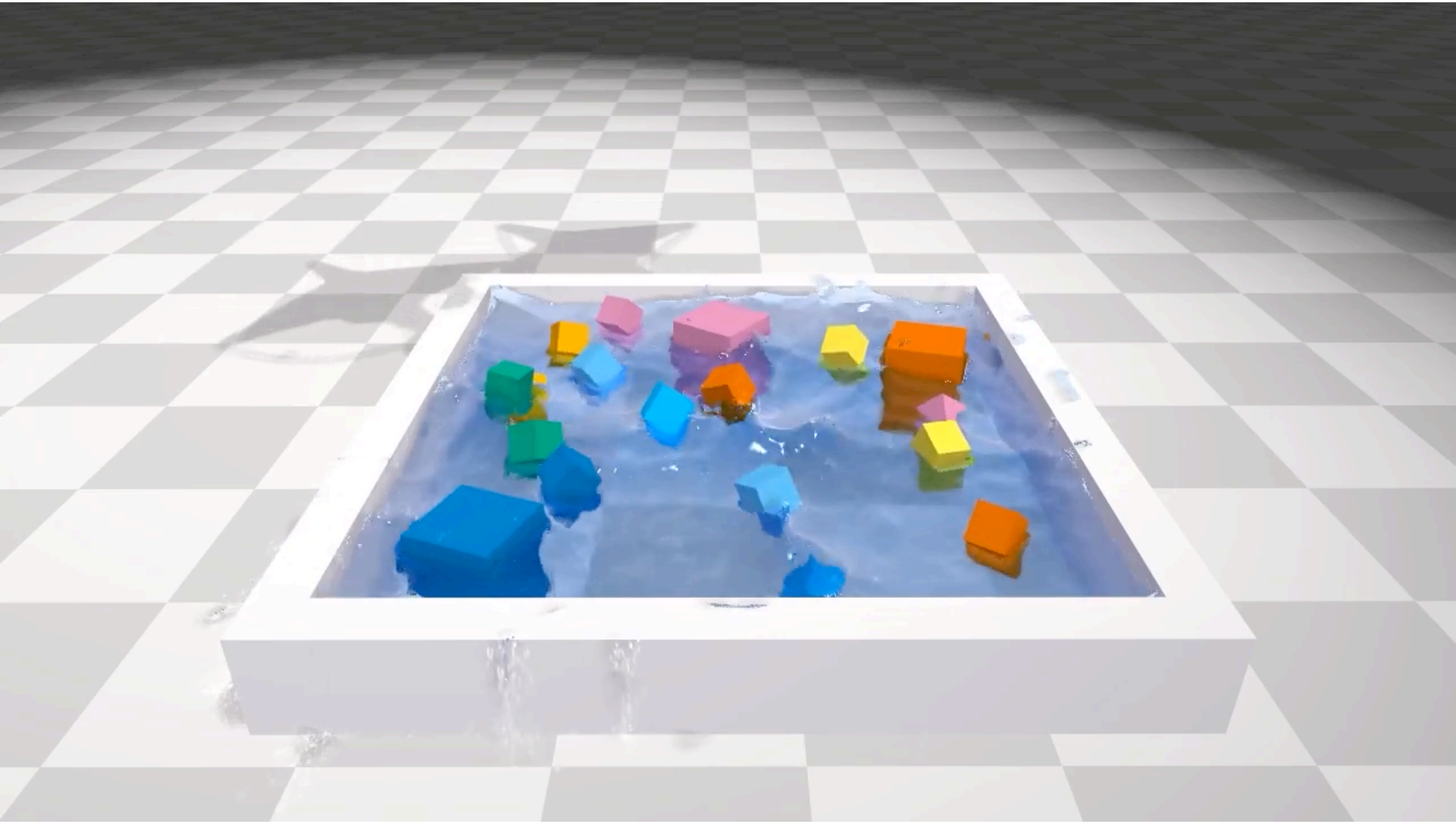


Can Model Everything using Particles!

- Unified Particle Physics for Real-Time Applications (Chantanez et al. SIGGRAPH 2014)
  - video







Real-Time Particles Demo (3DMark03)


3DMARK03

Generalizations (later)

- Mass-spring and deformable surface dynamics
  - surface represented as a set of points
  - forces between neighbors keep the surface coherent

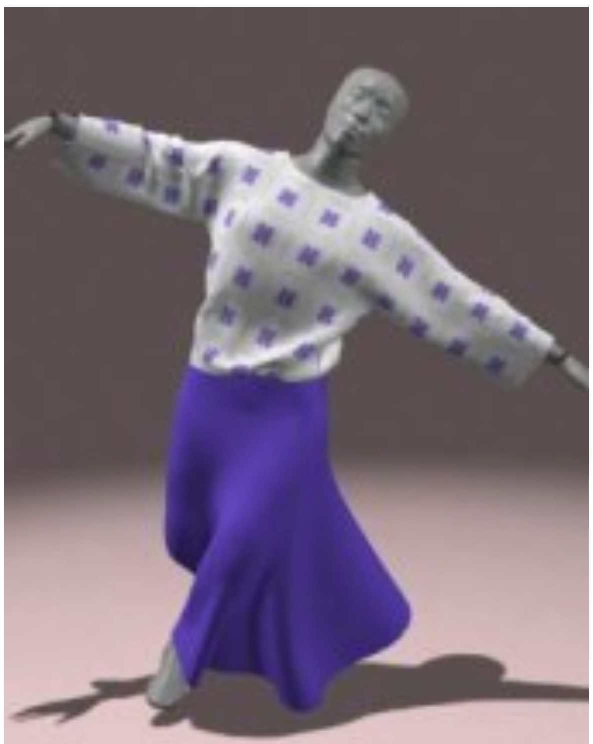


Image Witkin & Baraff

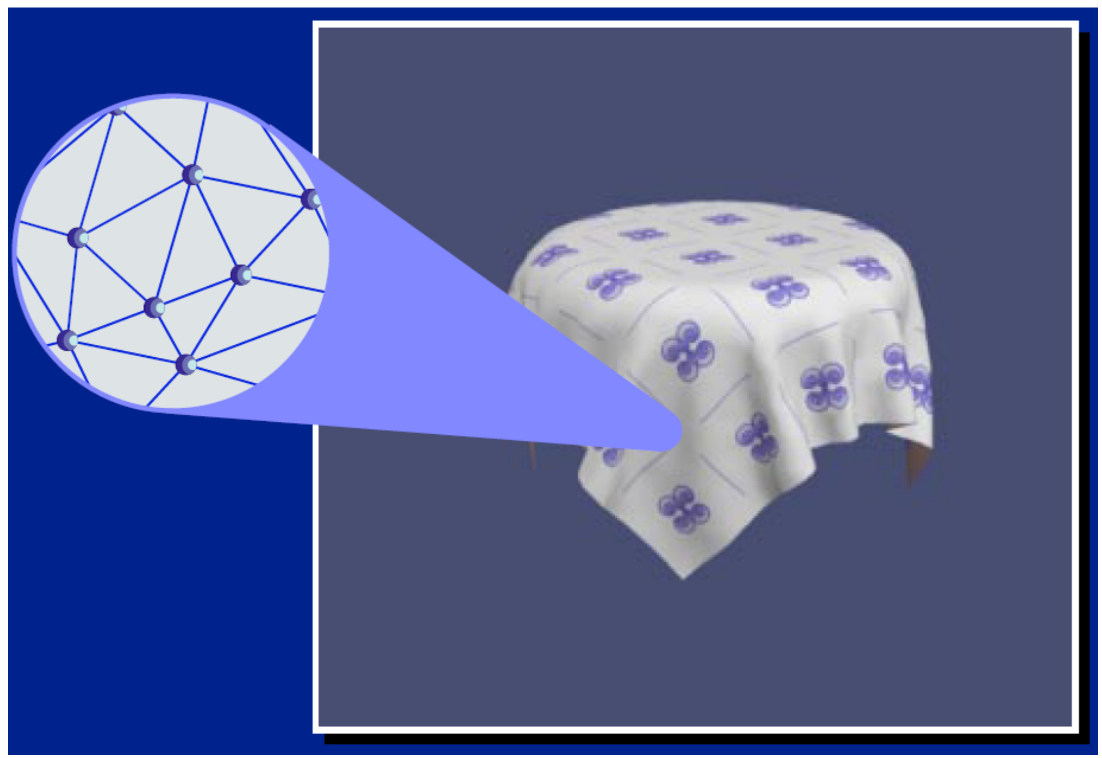


Image Michael Kass

Take-Home Message

- Particle-based methods can range from pure heuristics (hacks that happen to look good) to "real" simulation

- Basics are the same:
  Things always boil down to integrating ODEs!
  - Also in the case of grids/computational meshes

Andrew Selle et al.

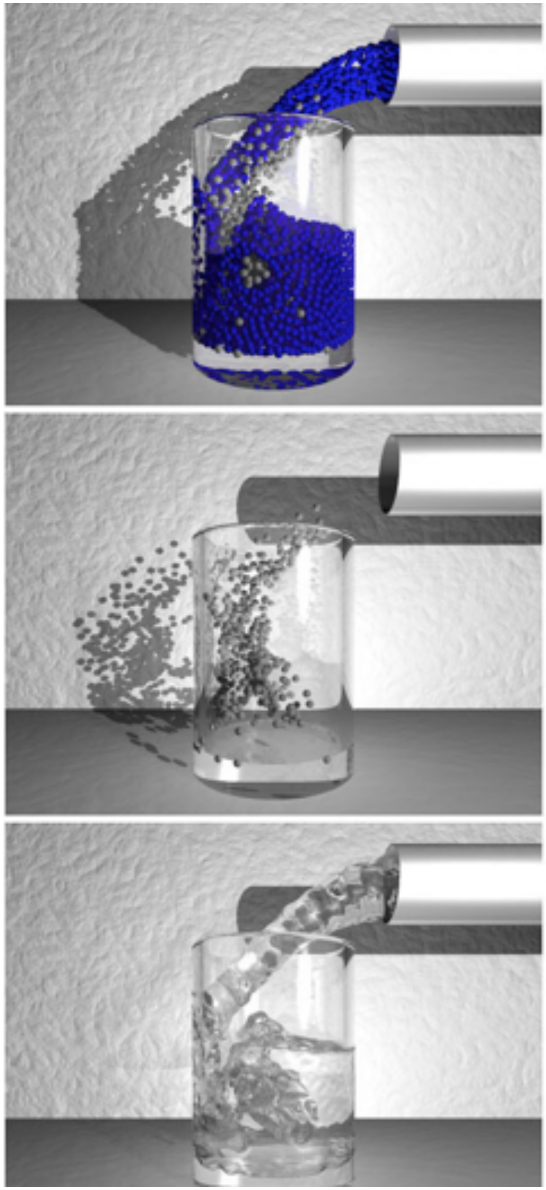
Further reading

Cloth Video

Selle, A., Su, J., Irving, G. and Fedkiw, R., "Robust High-Resolution Cloth Using Parallelism, History-Based Collisions, and Accurate Friction," IEEE TVCG 15, 339-350 (2009).

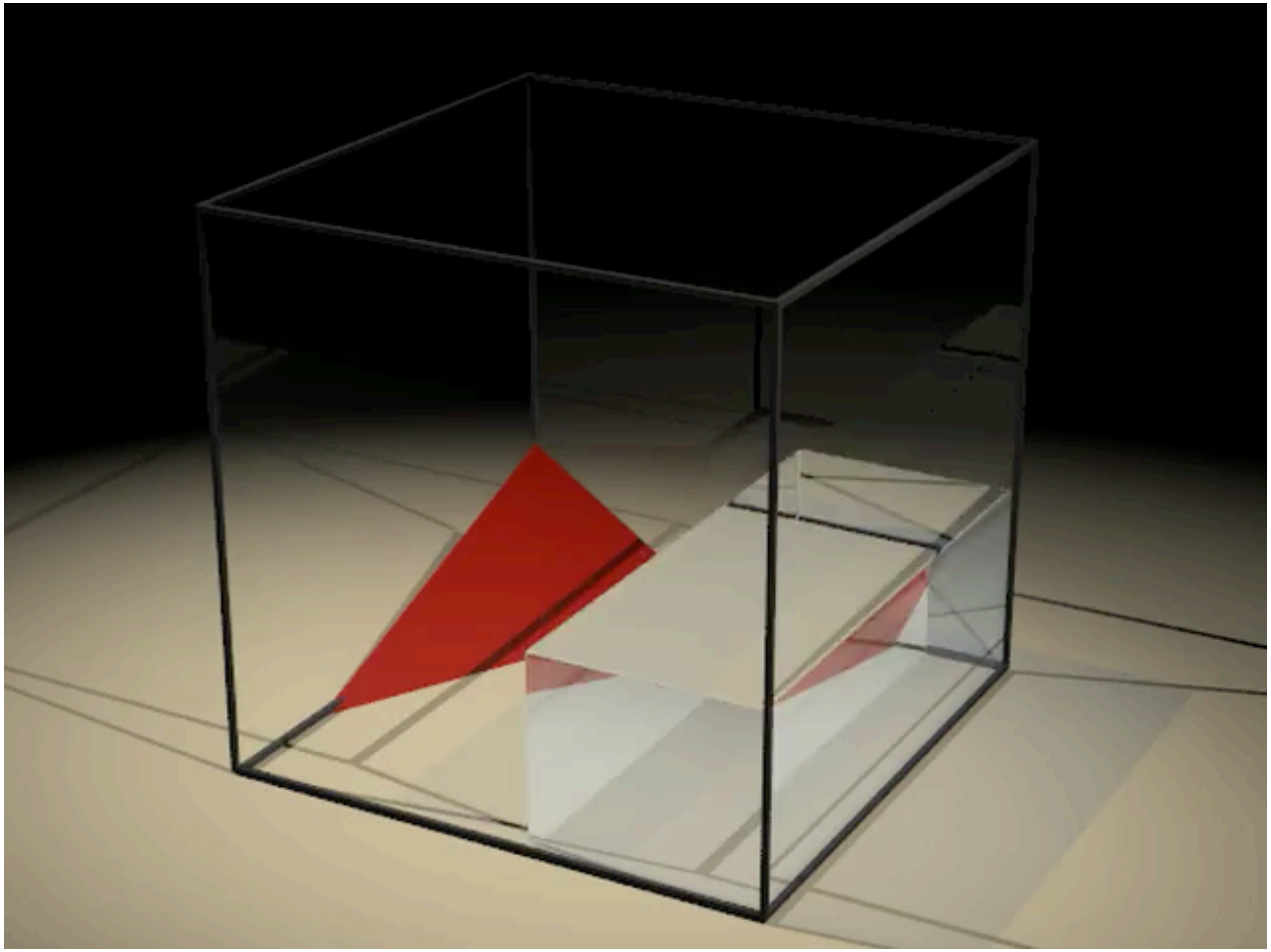
Generalizations

Müller et al. 2005



- It’s not all hacks:
  Smoothed Particle Hydrodynamics (SPH)
  – A family of “real” particle-based fluid simulation techniques.
  – Fluid flow is described by the Navier-Stokes Equations, a nonlinear partial differential equation (PDE)
    - SPH discretizes the fluid as small packets (particles!), and evaluates pressures and forces based on them.

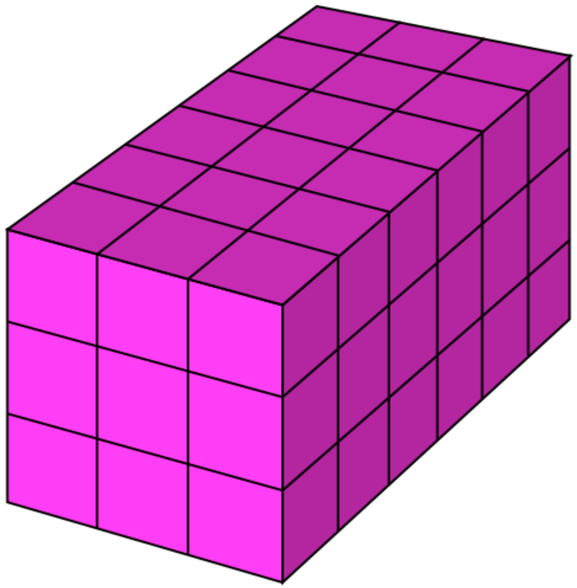
SPH Example



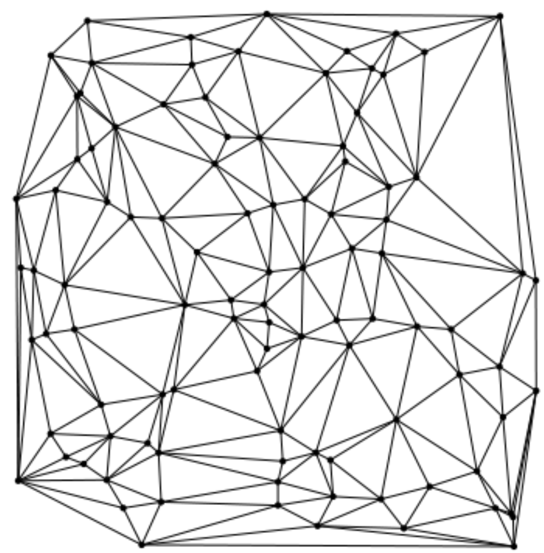
Predictive-Corrective Incompressible SPH. Barbara Solenthaler, Renato Pajarola. ACM Transactions on Graphics (SIGGRAPH), 2009

Meshless Techniques

- Most simulation techniques work on either regular grids or meshes constructed from triangles/tets
- PDEs defined on space are discretized on the grid.



Regular 3D grid

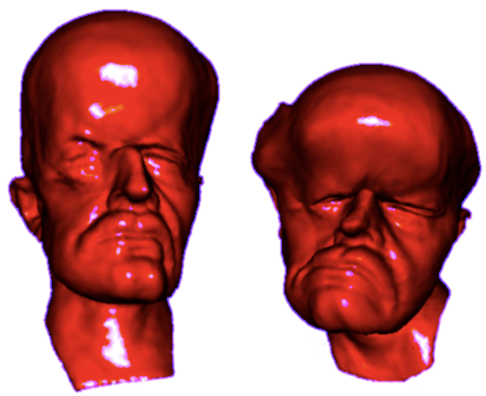


Irregular 2D grid

Images: Wikipedia

Meshless Techniques

- Most simulation techniques work on either regular grids or meshes constructed from triangles/tets
- In contrast, so-called *meshless methods* do not require the underlying space to be discretized
  - Instead, represent things using points (particles!)
  - They can still be "well-founded": SPH is an example.
  - Another example: Point-Based Animation of Elastic, Plastic and Melting Objects (Müller, Keiser, Nealen, Pauly, Gross, Alexa, SCA 2004)



Müller et al.

That’s all!

- Next time: particle systems