CS888 Advanced Topics in Computer Graphics:

Physics-Based Animation

Jan. 4, 2016

Physics-Based Animation

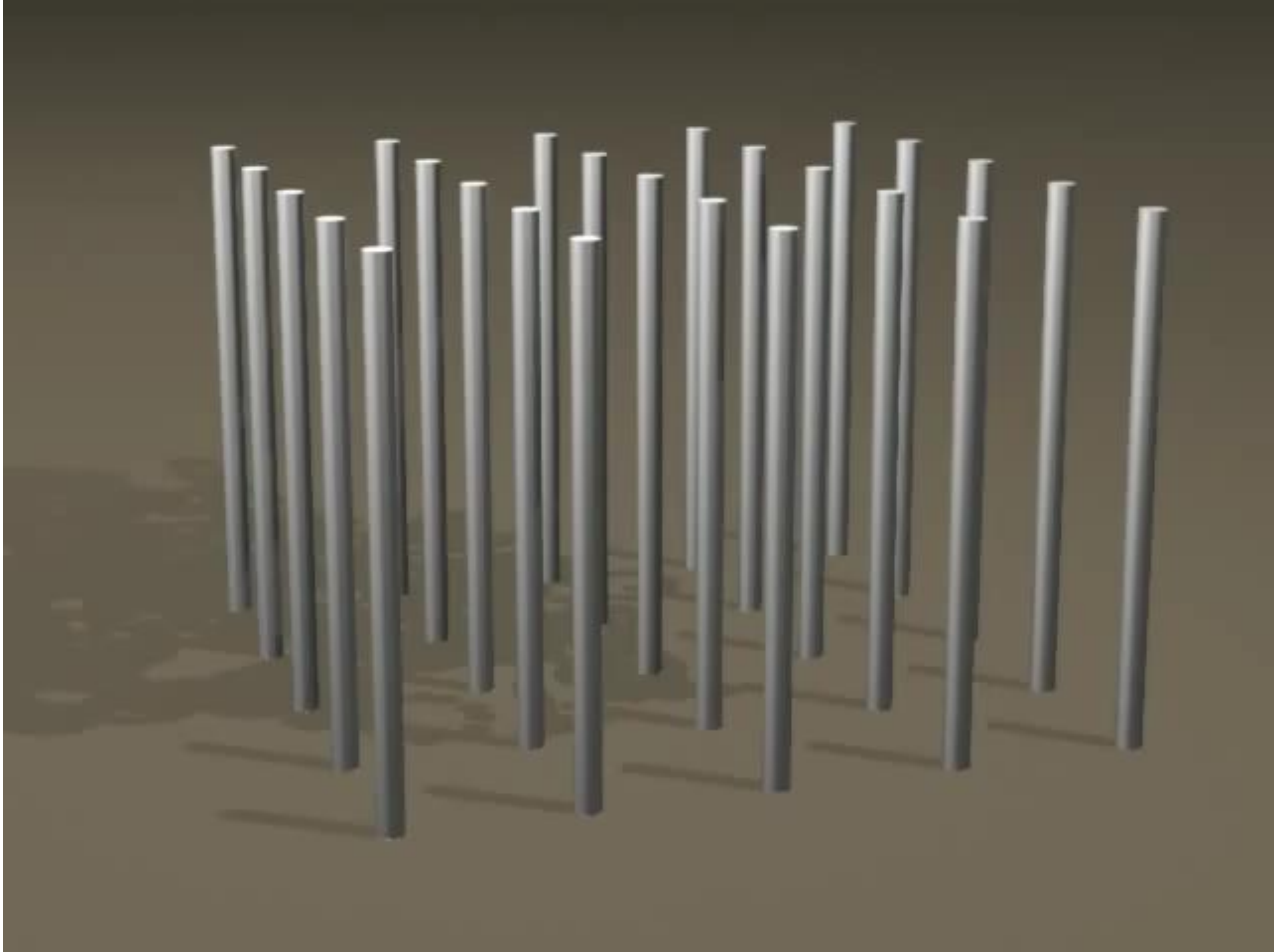
The use of physical simulation to generate animations of:

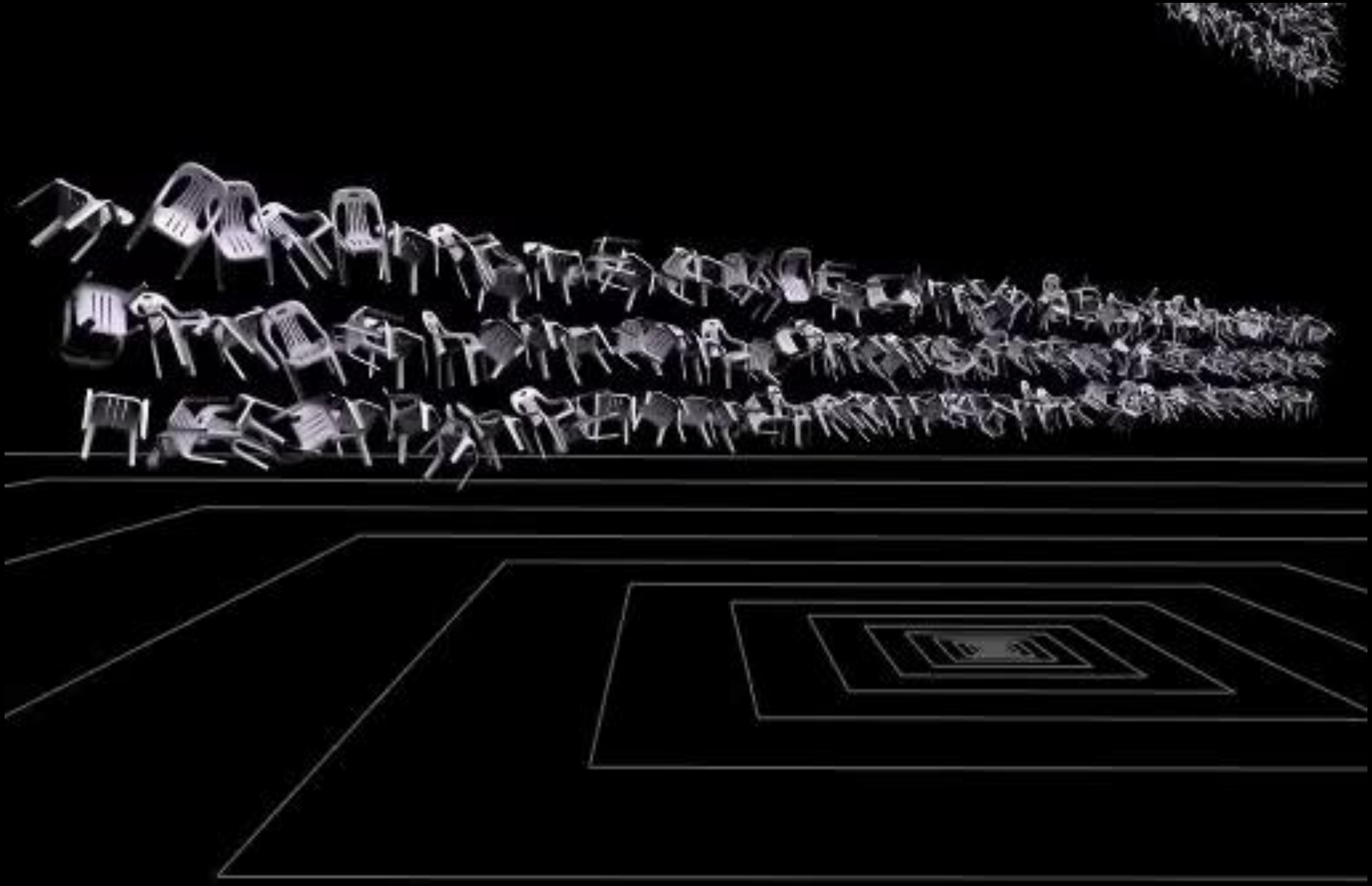
- Rigid bodies: “Perfectly” stiff or rigid objects
- Deformable objects: flesh, rubber, jello
- Shells/plates: Cloth, paper, sheet metal, plant leaves
- Rods/beams: Hair, strands, cords, slender tree branches
- Gases: Air, fire, smoke, explosions, bubbles
- Liquids: Water, oil, honey, slime, goop, oceans, waves

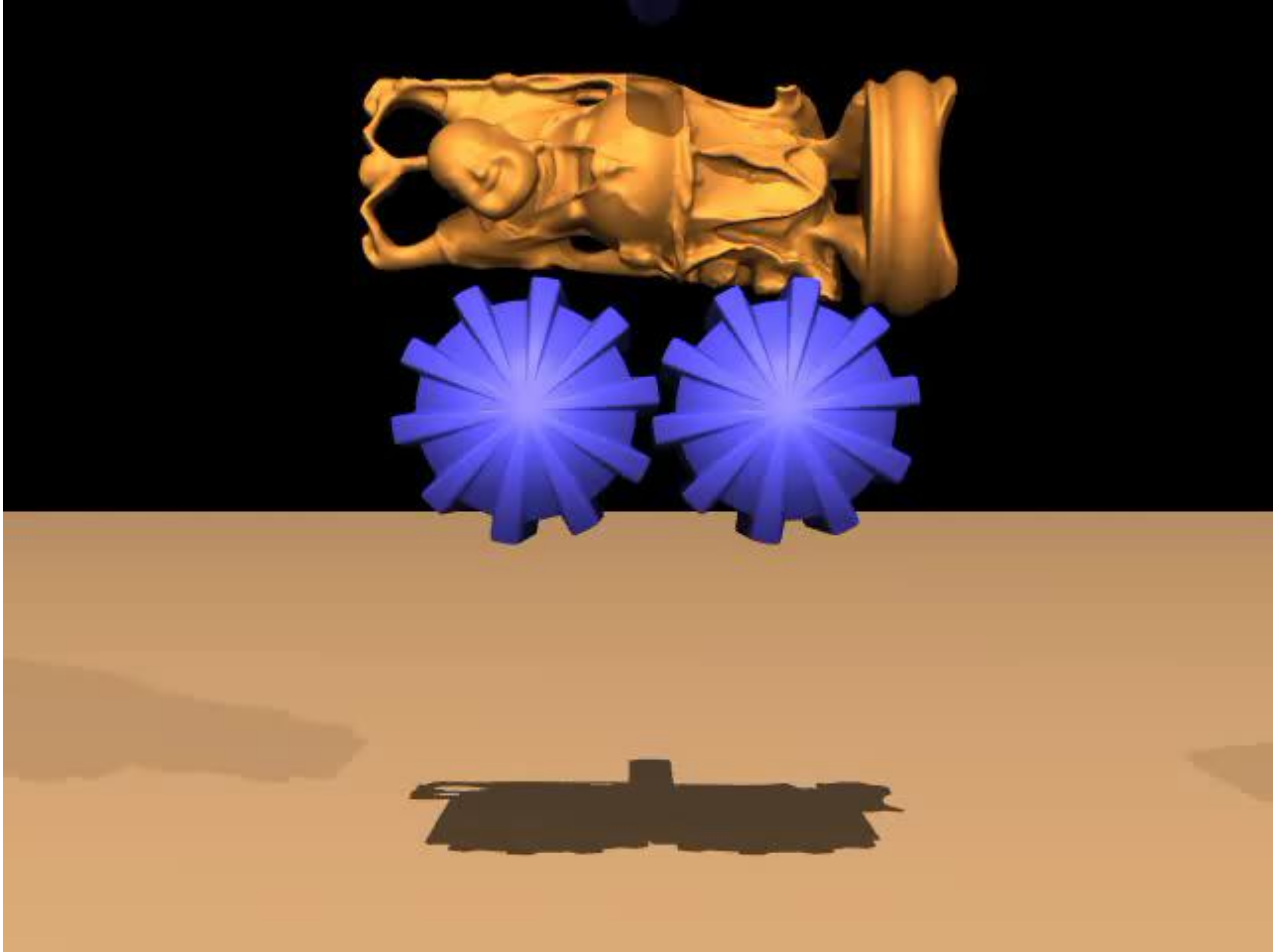
…and (m)any other visually interesting physical phenomena.

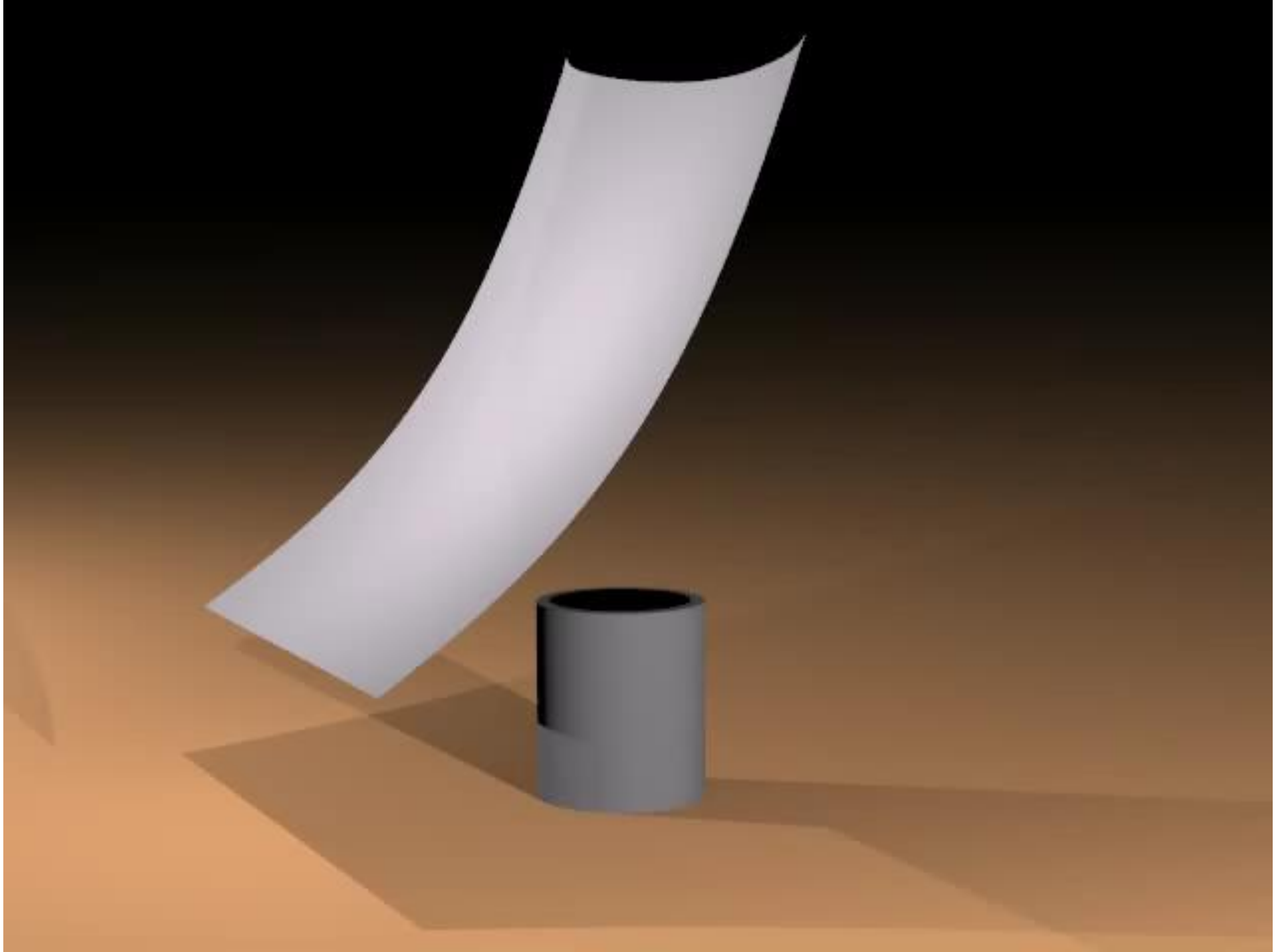
Why Use Simulated Physics?

- Too many *degrees of freedom* to model each by hand.
- Humans are good at spotting physical irregularities ("weird" motion).
- Save artists time (avoid "simulating" in their heads!) to instead focus on characters, story, aesthetics, etc.
- Directly capturing real motion (via video camera or motion capture, etc.) can be limiting.
- Simulation is often cheaper, safer, and makes otherwise "impossible" scenarios feasible.
- For interactive applications, animations must respond *on-the-fly* in a flexible way.

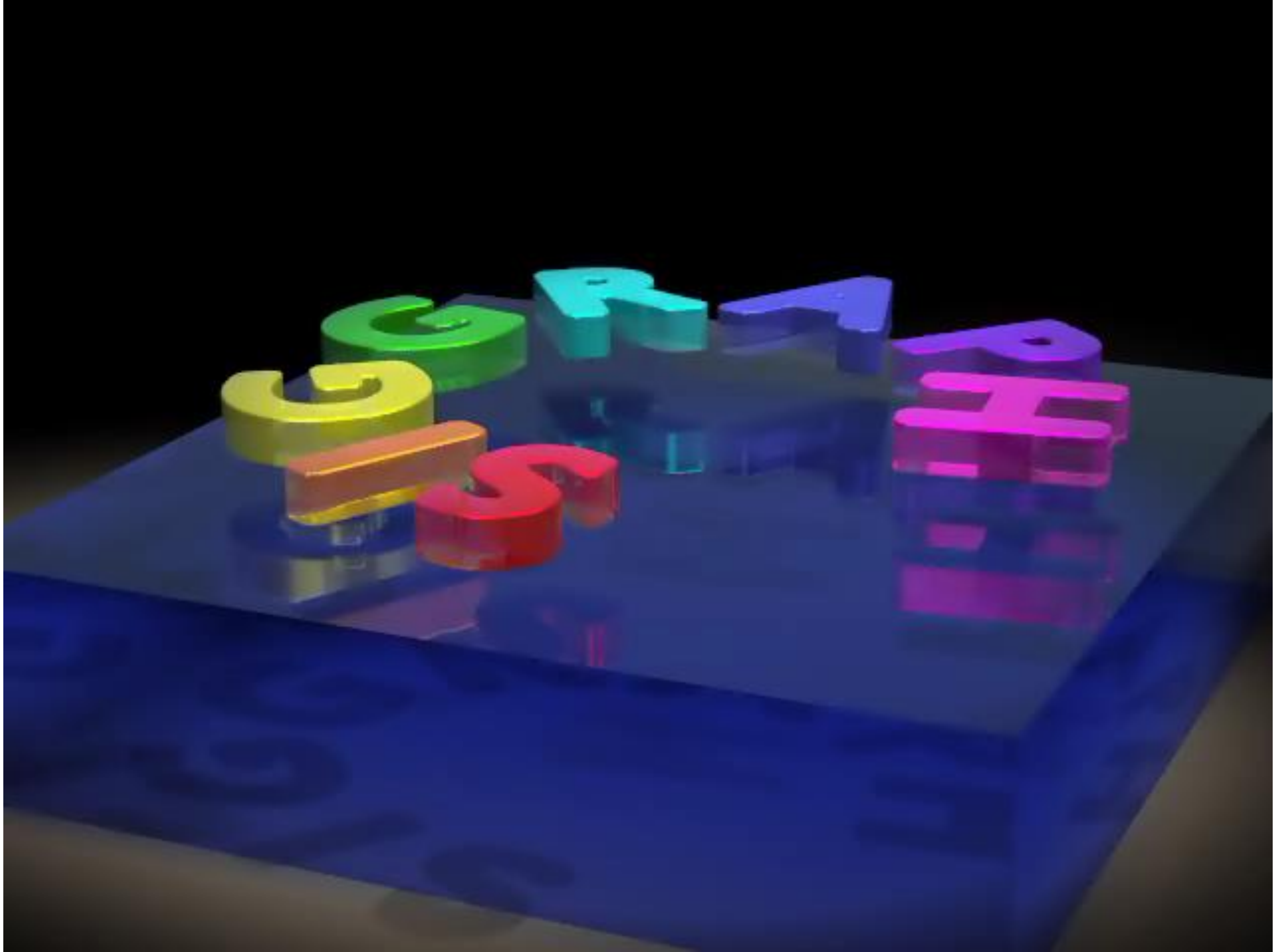






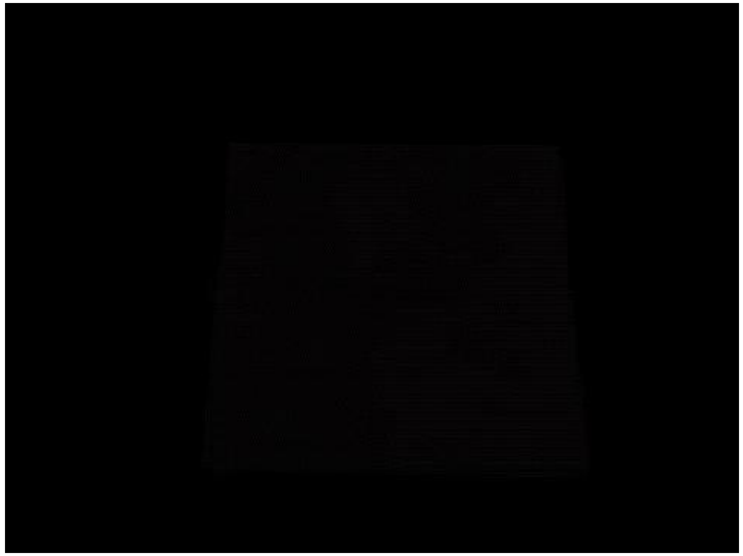


Helical perversion

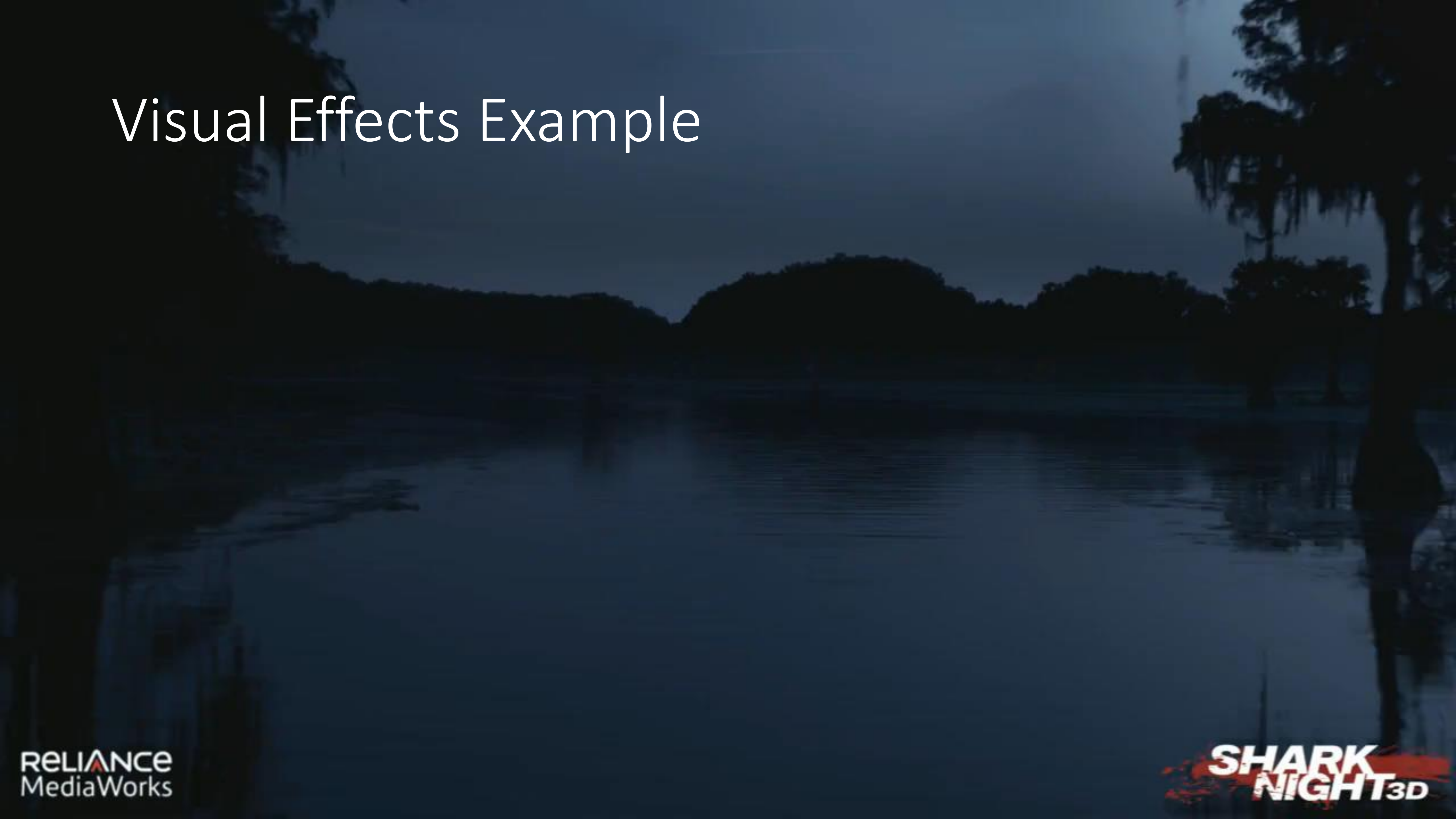

SIGGRAPH

Applications – Graphics and more

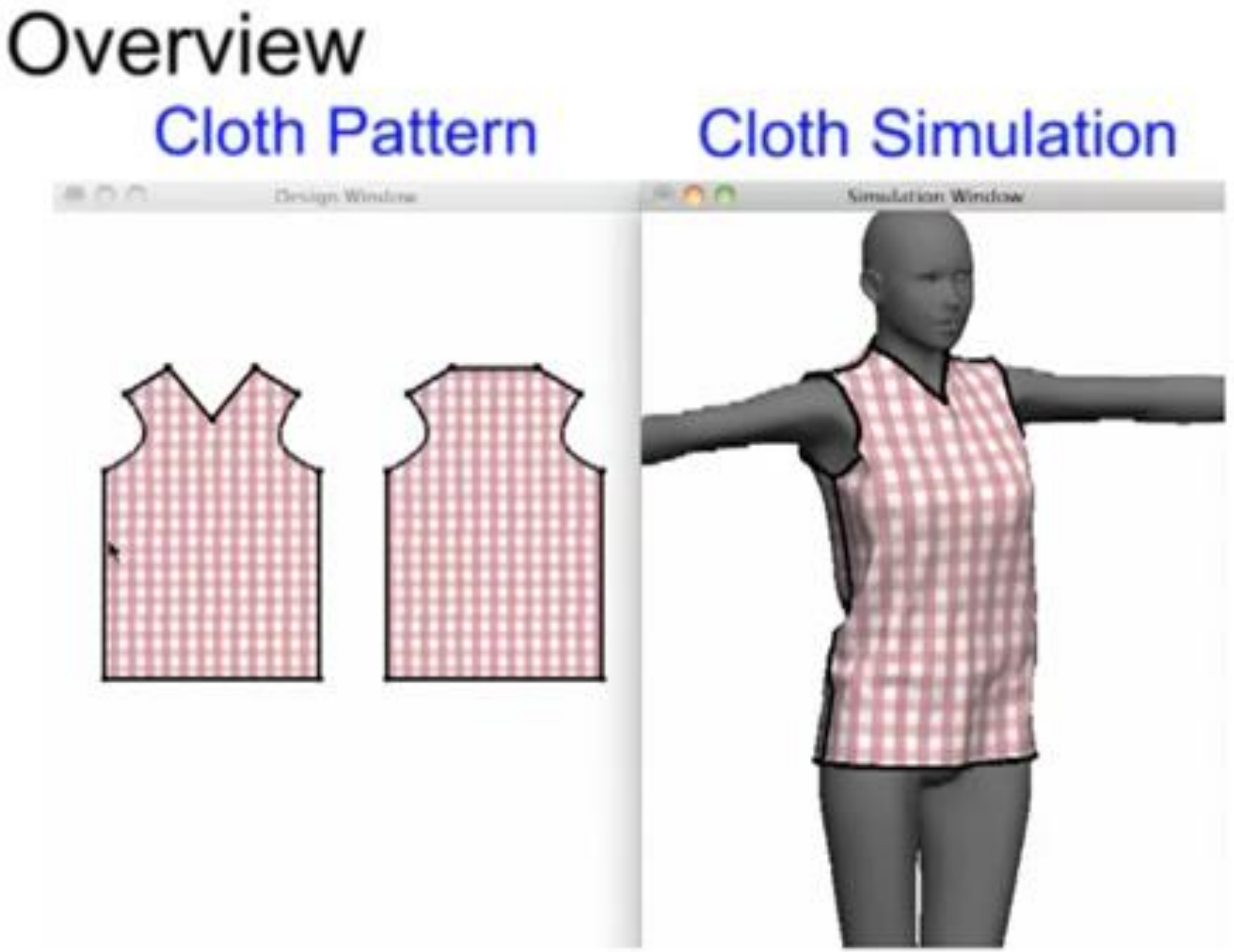
- Visual effects & animated movies
- Computer gaming
- Virtual surgery, and similar training/education tools
- “Virtual fitting room”
- Interactive design/fabrication
  - architecture, fashion, 3D printing...
- Similar techniques are applied in engineering, scientific computing, etc.



Visual Effects Example



Design – 3D Tailoring



SIGGRAPH 2011: Sensitive Couture for Interactive Garment Editing and Modeling

Design – Fabrication / 3D Printing

Spin-It: Optimizing Moment of Inertia for Spinnable Objects

Moritz Bächer — Disney Research Zurich
Emily Whiting — ETH Zurich
Bernd Bickel — Disney Research Zurich
Olga Sorkine-Hornung — ETH Zurich

Disney Research ETH zürich igl

SIGGRAPH 2014: Spin-It: Optimizing Moment of Inertia for Spinnable Objects

Thrilling Administrative Details!

Course Organization

Mon/Wed at 2:30-3:50pm in DC 3313.

Instructor: Christopher Batty (DC 3605)
Office hours by appointment (email me).

E-mail: christopher.batty@uwaterloo.ca

Course web page:

https://cs.uwaterloo.ca/~c2batty/courses/CS888_2016/

Grades will be posted on LEARN:

https://learn.uwaterloo.ca/

Piazza forums

Course link: http://piazza.com/uwaterloo.ca/winter2016/cs888/home

The sign-up link is on the course website.

Used for course announcements, online discussion, questions, etc.

Feel free to email me, but if it's something that others could also benefit from, **please use the Piazza forum**.

Course Organization

- Primarily seminar-style – paper reading, paper presentations, and group discussions.
- A few lectures to set the stage.
- Course project – implement a physical simulation technique.
- Do one paper review per week.

- See preliminary schedule on the website. (Roughly: first 2/3 on solids of various kinds, last 1/3 on fluids.)

Grade Breakdown

- Project: 40%
- Presentations: 25%
- Reviews: 20%
- Participation/discussion: 15%

Late penalty of 25% per day.

Attendance is expected at all classes. If a class must be missed for research (conference, deadline, etc.), **notify me one week prior**.

Background & Resources

- You should have some familiarity with computer graphics and numerical methods.
- I’ll cover some background in the first few lectures. If something is unfamiliar, let me know.

- A nice general source for basics is Baraff & Witkin’s SIGGRAPH course notes (albeit slightly dated).
- Links to a variety of additional (optional) material are on the web site.

Presentations

Describe :

- Key *novel* elements of the paper, and their significance.
- Relationship to similar work.
- Strengths and weaknesses.
- Possible future extensions or directions.

Length: 20-25 minutes.

2 presentations each over the term.

Steve Mann has some advice for giving talks:

http://www.cgl.uwaterloo.ca/~smann/Talks/CGL.98.11.24/

http://www.cgl.uwaterloo.ca/~smann/GSInfo/talk_guidelines.html

You can find many other good sources online.

Presentations

- See the list of topics (by week) and corresponding papers to choose from on the course website.

https://cs.uwaterloo.ca/~c2batty/courses/CS888_2016/schedule.html

- Email me your top 3 preferred slots for the first round of presentations by Friday at noon. (No guarantees.) First slot is Jan 18.

- The 1st round topics are: (1) Rigid bodies, (2) Deformables, (3) Cloth & shells.

Presentations

A typical format is…

- Motivate the topic/problem
- Briefly outline key related work
- Describe and explain the central novel contributions of the paper
- Show and discuss results
  - e.g. animations, graphs, comparisons to theory or experiment, etc.
- Provide a critique of the paper (both good and bad)
- Conclude briefly

Presentations - Tips

- Don’t explain every tiny detail – focus on core/novel contributions
- Prefer diagrams and images (and your voice) over lots of supertiny text
- Avoid overwhelming the audience with *too many* equations
- Talk to the audience, not the slides.
- Do not just recycle the authors’ slides if they exist. (Borrowing figures, graphs, results is fine.)
- Practice!

Discussions

- Following each presentation, we'll have ~15-20 mins for questions/discussion to dive further into technical details, clarify any confusion, debate the merit of the work, etc.

- Everyone should read the papers and bring comments/questions/critiques. Bring a PDF or print-out to refer to.

- Since we will all read the papers, goal is *not* (necessarily/strictly) to grill the presenter, but rather to discuss as a group.

- Some classes will have no presenter; just (longer) discussions.

Course Project

- Pick a method or technique for a physical system, implement it, and demonstrate its use.
- Should be non-trivial, but need not (necessarily) be novel.
- Solo or with a partner.
- Can rely on existing code/libraries, but must be documented.

- I am happy to try to arrange 1:1 meetings to discuss projects, at any point in the process.

Course Project –2014 edition projects

- Multiple liquid (SPH) simulation
- Rigid bodies with magnetic interaction forces
- Cloth with collisions
- Finite element deformable objects
- Spray/foam simulator

Some quite nice examples from similar courses:

Liquid: http://www.yiningkarlli.com/projects/arielflip/

Rigid bodies: https://benedikt-bitterli.me/rbs.html

Course Project: Deliverables

- 1-2 page project proposal – due Feb. 12 at latest. But the sooner you start the better!
- Short presentation & demo during the last week of class.
- Final submission – tentatively due April 15. (Won't be earlier.)
  - Final report (PDF) in SIGGRAPH paper format describing what you achieved and how.
  - An animation clip illustrating the project results.
  - The associated code.

Paper Reviews

- Pick **one** of the papers to be presented/discussed each week.
- Write a “SIGGRAPH-style” review of the paper. A LaTeX form is posted on the course page. Expected length is about a page or so.
- Due Sunday at 5pm prior to the week of the associated paper presentation/discussion.

See SIGGRAPH review format here:

http://s2015.siggraph.org/submitters/technical-papers/review-form

Analyzing a paper

Imagine you are a reviewer deciding whether to accept or reject…

Questions to ask yourself:

- Did the authors clearly motivate why the problem is relevant/important?
- Are the contributions truly novel wrt. previous work?
- How *substantial* are the contributions?
- Why did the authors make the [technical/design/theoretical/implementation] choices they did? Are they justified?
- Do the results actually achieve/support the paper’s claims?
- Are the writing and figures clear?

Reading/Reviewing Tips

MIT's Fredo Durand has some tips for reviewing papers:

- http://people.csail.mit.edu/fredo/review.pdf

Keshav offers some great strategies for reading papers (and more good references):

http://blizzard.cs.uwaterloo.ca/keshav/home/Papers/data/07/paper-reading.pdf

Introductions

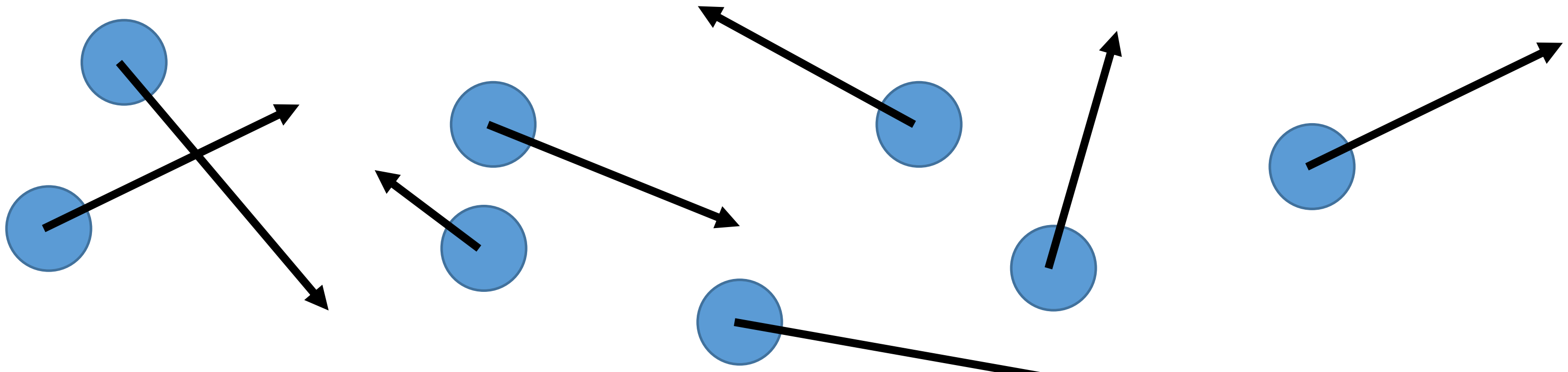
Questions for you…

1. What level of computer graphics courses have you taken (at Waterloo or elsewhere)?
2. What level of scientific computing / numerical methods courses have you taken?
3. Summarize any other relevant background or experience.
4. What topic(s) are you most/least interested in?

Particle Systems

Particle Systems

Particle system: A collection of point particles that obey rules dictating their creation, movement, deletion, and other attributes and behaviors.



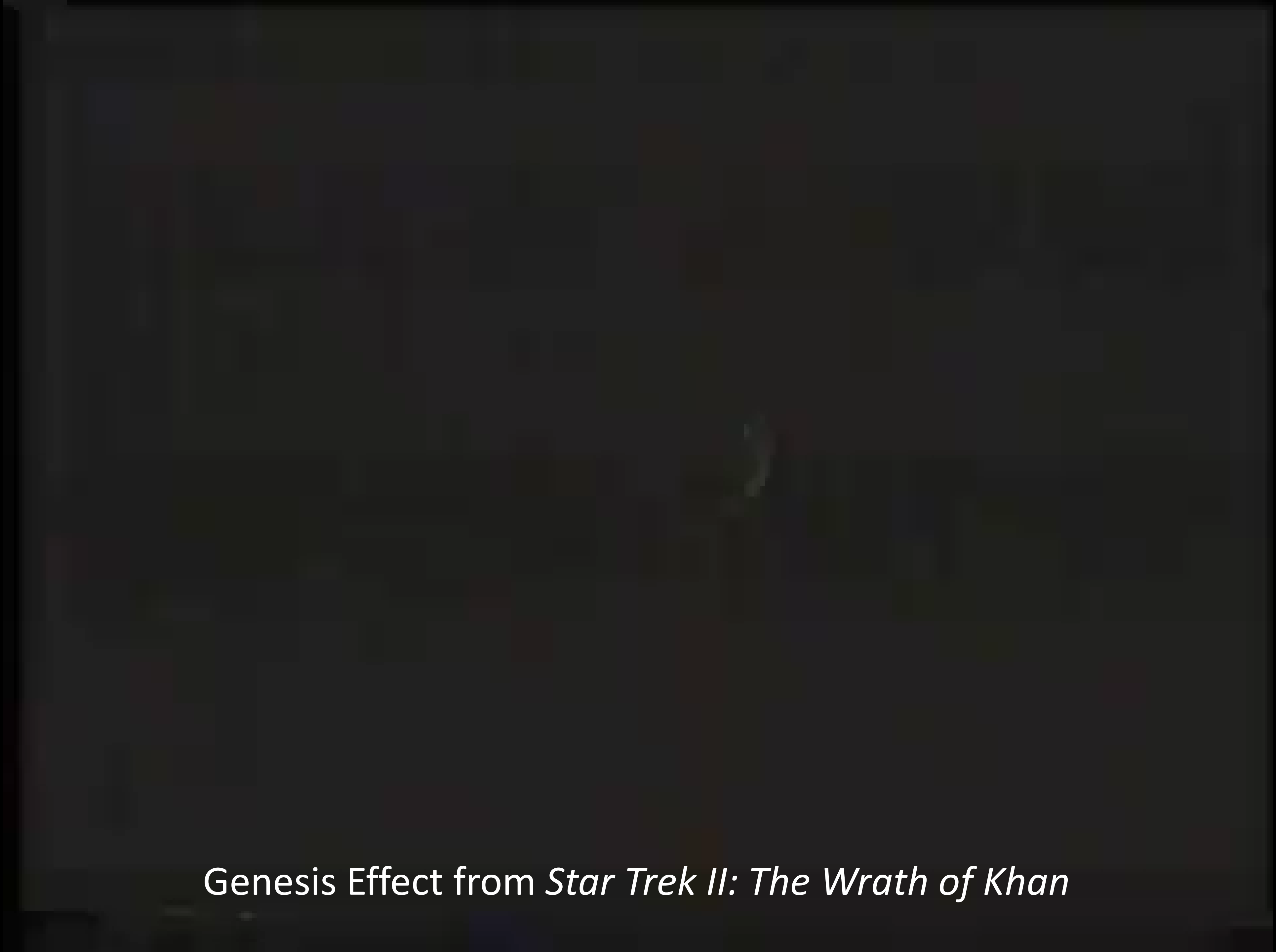
Particle Systems

- Often used for ad hoc modelling of “fuzzy”/complex phenomena, with…
  - ill-defined or changing boundaries
  - chaotic motion
  - e.g., fire, waterfalls, dust, clouds, flocking animals, etc.

- Common in 3D software (Maya, 3DSMax, Houdini, etc.)

Classic examples:

- 1982’s “Star Trek II: The Wrath of Khan”, modeling a fiery explosion transforming a planet.
- Karl Sims’ 1988 animation “Particle Dreams”.



Genesis Effect from *Star Trek II: The Wrath of Khan*

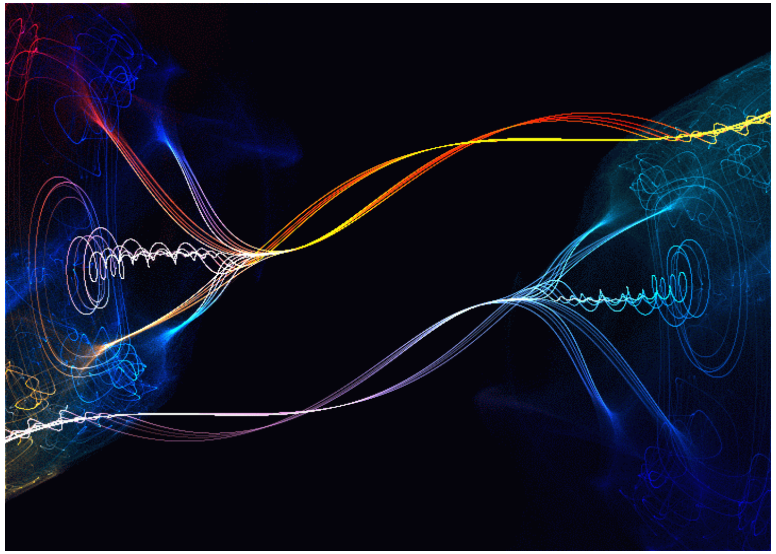
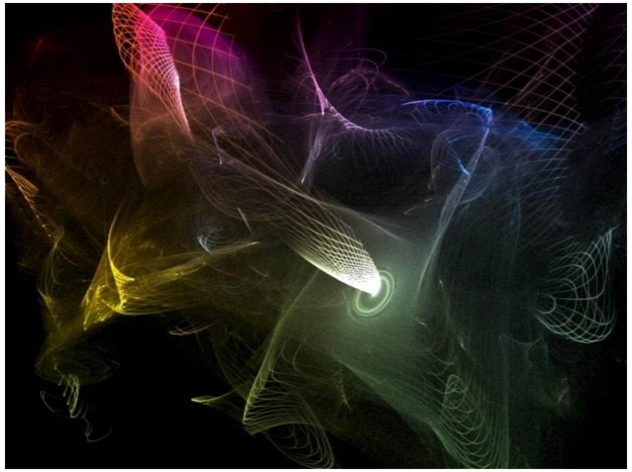
PARTICLE DREAMS

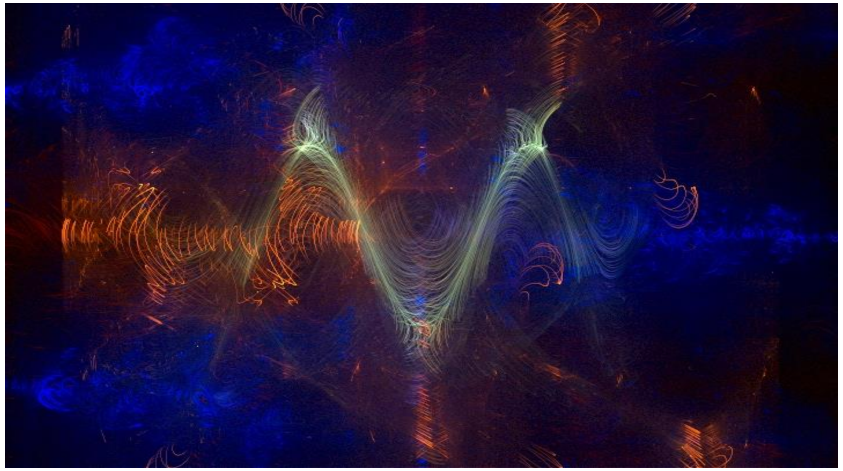
Karl Sims

Optomystic

More Recent Example – "Spore"



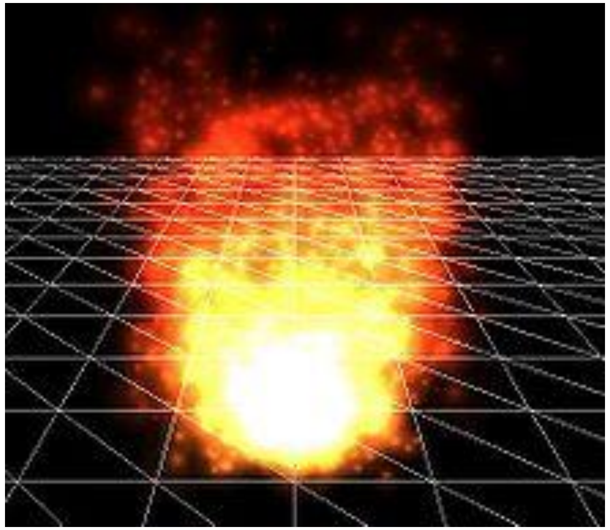


http://www.imagesavant.com/

Particle Systems

Possible particle data/attributes:

- Position (x,y,z)
- Velocity (x,y,z)
- Orientation
- Mass
- Color
- “State”
- Age
- Temperature
- etc. (whatever else you like!)



Particle Systems

At each frame of animation:

- Create new particles and assign initial attributes.
- Update existing particle position/velocity/attributes according to chosen rules.
- Delete “expired” (old) particles.

Rules can also incorporate some randomness.

Designing the rules is where the art (and maybe science) comes in.

Example: Star Trek II: Genesis Effect

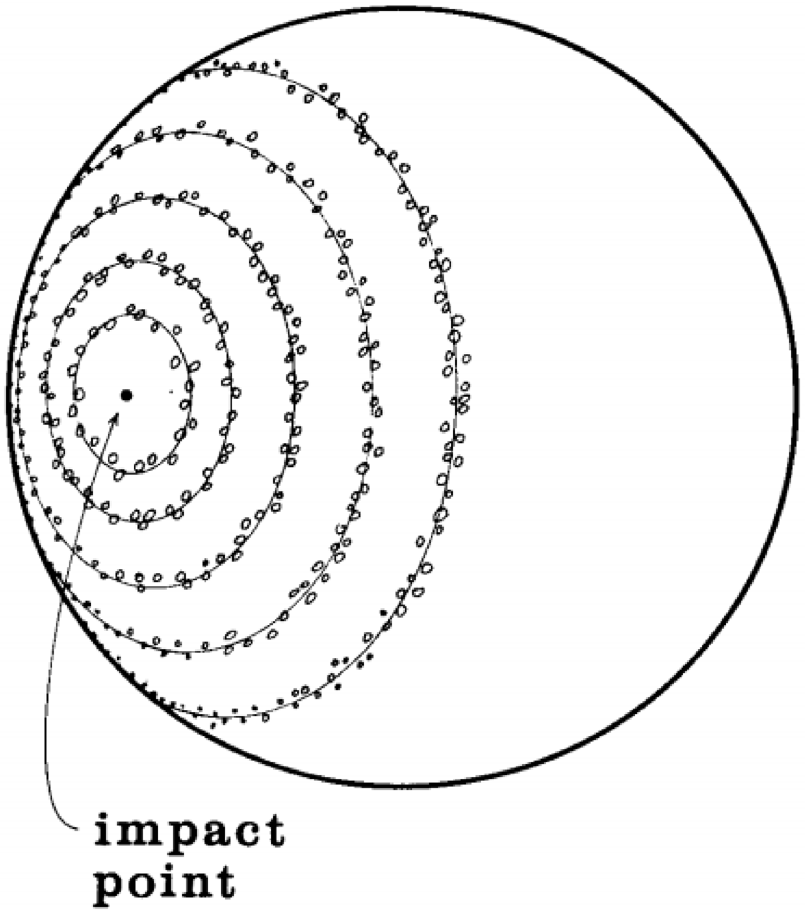


Fig. 2. Distribution of particle systems on the planet's surface.

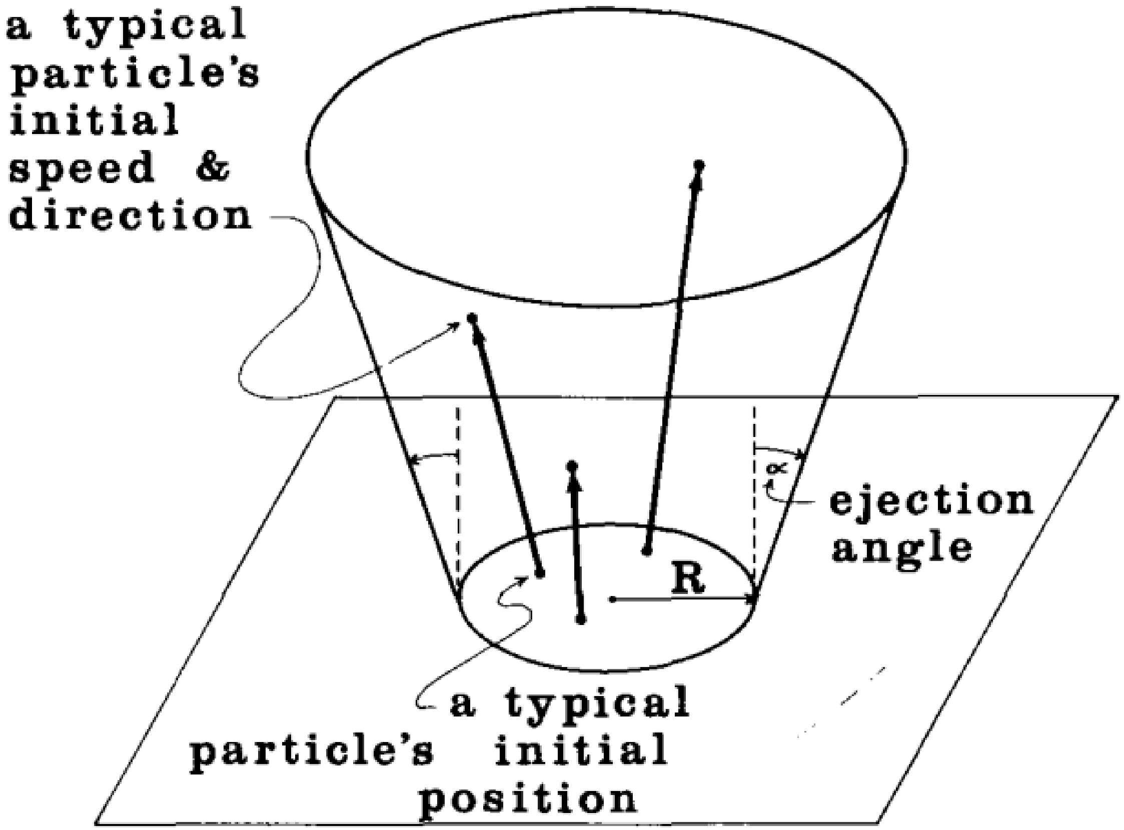


Fig. 3. Form of an explosion-like particle system.

From [Reeves 1983]

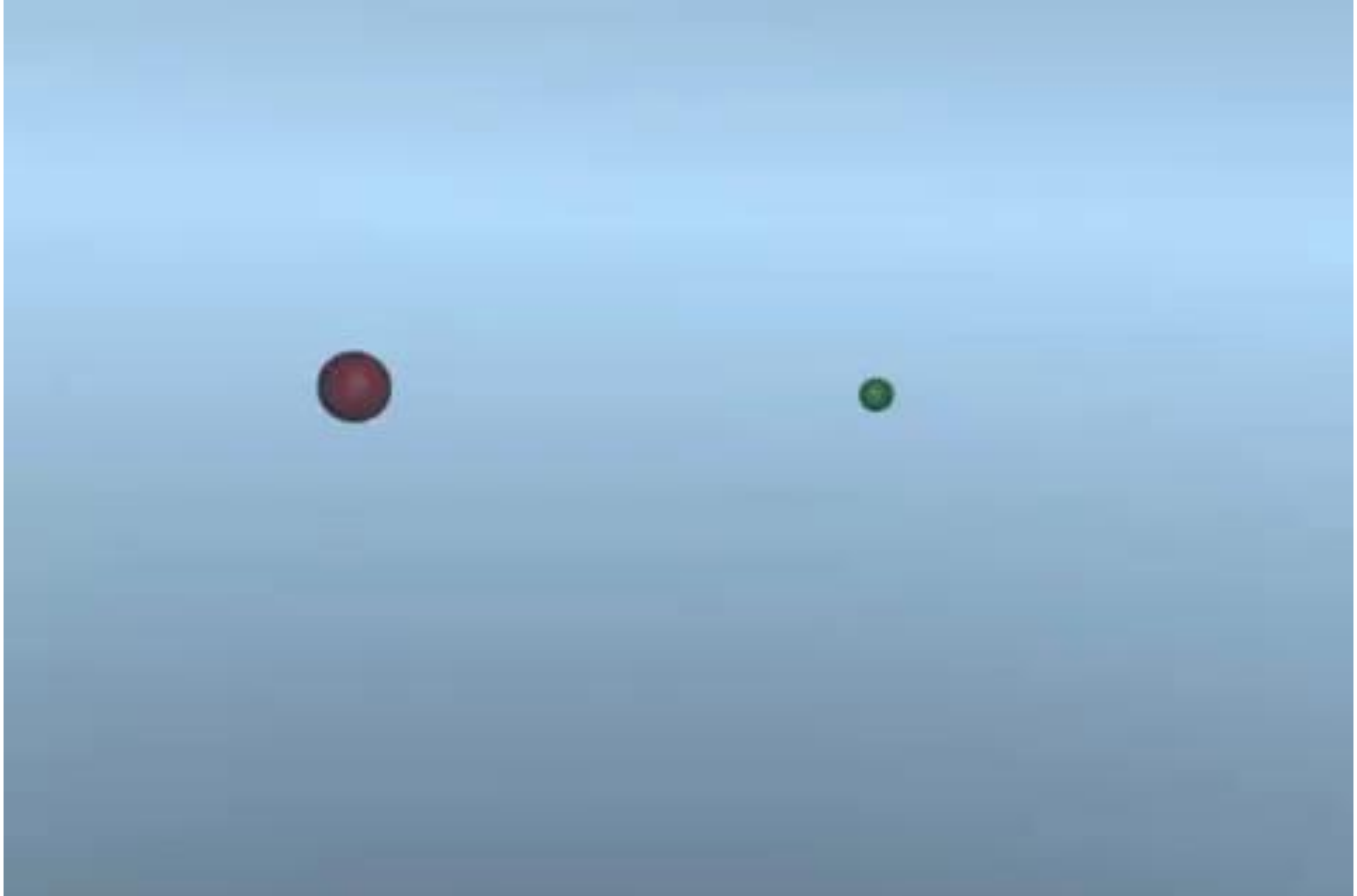
Flocking (“Boids”)

Simple rules relating to interactions between nearby particles can yield emergent, flocking-like behaviour.

- Collision Avoidance (separation)
- Velocity Matching (alignment)
- Flock Centering (cohesion)

For details see:

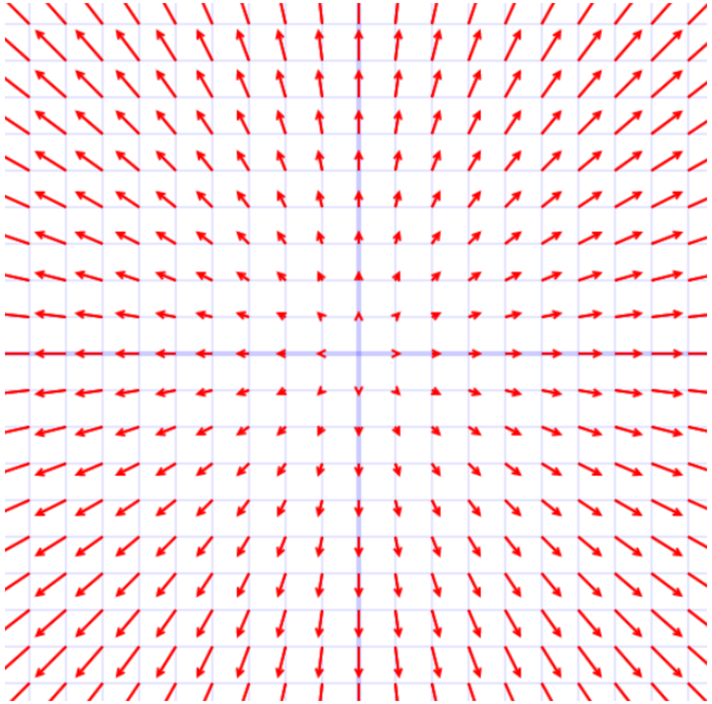
“Flocks, herds and schools: A distributed behavioral model.” [Reynolds, 1987]



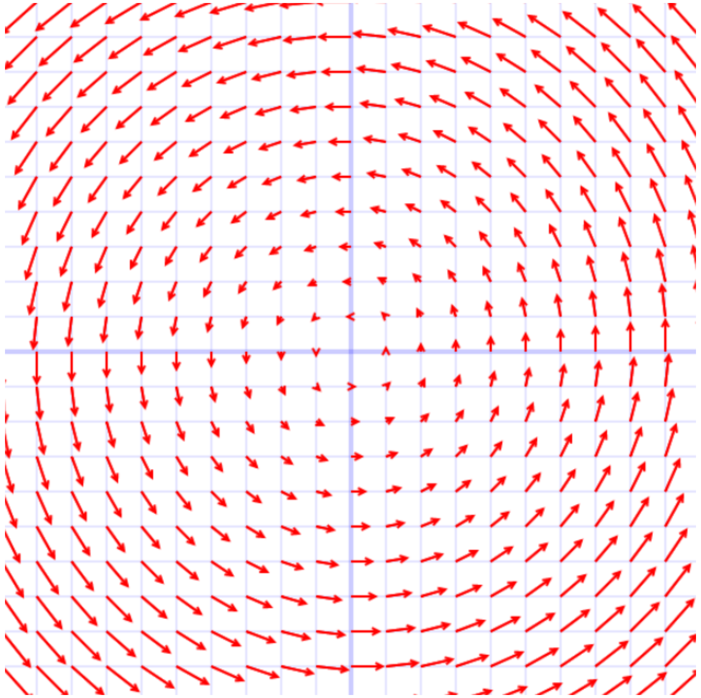
Simple Flocking Animation in 3D

Vector Fields

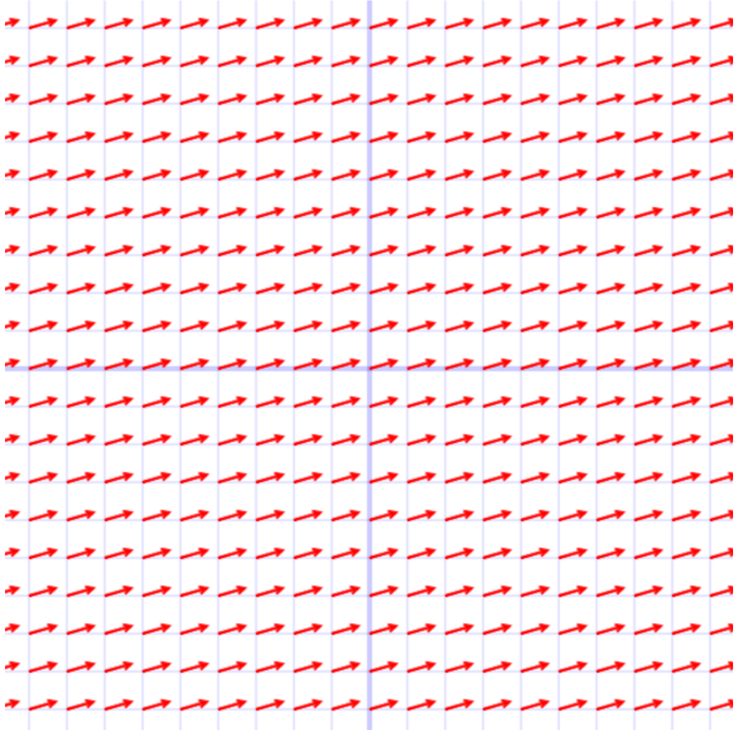
Particle motion can also be given/affected by a function that takes a 3D position and returns a 3D velocity, i.e., a *vector field*.



Radial expansion: $V = (x, y)$

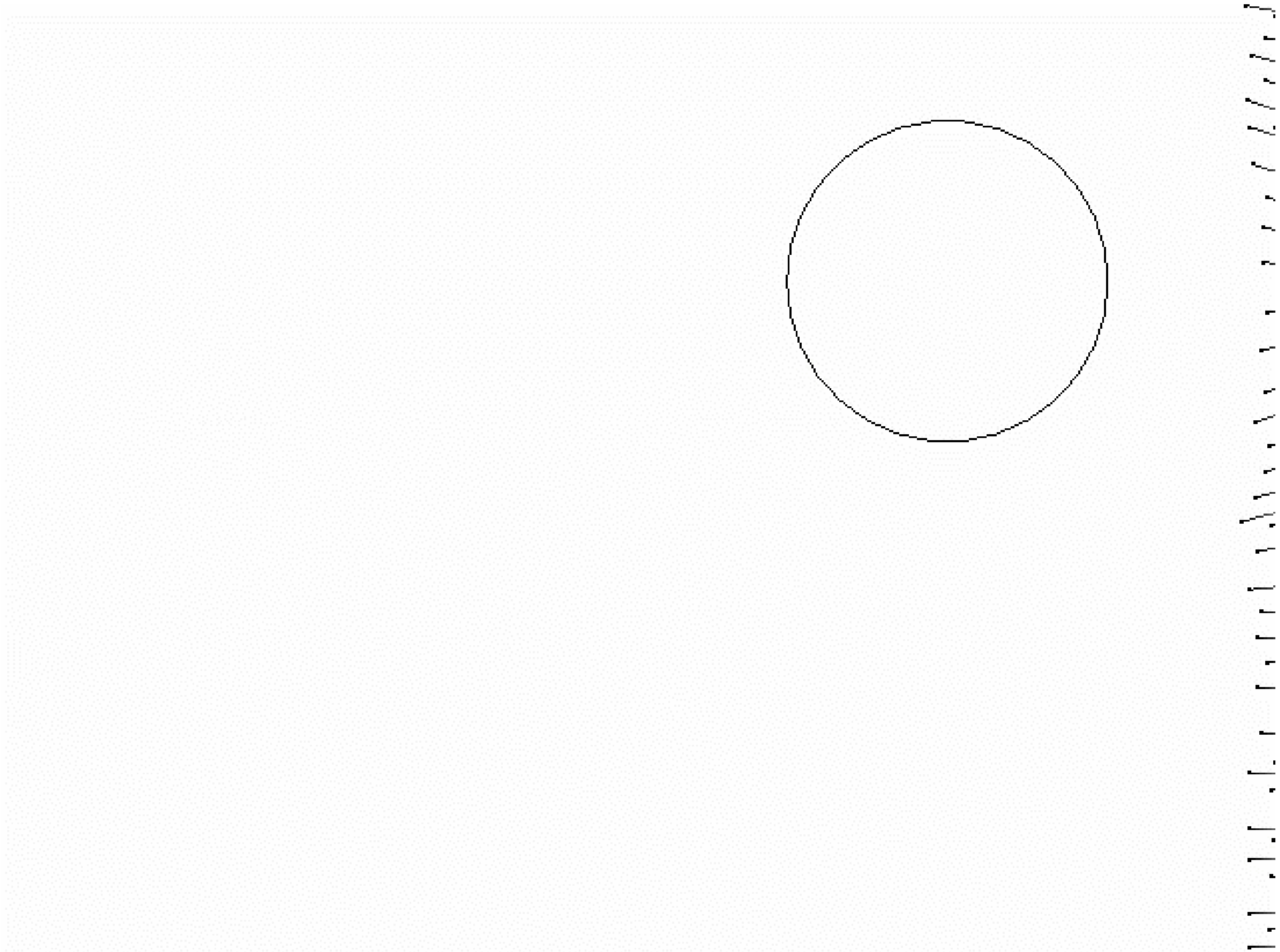


Rotational Vortex: $V = (-y, x)$



Constant Wind: $V = (7,2)$

Particles driven by a vector field



Basic Time Integration

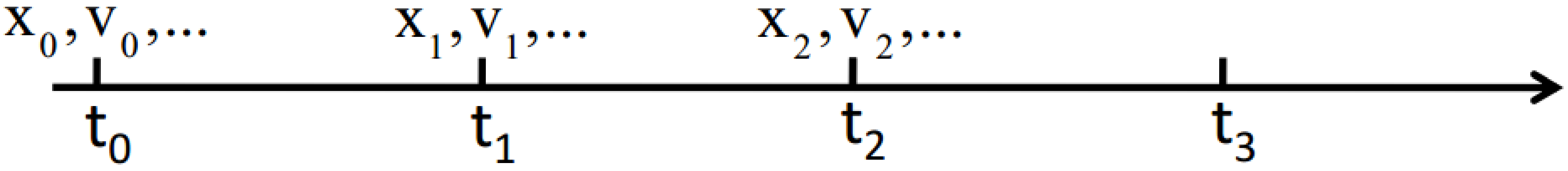
Solving For Particle Motion

Given a particle P at time T=t with:

- Current position X=(x,y)
- Velocity function V(X,t) = (u,v)

…how do we determine the new particle position at time T = t+Δt?

This task is called *time integration*. Δt is the time step.



Time Integration (for 1st order dynamics)

Recall: velocity V is the time derivative of position X.

i.e., rate of change of the particle position with respect to time.

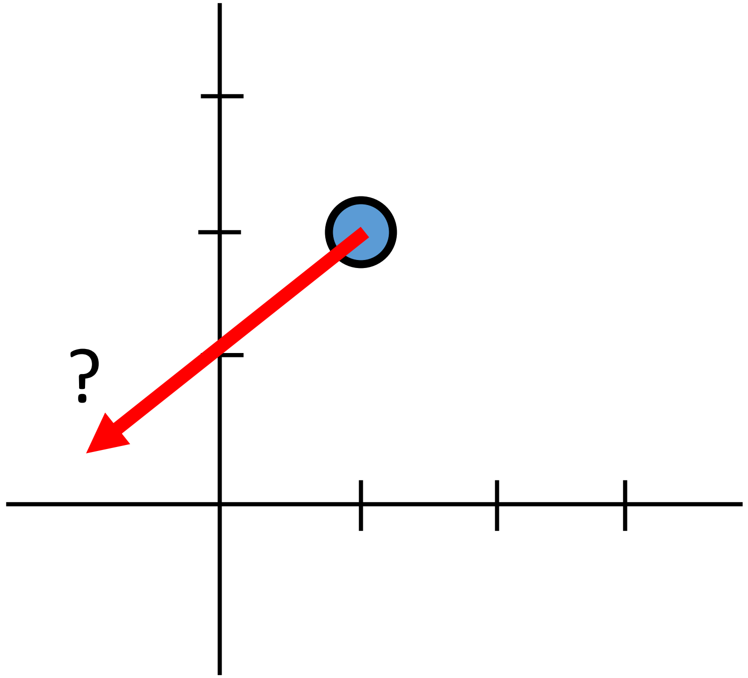
$$\frac{dX}{dt} = V$$

This is a *differential equation* relating X and V by a (time) derivative.

Given V and initial values for X, solve for X at subsequent times.

Time Integration

e.g., consider a particle with current position X = (1,2) and given (constant) velocity V = (-1,-1) m/s, taking a *time step* of length Δt = 0.5 s.



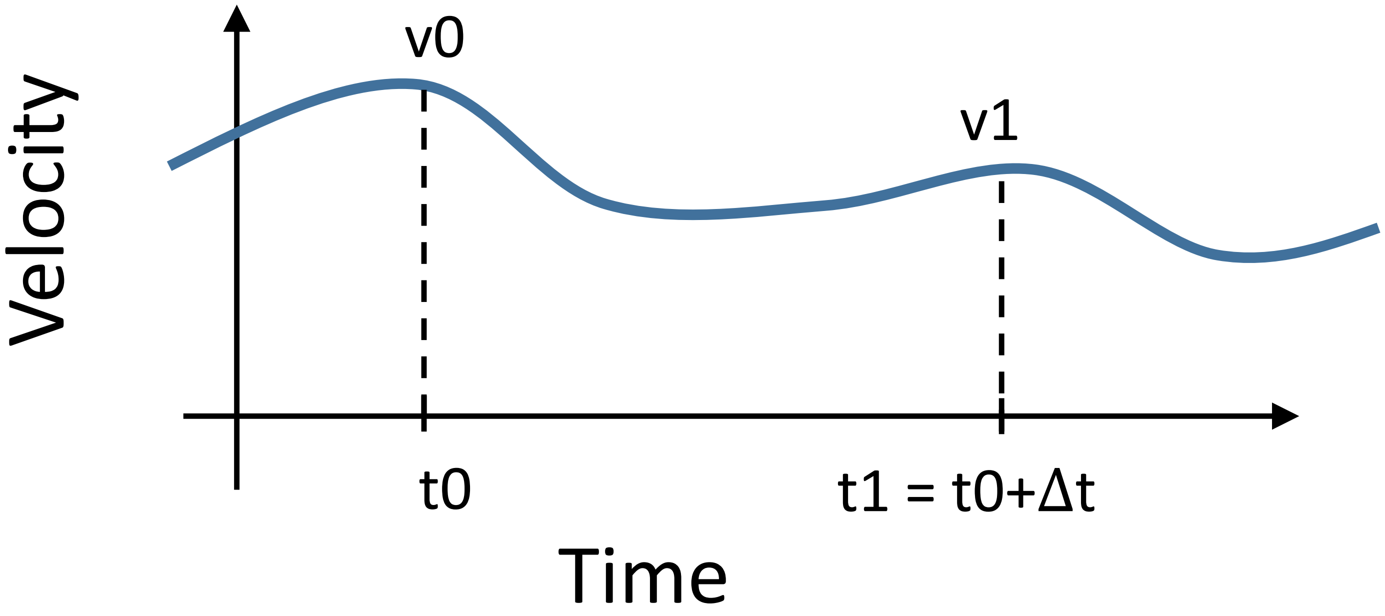
$$X^{t+\Delta t} = X^t + V\Delta t$$

Solution: $X^{t+\Delta t}$ = (0.5, 1.5)

Time Integration (1D)

Finding the new position requires *integrating velocity over time*.

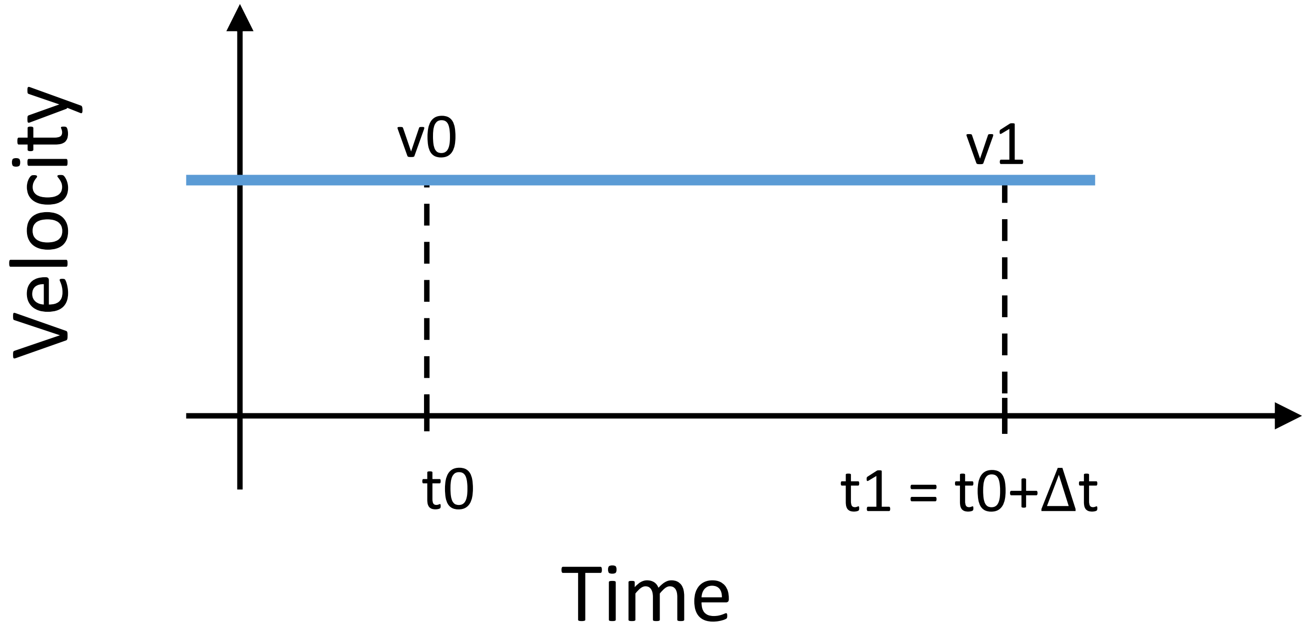
$$X^{t+\Delta t} = X^t + \int_t^{t+\Delta t} V dT$$



i.e., find the area under this curve.

Time Integration (1D)

In our example, V was a constant, so the (rectangular) area was *exactly* $V\Delta t$.



$$X^{t+\Delta t} = X^t + V\Delta t$$

What About *Time-Varying* Velocity?

Velocity function could depend on many factors, including current time, position, “state”…

$$\frac{dX}{dt} = V(t, X(t), \dots)$$

e.g., $V = (17t\log(t)\tan(y), \operatorname{arcsinh}(t)^t x^2)$.

In general, ***we can’t solve the integral exactly***. We must approximate.

Numerical Integration

We will use *numerical integration*.

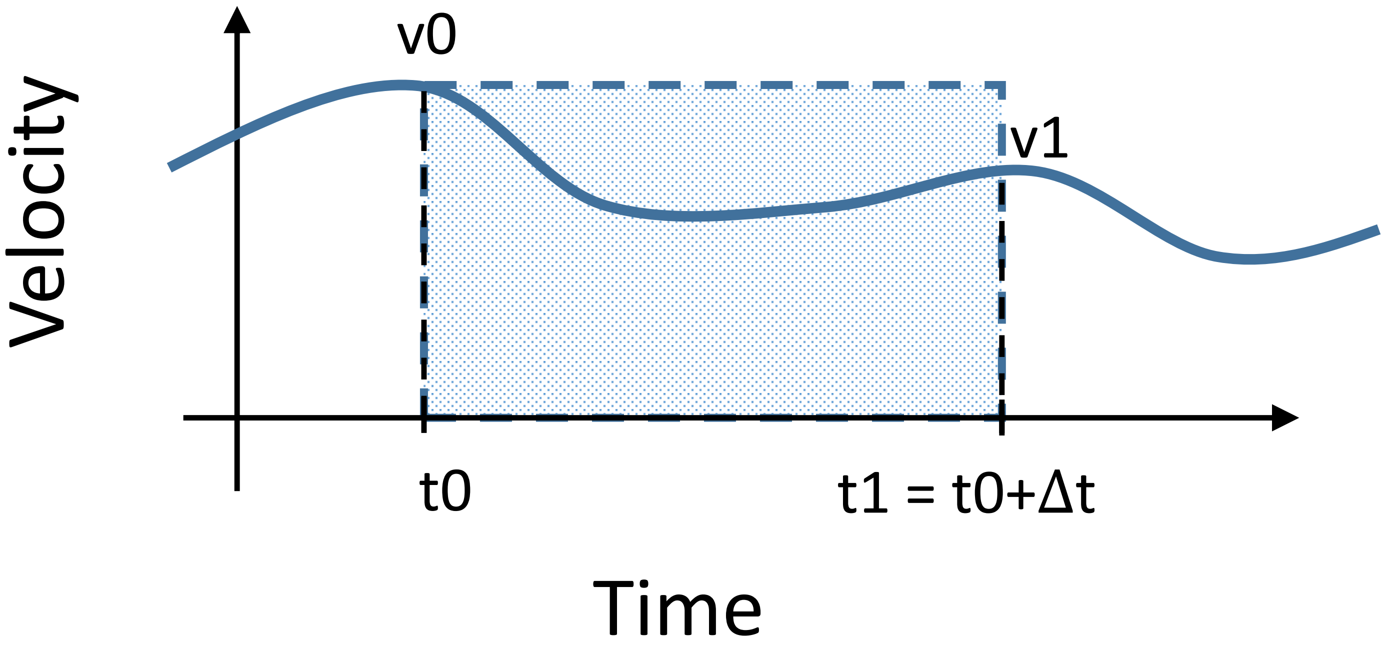
Simple idea: “Ignore” that the velocity may change *during* the time step. Then…

$$X^{t+\Delta t} = X^t + V(t)\Delta t$$

i.e., Evaluate V at the **current time** t, and use it to take only a single step. Repeat on the next step.

Numerical Integration

This approximates the true area as a rectangle, using the starting velocity, v0.



Forward Euler

This simple scheme is called Forward Euler.

$$X^{t+\Delta t} = X^t + V(t, X^t)\Delta t$$

Example:

- X(t=0) = (0, 1)
- V = (-y, x),
- Δt=0.5

Find X(t=1.5).

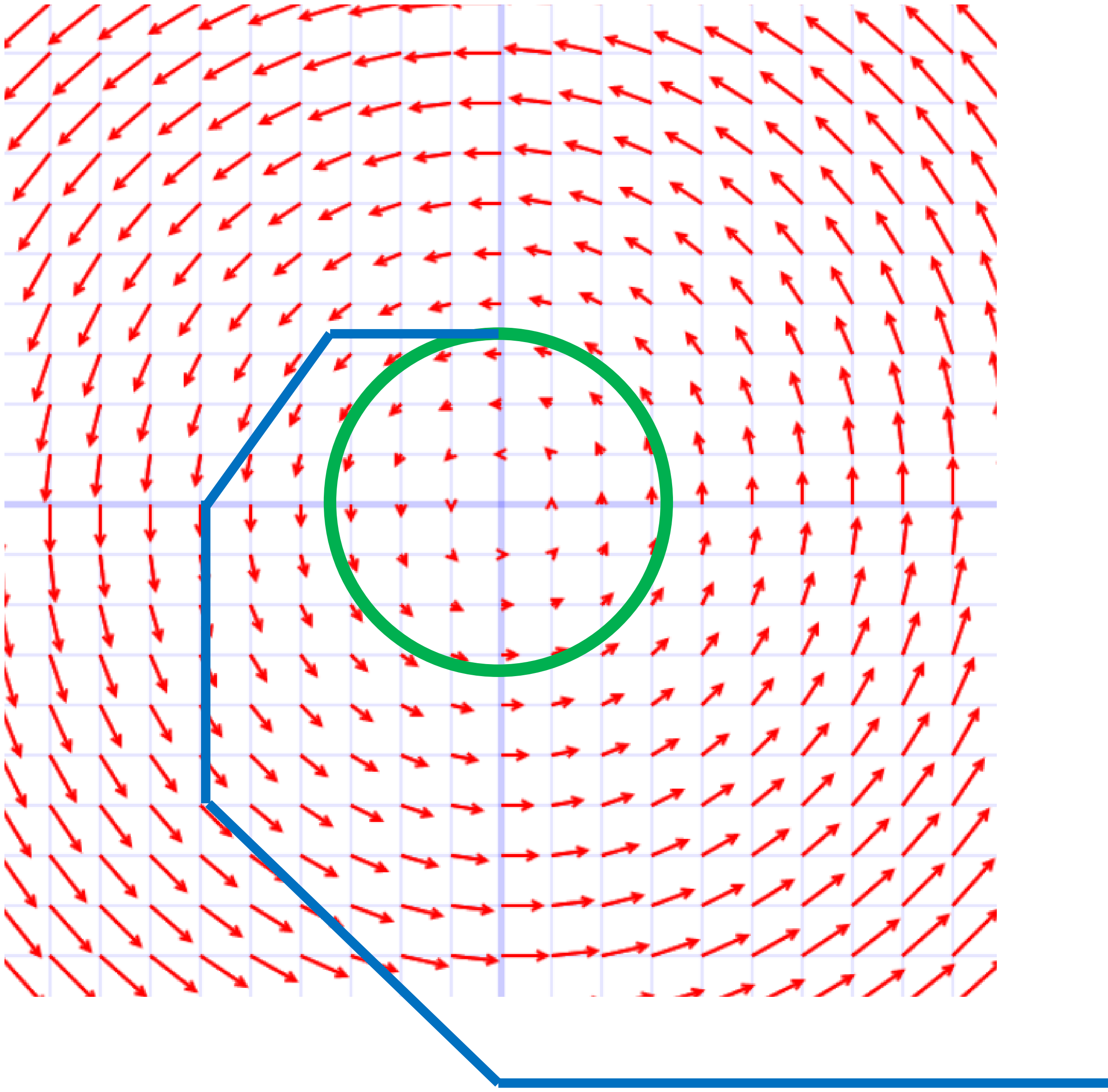
X(0.5) = (0,1) + 0.5(-1,0) = (-0.5, 1)
X(1) = (-0.5, 1) + 0.5 (-1, -0.5) = (-1, 0.75)
X(1.5) = (-1,0.75) + 0.5(-0.75,-1) = (-1.375, 0.25)
…

Vector Fields

This is the vector field V = (-y,x).

Compare the expected true trajectory (green) to the behaviour of our numerical solution (blue) with forward Euler…

Lots of drift!

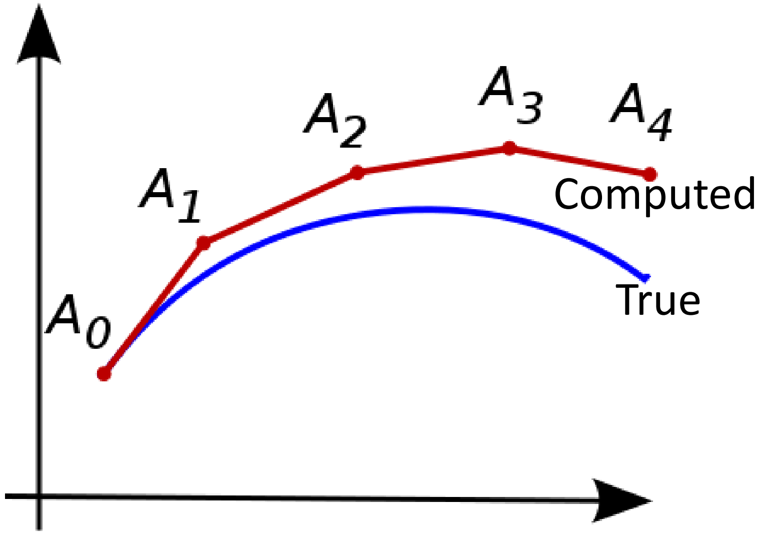


Forward Euler – Points to Note

1. Accumulated error can cause the numerical solution to drift away from the true solution.

2. But, the smaller the time step Δt, the more accurate the approximate trajectory becomes.

3. If the time step is too large, the result can "blow up" and yield garbage answers. Forward Euler has a (problem-dependent) maximum *stable* time step…



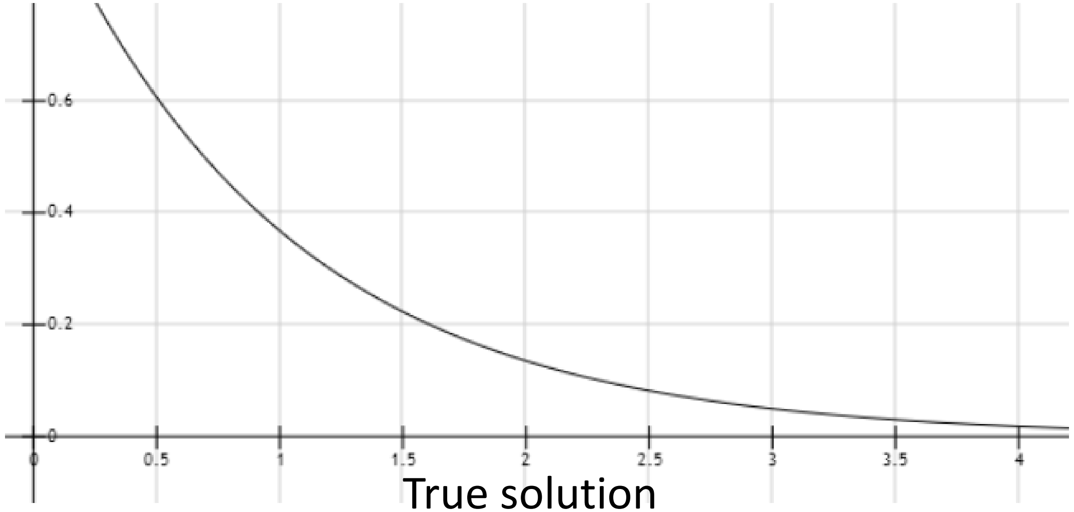
Forward Euler – Instability

Consider the 1D function: $\frac{dx}{dt} = -x$, with x(t=0) = 1.

True solution is: e^{-t}

Always positive, decays smoothly.

Numerical solution for $\Delta t = 3$?



True solution

Result: 1, -2, 4, -8, 16, etc.

Wrong!

The sign flips madly, the magnitude increases instead of decreasing.

Other Time Integration Schemes

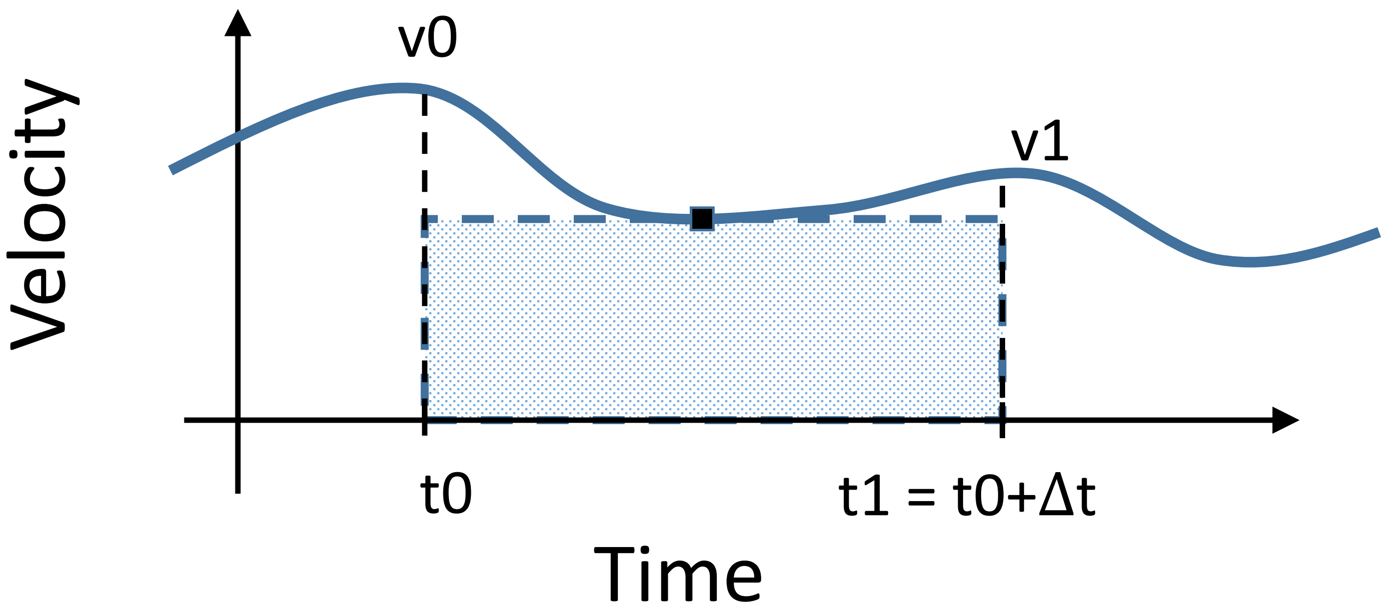
Forward Euler uses the velocity at the *start* of a time step to perform the integration.

Other common schemes use the (possibly approximate) velocity at the *middle*, *end*, and/or *other* instants to increase accuracy and stability.

e.g. midpoint method, trapezoidal rule, implicit Euler, Runge Kutta schemes, etc.

(Explicit) Midpoint method

Use the approximate velocity at the time step *midpoint* to estimate the integral. (AKA 2^{nd} order Runge Kutta or RK2).



(Explicit) Midpoint method

First, estimate the midpoint position halfway through a time step:

$$X^{mid} = X^t + \frac{\Delta t}{2} V(t, X^t)$$

Then, use the velocity evaluated at the midpoint to determine the final position.

$$X^{t+\Delta t} = X^t + \Delta t V\left(t + \tfrac{1}{2}\Delta t, X^{mid}\right)$$

E.g., Try the FE and RK2 on the circular velocity V = (-y,x).

(More on time integrators next time…)

Adding Some Physics

Newton's 2^{nd} Law

Rather than prescribe velocities, we often want to use physics (classical mechanics) to solve for *both* X *and* V, given a set of applied forces, F.

First, assign each particle some fixed mass, M.

Then, recall Newton's 2^{nd} law: Force = Mass x Acceleration.

2^{nd} order dynamics

Earlier, we had a given velocity, V, dictating how we update position X.

$$\frac{dX}{dt} = V$$

Now, we instead have given forces, F, and Newton's 2^{nd} law, $F = ma$.

Acceleration is the 2^{nd} time derivative of position X, so we have a 2^{nd} order differential equation…

$$m\frac{d^2X}{dt^2} = F$$

2^{nd} order dynamics

We can split this into two 1^{st} order equations…

$$m\frac{d^2X}{dt^2} = F \quad \Longrightarrow \quad m\frac{dV}{dt} = F, \qquad \frac{dX}{dt} = V.$$

Time integrate each of these (e.g. via forward Euler, midpoint, etc.) to evolve the system.

Forward Euler, revisited

Position Update:

$$\frac{dX}{dt} = V \quad \Longrightarrow \quad X^{t+\Delta t} = X^t + V^t \Delta t$$

Velocity Update:

$$m\frac{dV}{dt} = F \quad \Longrightarrow \quad V^{t+\Delta t} = V^t + \frac{F(t, X^t)}{m}\Delta t$$

Forces

What physical forces might we use to drive a particle system?

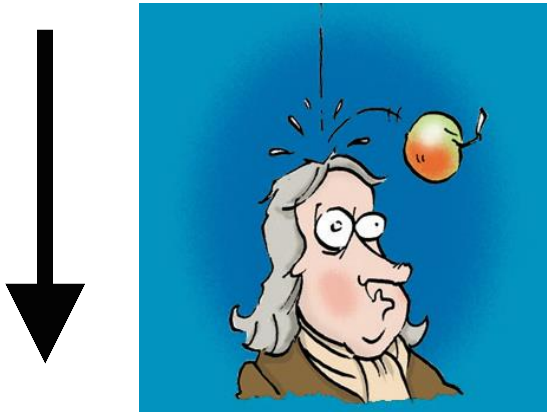
- Gravity
- Wind / Air drag
- Springs (between particles!) / Elasticity
- Damping / Viscosity
- Friction
- Collisions/Contact
- Magnetism
- “Control” / User
- …

Given the set of forces $F_1, F_2, \ldots, F_n$, sum up to get net force on a particle.

Forces: Gravity

1. Earth-specific gravity (treated as a constant):

$$F = (0, -9.81, 0)\ m/s^2$$

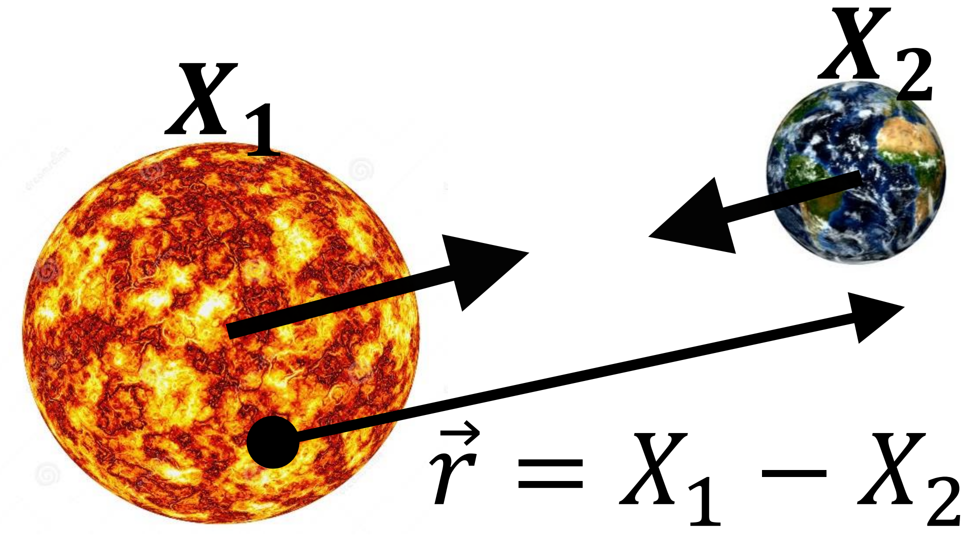


Simple example: An initially stationary apple at a height of 10m falls under gravity, with time steps Δt of length 0.1 seconds. Apply time integration to estimate its impact time.

Forces: Gravity

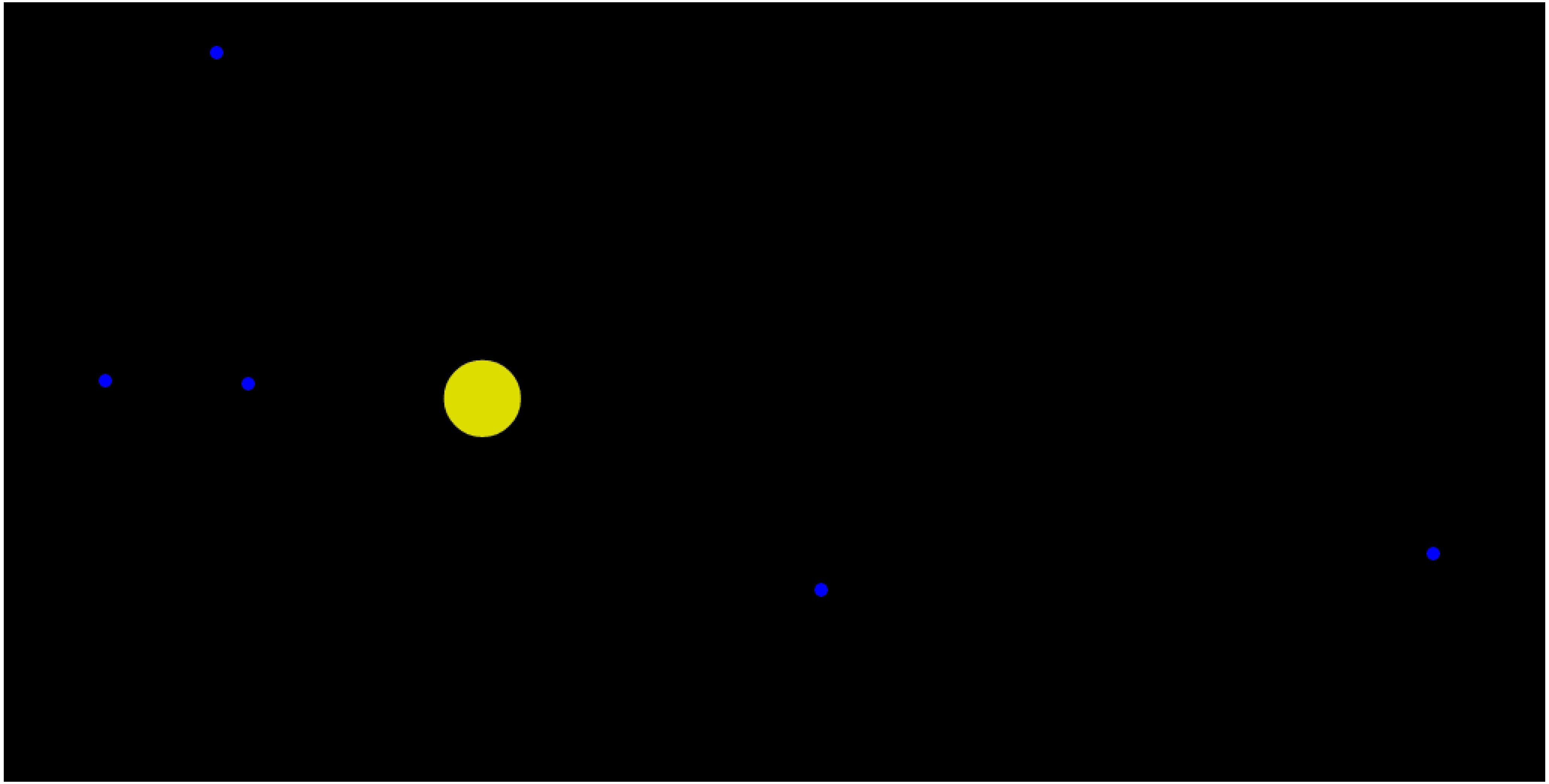
2. N-body gravitation:

$$F = -\frac{Gm_1 m_2}{\|\vec{r}\|^3}\vec{r}$$



Gravitation Simulation

http://wxs.ca/js/jsgravity/



Forces: Springs!

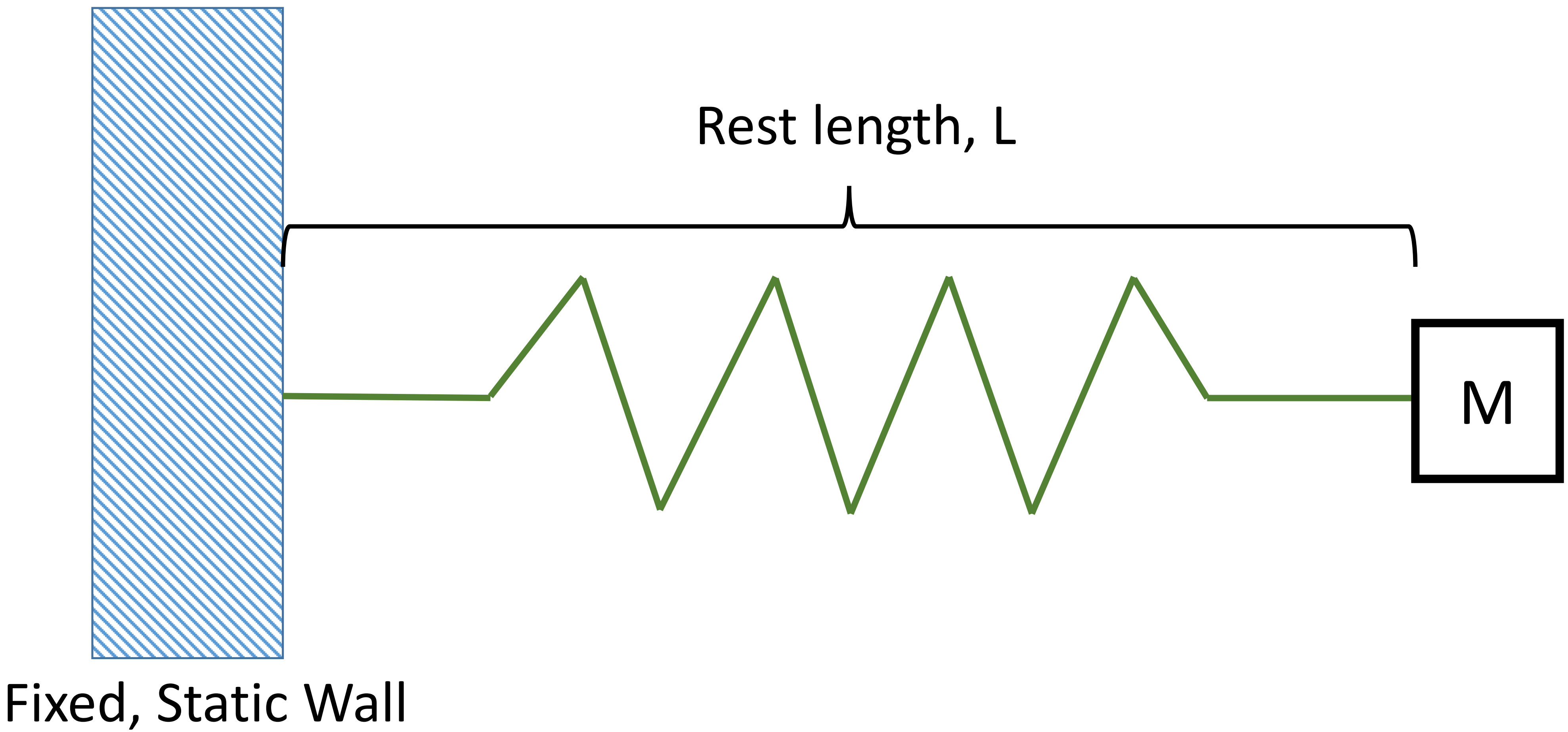
A very simple way to model complex structures (e.g., hair, cloth, jello) is connecting particles with *spring forces*.

AKA mass-spring systems.

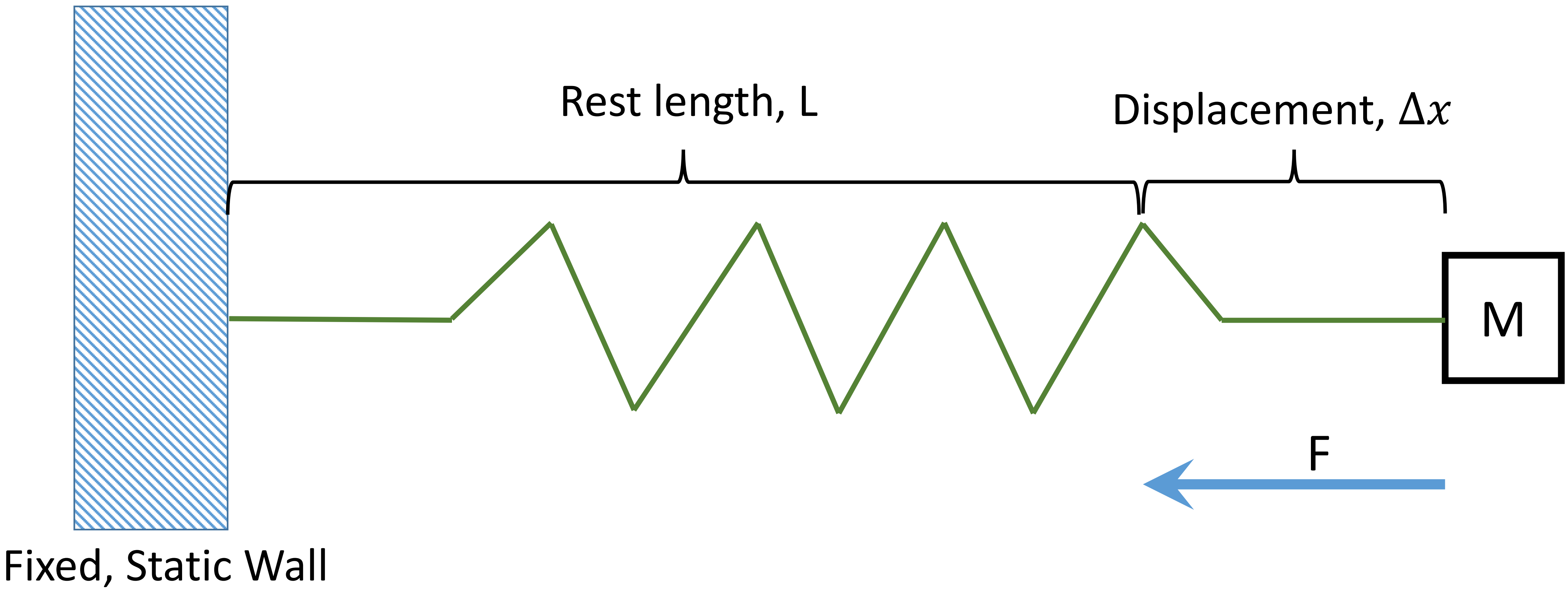
Each spring…

- connects two particles.
- has a given *rest length, L.*
- has a given “spring constant” or *stiffness coefficient, k.*

Spring in 1D



Spring in 1D



Hooke's Law for linear springs

The restoring force…

- Is linearly proportional to the amount of displacement (from the rest length).
- Acts in the opposite direction to the displacement:

$$F = -k\Delta x$$

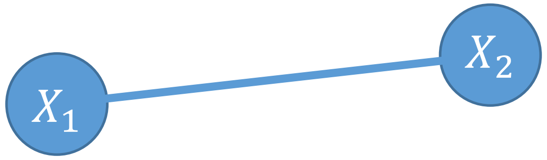
where *k* is the proportionality constant that controls the spring stiffness.

Stiffer materials typically require smaller timesteps for **stability**!

Hooke's Law for 3D springs

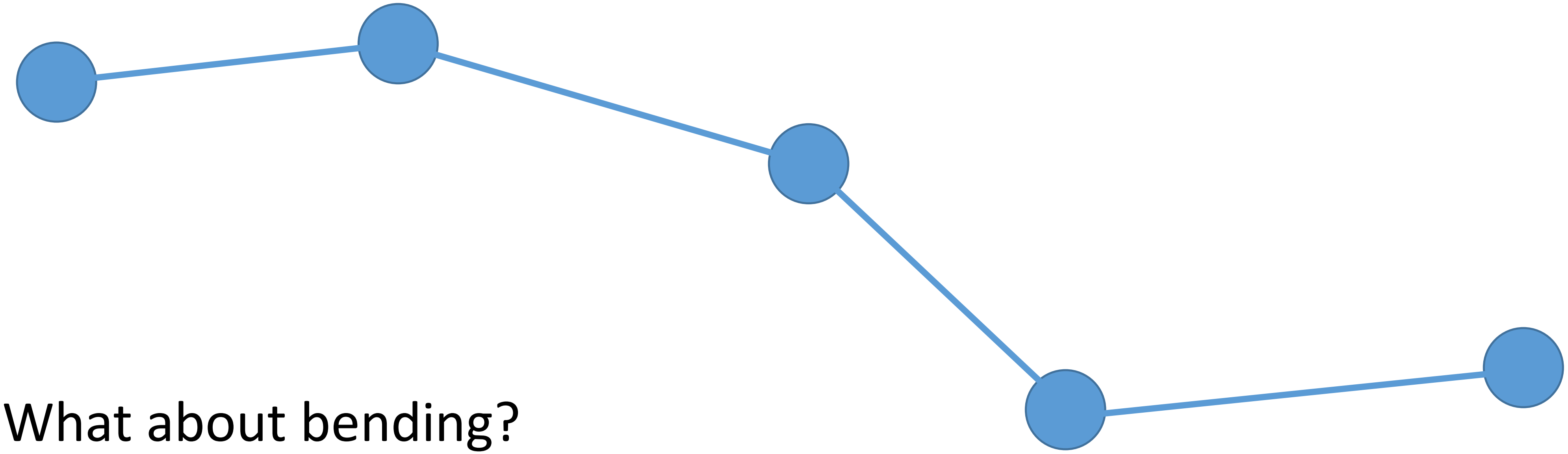
For a spring joining 2 particles with position vectors X_1 and X_2:

$$F_1 = -F_2 = -k\underbrace{(\|X_1 - X_2\| - L)}_{\text{Displacement}}\overbrace{\frac{X_1 - X_2}{\|X_1 - X_2\|}}^{\text{Direction}}$$



Springs for Hair and Cloth

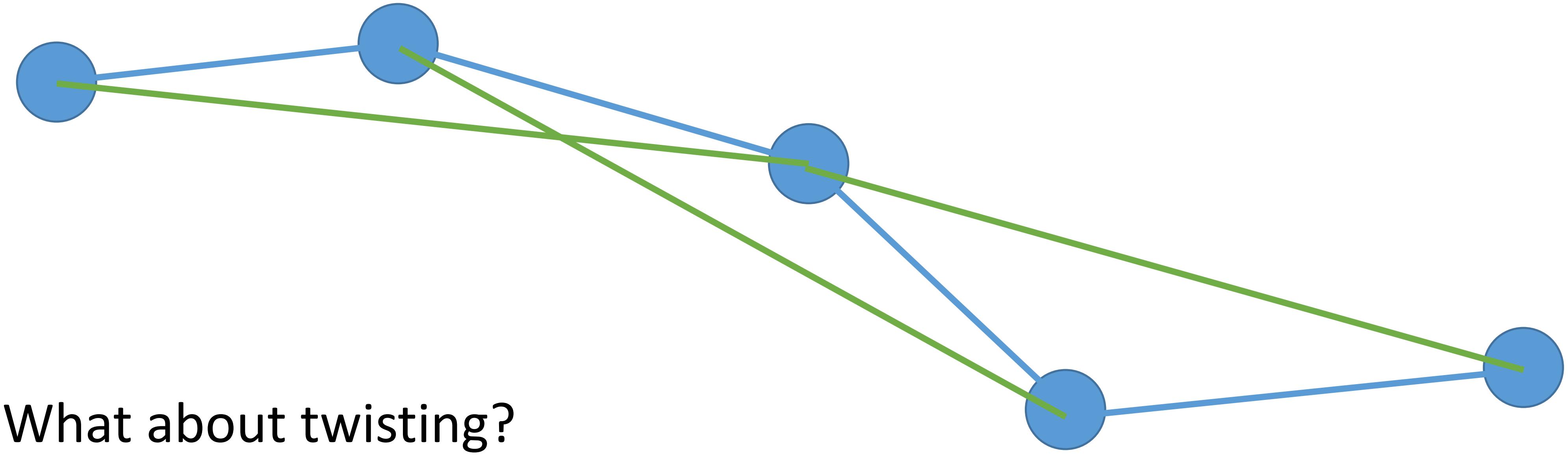
A single chain of masses and springs can model a strand of hair.



What about bending?

Springs for Hair and Cloth

Add *alternating* springs. This discourages the hair from collapsing when you bend it.



What about twisting?

See "A Mass Spring Model for Hair Simulation" [Selle et al. 2008]

[Selle et al. 2008]

Summary

- Particle systems can model diverse phenomena, in non-physical and physical ways.

- Time integration methods advance a simulation through time
  - e.g. forward Euler, midpoint, etc.

- By solving the equations of motion for particles and particle systems, we can capture more physically meaningful behaviours.

- Remember: **Email me your top 3 preferred slots for the first round of presentations by Friday noon.** (No guarantees.) First slot is Jan 18.