



Passive Dynamics and Particle Systems

COS 426

Computer Animation

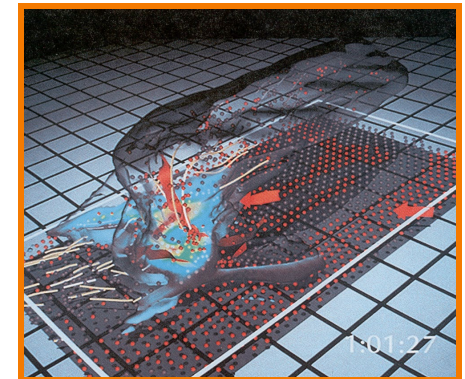


- Animation
 - Make objects change over time according to scripted actions



Pixar

- Simulation / dynamics
 - Predict how objects change over time according to physical laws

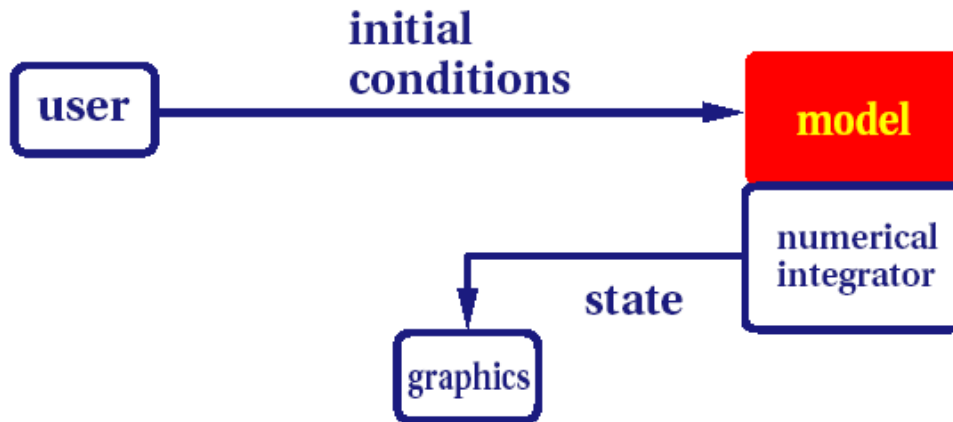


University of Illinois

Dynamics

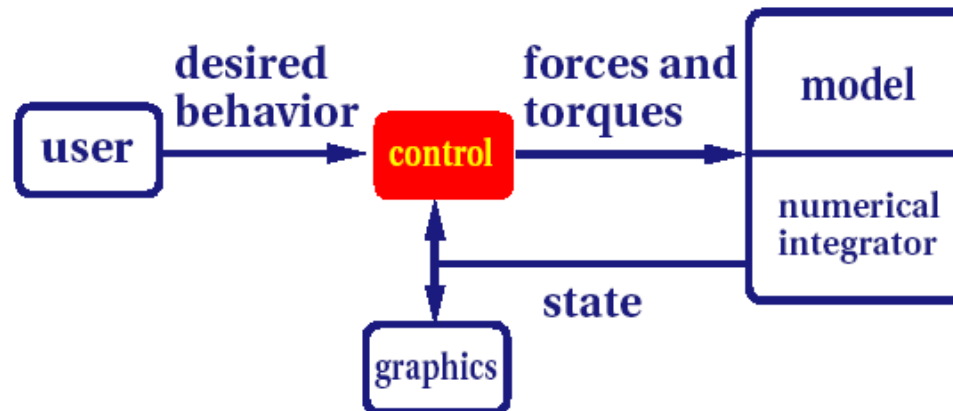


Passive--no muscles or motors



particle systems
leaves
water spray
clothing

Active--internal source of energy

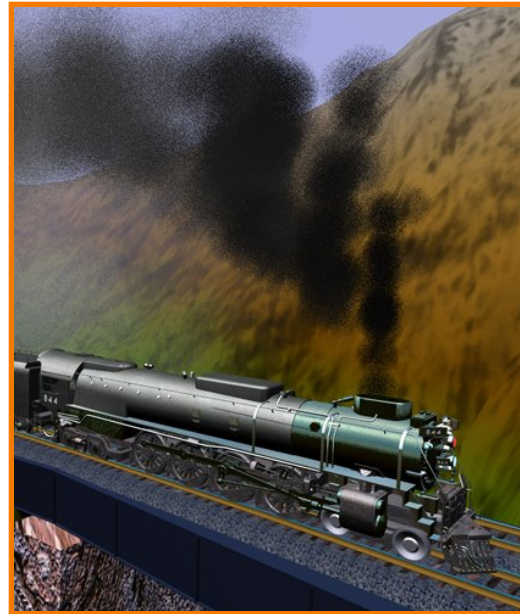


running human
trotting dog
swimming fish

Passive Dynamics



- No muscles or motors
 - Smoke
 - Water
 - Cloth
 - Fire
 - Fireworks
 - Dice



Passive Dynamics



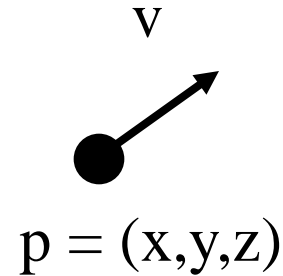
- Physical laws
 - Newton's laws
 - Hooke's law
 - Etc.
- Physical phenomena
 - Gravity
 - Momentum
 - Friction
 - Collisions
 - Elasticity
 - Fracture



Particle Systems

- A particle is a point mass

- Position
- Velocity
- Mass
- Drag
- Elasticity
- Lifetime
- Color



- Use lots of particles to model complex phenomena
 - Keep array of particles
 - Newton's laws

Particle Systems



- For each frame:
 - For each simulation step (Δt)
 - Create new particles and assign attributes
 - Update particles based on attributes and physics
 - Delete any expired particles
 - Render particles

Creating Particles



- Where to create particles?
 - Predefined source
 - Where particle density is low
 - Surface of shape
 - etc.

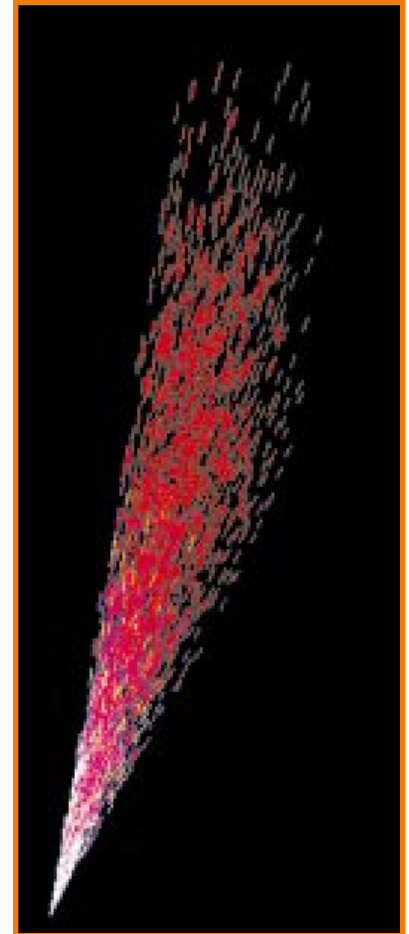
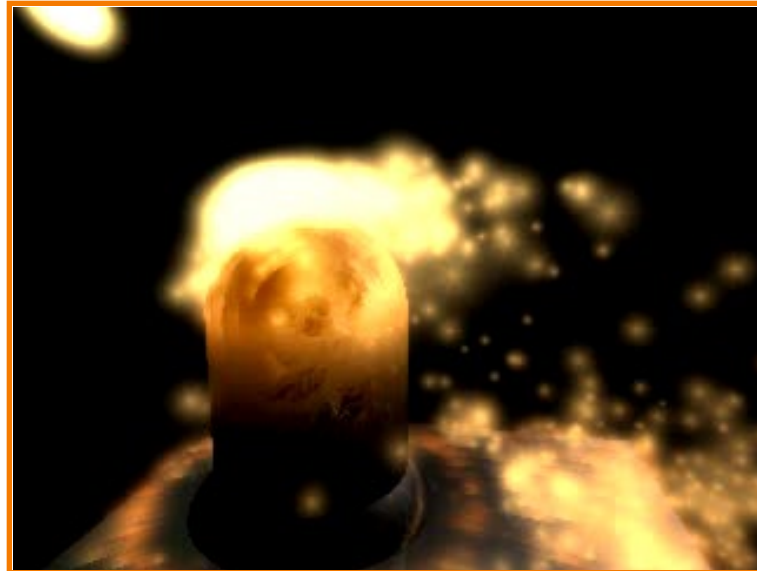
Reeves



Creating Particles



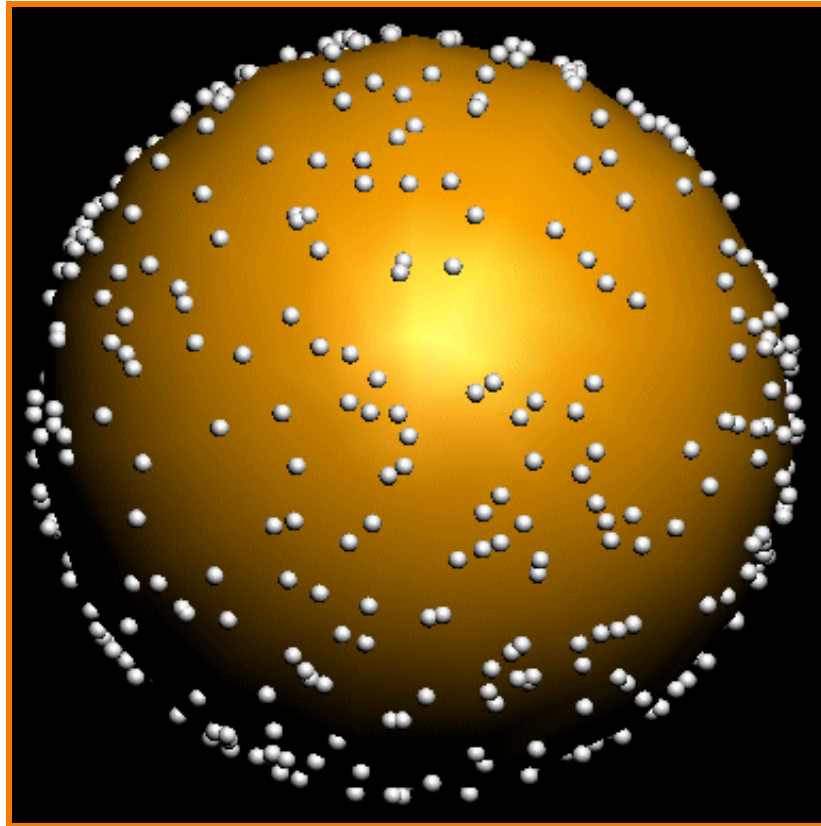
- Example: particles emanating from shape
 - Line
 - Box
 - **Circle**
 - **Sphere**
 - Cylinder
 - Cone
 - Mesh



Creating Particles



- Example: particles emanating from sphere



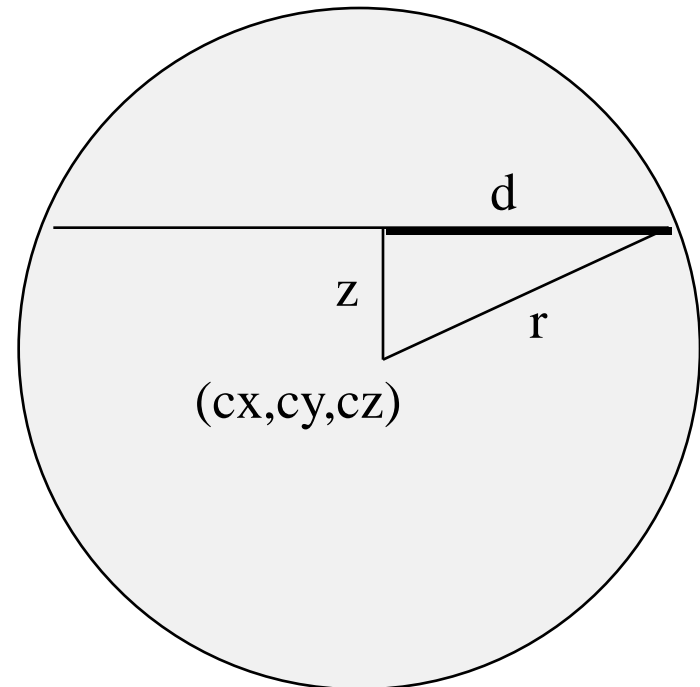
Creating Particles



- Example: particles emanating from sphere

Selecting random position on surface of sphere

1. $z = \text{random} [-r, r]$
2. $\text{phi} = \text{random} [0, 2\pi)$
3. $d = \text{sqrt}(r^2 - z^2)$
4. $p_x = c_x + d * \cos(\text{phi})$
5. $p_y = c_y + d * \sin(\text{phi})$
6. $p_z = c_z + z$



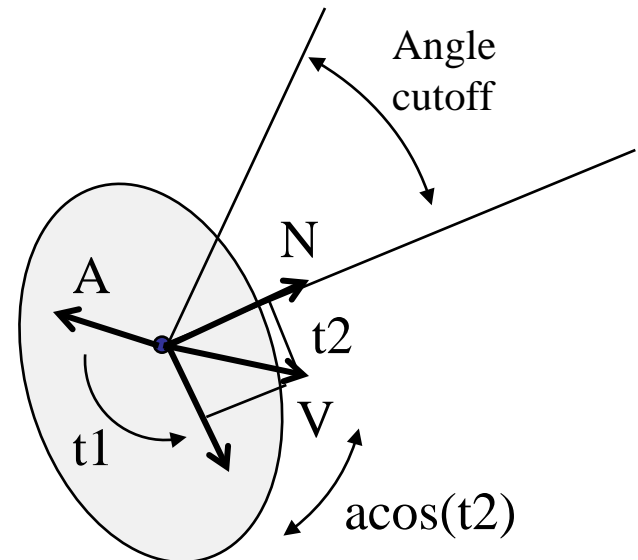
Creating Particles



- Example: particles emanating from sphere

Selecting random direction within angle cutoff of normal

1. N = surface normal
2. A = any vector on tangent plane
3. $t1$ = random $[0, 2\pi)$
3. $t2$ = random $[0, \sin(\text{angle cutoff}))$
4. V = rotate A around N by $t1$
5. V = rotate V around $V \times N$ by $\text{acos}(t2)$



Particle Systems



- For each frame:
 - For each simulation step (Δt)
 - Create new particles and assign attributes
 - Update particles based on attributes and physics
 - Delete any expired particles
 - Render particles



Equations of Motion

- Newton's Law for a point mass
 - $f = ma$
- Computing particle motion requires solving second-order differential equation

$$\ddot{x} = \frac{f(x, \dot{x}, t)}{m}$$

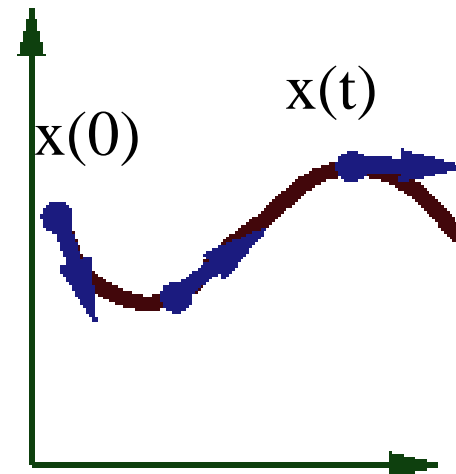
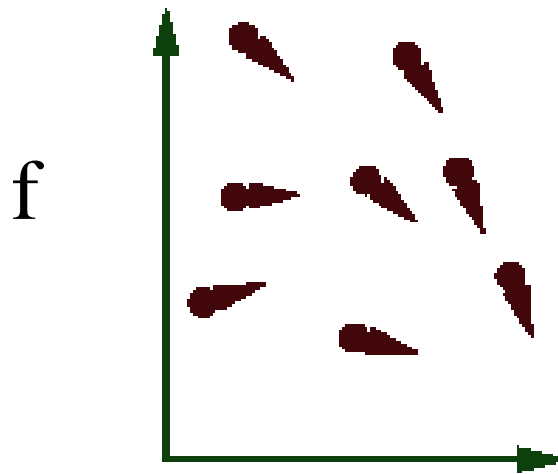
- Add variable v to form coupled **first-order** differential equations: “state-space form”

$$\begin{cases} \dot{x} = v \\ \dot{v} = \frac{f}{m} \end{cases}$$

Solving the Equations of Motion



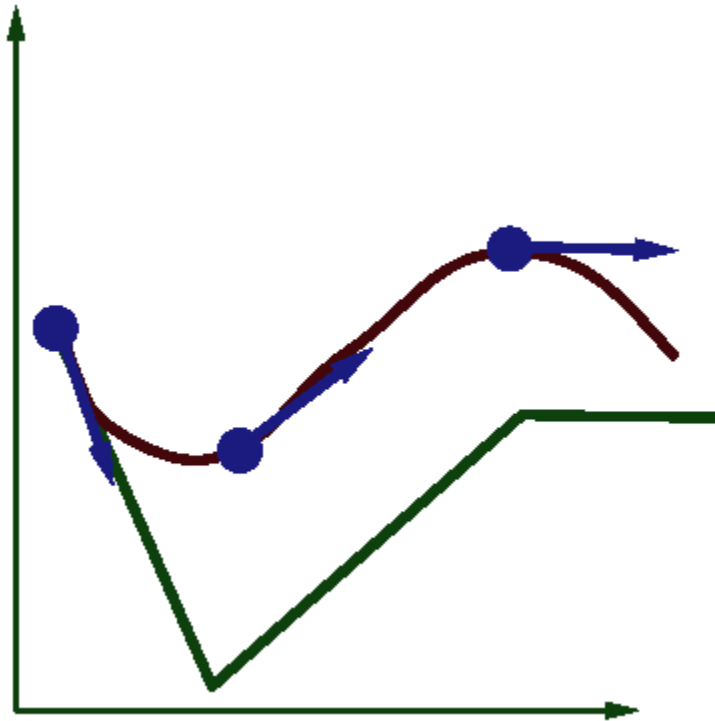
- Initial value problem
 - Know $x(0)$, $v(0)$
 - Can compute force (and therefore acceleration) for any position / velocity / time
 - Compute $x(t)$ by forward integration



Solving the Equations of Motion



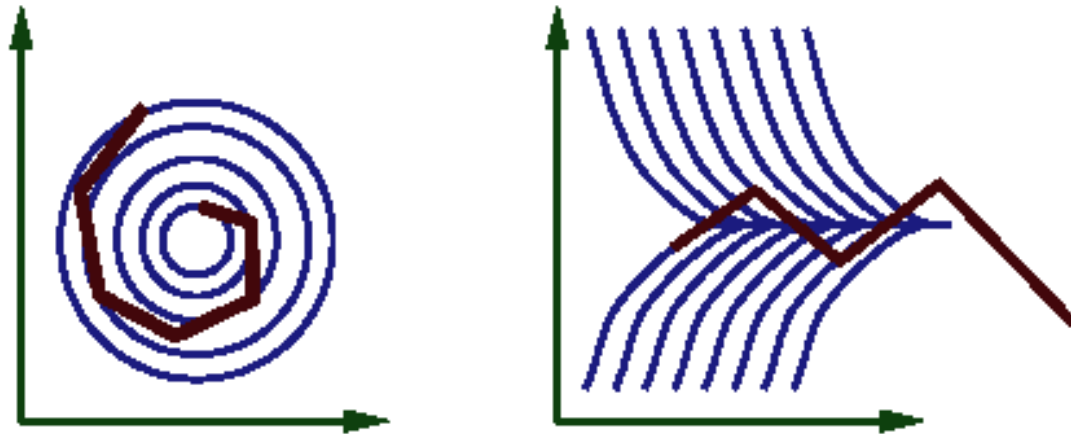
- Forward (explicit) Euler integration
 - $x(t+\Delta t) \leftarrow x(t) + \Delta t v(t)$
 - $v(t+\Delta t) \leftarrow v(t) + \Delta t f(x(t), v(t), t) / m$



Solving the Equations of Motion



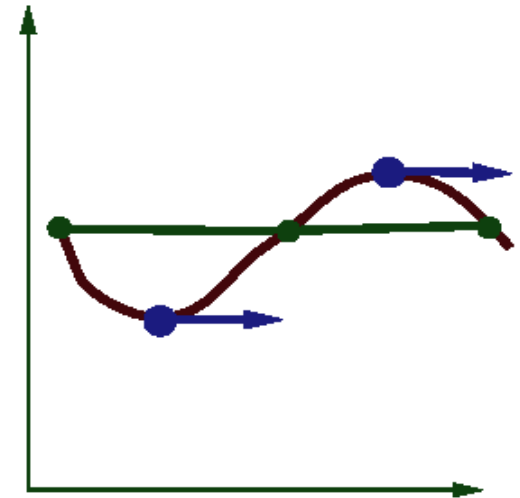
- Forward (explicit) Euler integration
 - $x(t+\Delta t) \leftarrow x(t) + \Delta t v(t)$
 - $v(t+\Delta t) \leftarrow v(t) + \Delta t f(x(t), v(t), t) / m$
- Problem:
 - Accuracy decreases as Δt gets bigger



Solving the Equations of Motion



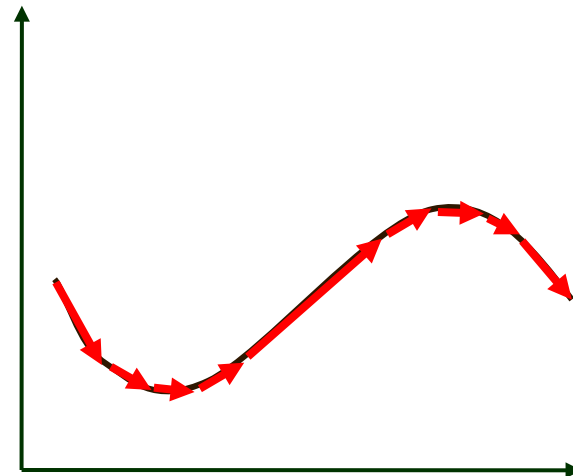
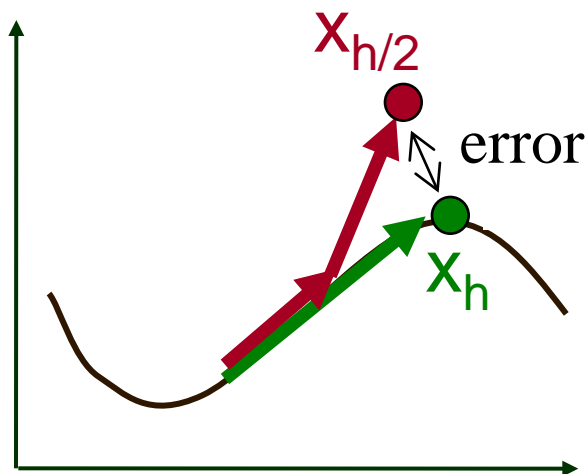
- Midpoint method (2nd-order Runge-Kutta)
 1. Compute an Euler step
 2. Evaluate f at the **midpoint** of Euler step
 3. Compute new position / velocity using midpoint velocity / acceleration
 - $x_{\text{mid}} \leftarrow x(t) + \Delta t / 2 * v(t)$
 - $v_{\text{mid}} \leftarrow v(t) + \Delta t / 2 * f(x(t), v(t), t) / m$
 - $x(t+\Delta t) \leftarrow x(t) + \Delta t v_{\text{mid}}$
 - $v(t+\Delta t) \leftarrow v(t) + \Delta t f(x_{\text{mid}}, v_{\text{mid}}, t) / m$



Solving the Equations of Motion



- Adaptive step size
 - Repeat until error is below threshold
 1. Compute x_h by taking one step of size h
 2. Compute $x_{h/2}$ by taking 2 steps of size $h / 2$
 3. Compute error = $|x_h - x_{h/2}|$
 4. If (error < threshold) break
 5. Else, reduce step size and try again





Particle System Forces

- Force fields
 - Gravity, wind, pressure
- Viscosity/damping
 - Drag, friction
- Collisions
 - Static objects in scene
 - Other particles
- Attraction and repulsion
 - Springs between neighboring particles (mesh)
 - Gravitational pull, charge



Particle System Forces

- Gravity
 - Force due to gravitational pull (of earth)
 - g = acceleration due to gravity (m/s^2)

$$f_g = mg$$

↓

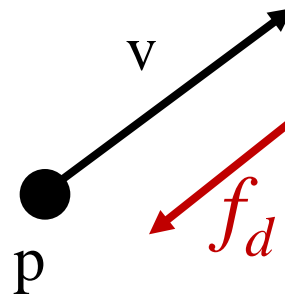
$$g = (0, -9.80665, 0)$$

Particle System Forces



- Drag
 - Force due to resistance of medium
 - k_{drag} = drag coefficient (kg/s)

$$f_d = -k_{\text{drag}} v$$

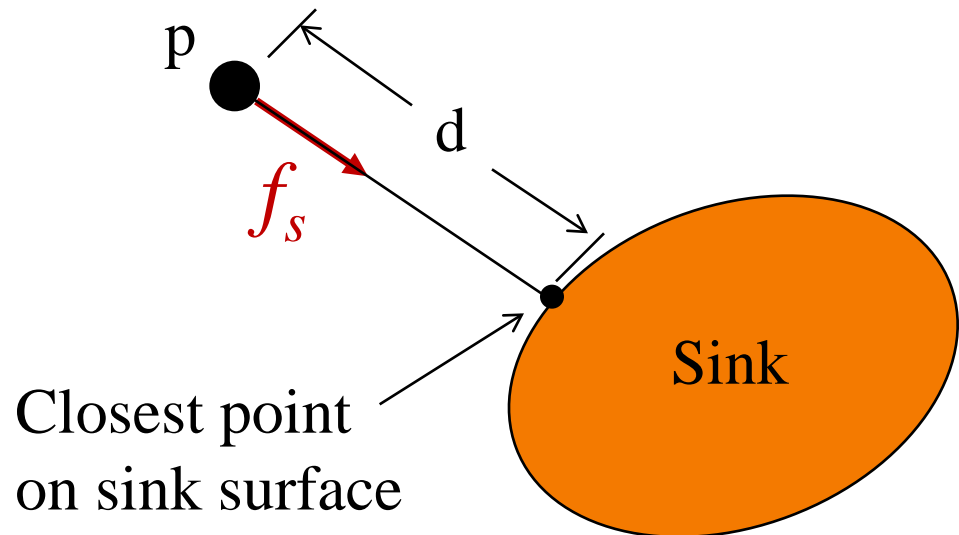


- Air resistance sometimes taken as proportional to v^2

Particle System Forces

- Sinks
 - Force due to attractor in scene

$$f_s = \frac{\text{intensity}}{ca + la \cdot d + qa \cdot d^2}$$



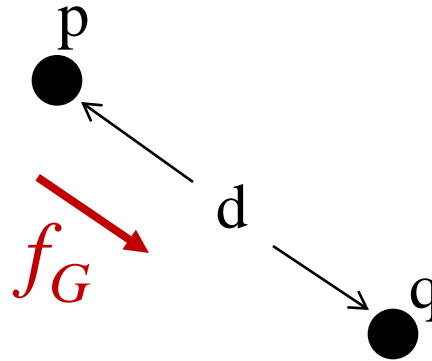


Particle System Forces

- Gravitational pull of other particles
 - Newton's universal law of gravitation

$$f_G = G \frac{m_1 \cdot m_2}{d^2}$$

$$G = 6.67428 \times 10^{-11} \text{ N m}^2 \text{ kg}^{-2}$$



Particle System Forces



- Springs
 - Hooke's law

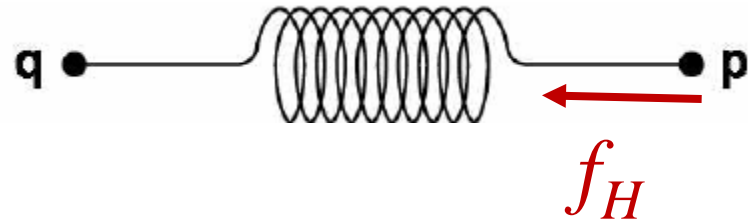
$$f_H(p) = k_s(d(p, q) - s) D$$

$$D = (q - p) / \|q - p\|$$

$$d(p, q) = \|q - p\|$$

s = resting length

k_s = spring coefficient



Particle System Forces



- Springs
 - Hooke's law with damping

$$f_H(p) = [k_s(d(p, q) - s) + k_d(v(q) - v(p) \cdot D)] D$$

$$D = (q - p) / \|q - p\|$$

$$d(p, q) = \|q - p\|$$

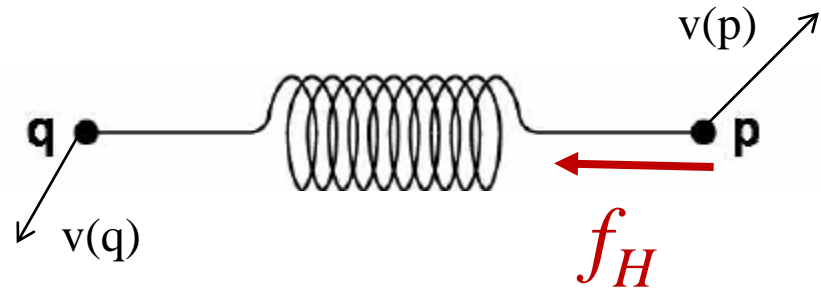
s = resting length

k_s = spring coefficient

k_d = damping coefficient

$v(p)$ = velocity of p

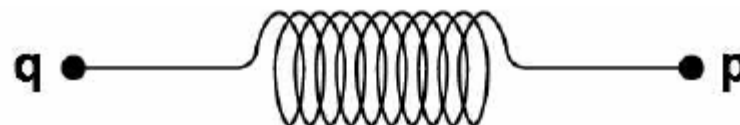
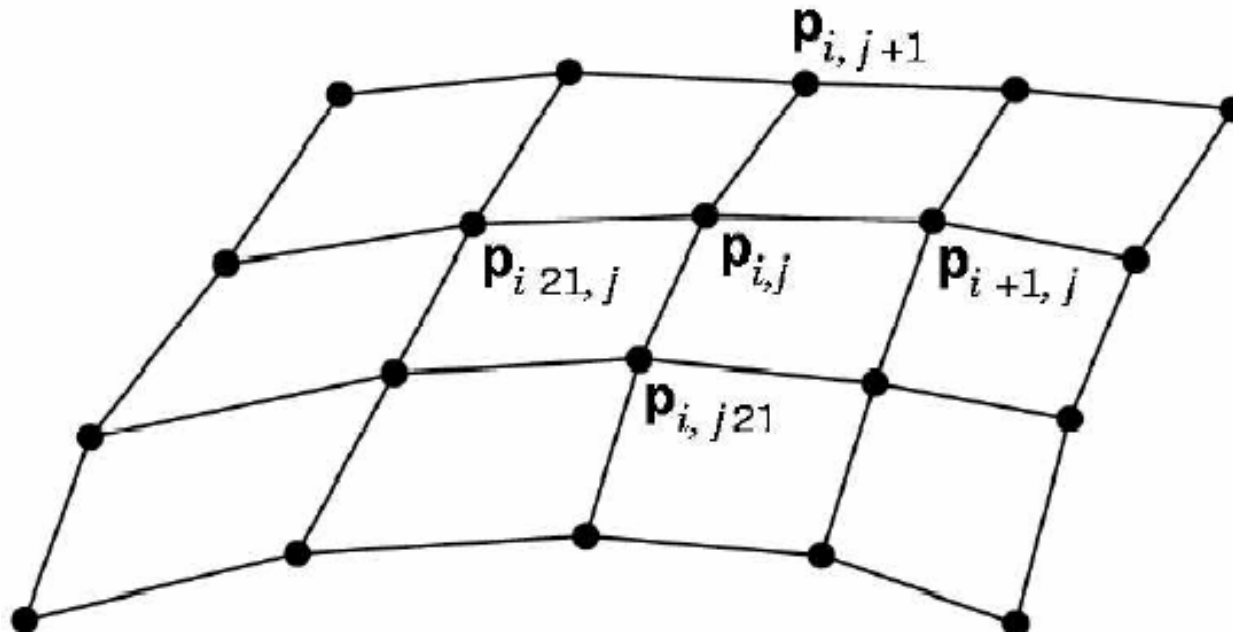
$v(q)$ = velocity of q



$$k_d \sim 2\sqrt{mk_s}$$

Particle System Forces

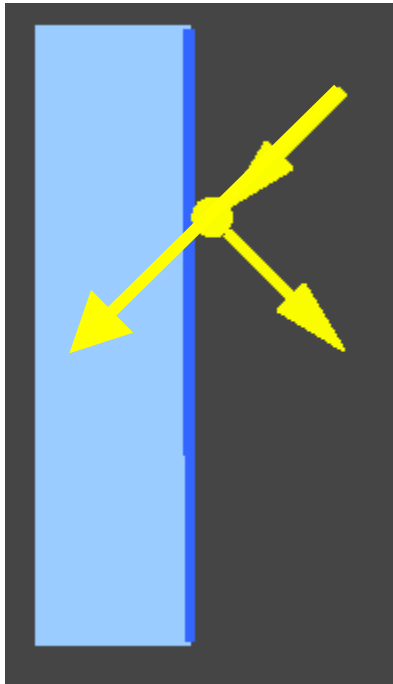
- Spring-mass mesh



Particle System Forces

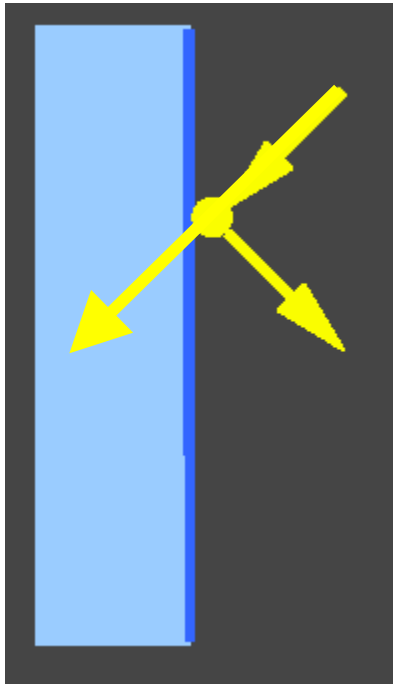


- Collisions
 - Collision detection
 - Collision response



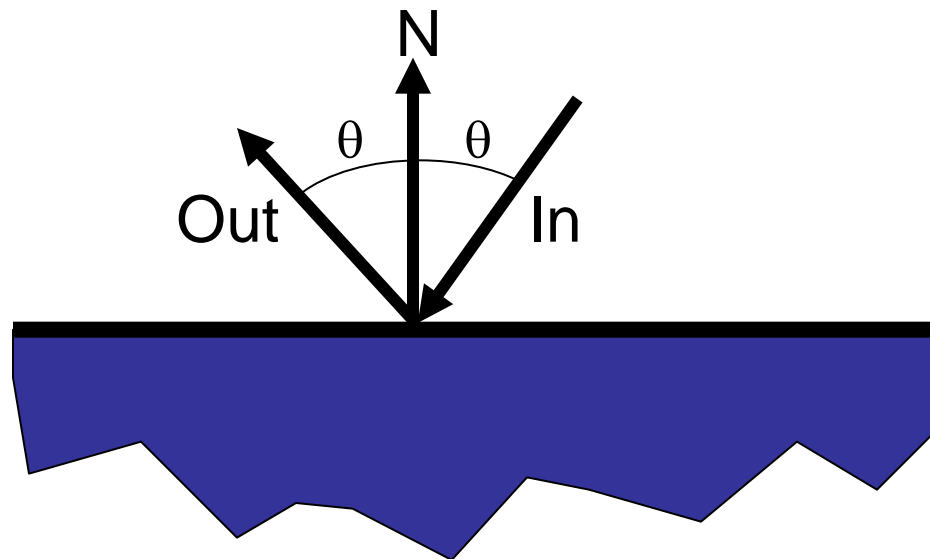
Particle System Forces

- Collision detection
 - Intersect ray with scene
 - Compute up to Δt at time of first collision, and then continue from there



Particle System Forces

- Collision response
 - No friction: elastic collision
(for $m_{\text{target}} \gg m_{text{particle}}$: specular reflection)



- Otherwise, total momentum conserved,
energy dissipated if inelastic

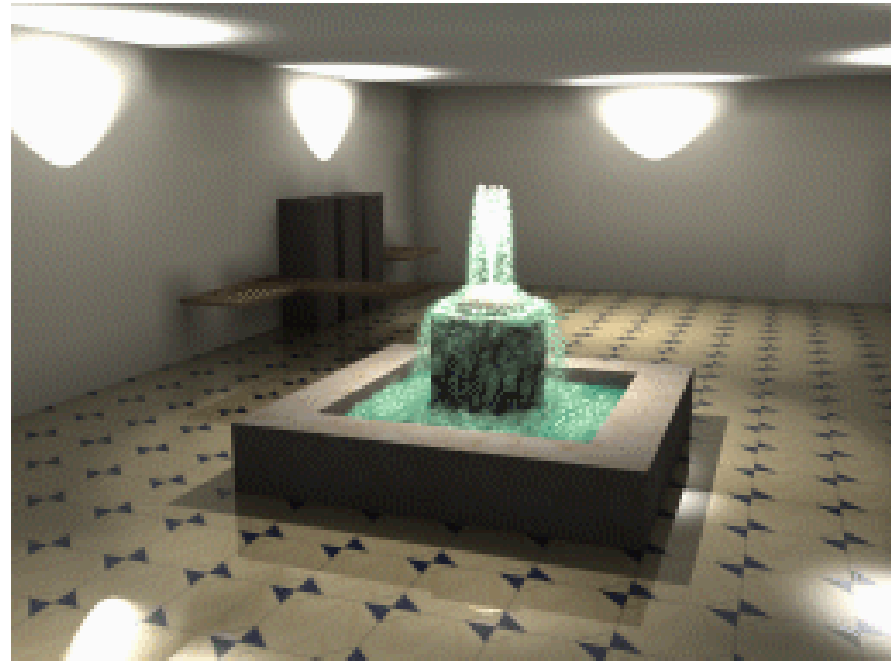
Particle Systems



- For each frame:
 - For each simulation step (Δt)
 - Create new particles and assign attributes
 - Update particles based on attributes and physics
 - Delete any expired particles
 - Render particles

Deleting Particles

- When to delete particles?
 - When life span expires
 - When intersect predefined sink surface
 - Where density is high
 - Random



Particle Systems



- For each frame:
 - For each simulation step (Δt)
 - Create new particles and assign attributes
 - Update particles based on attributes and physics
 - Delete any expired particles
 - Render particles

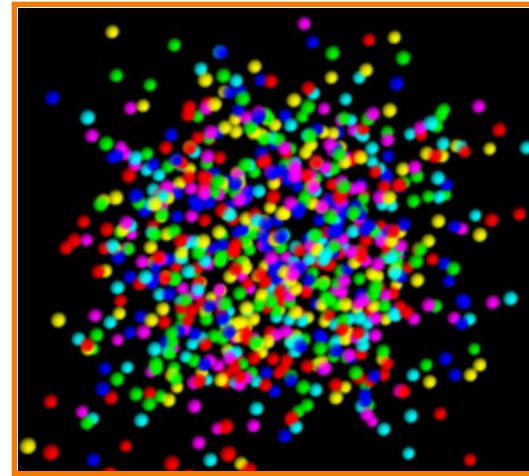
Rendering Particles



- Rendering styles

- **Points**

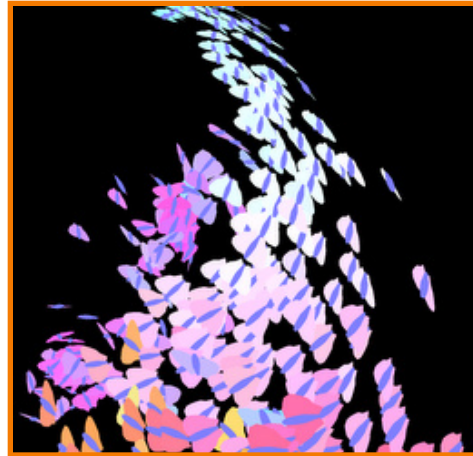
- Polygons
 - Shapes
 - Trails
 - etc.



Rendering Particles



- Rendering styles
 - Points
 - Textured polygons: sprites
 - Shapes
 - Trails
 - etc.



Rendering Particles



- Rendering styles
 - Points
 - Polygons
 - Shapes
 - Trails
 - etc.



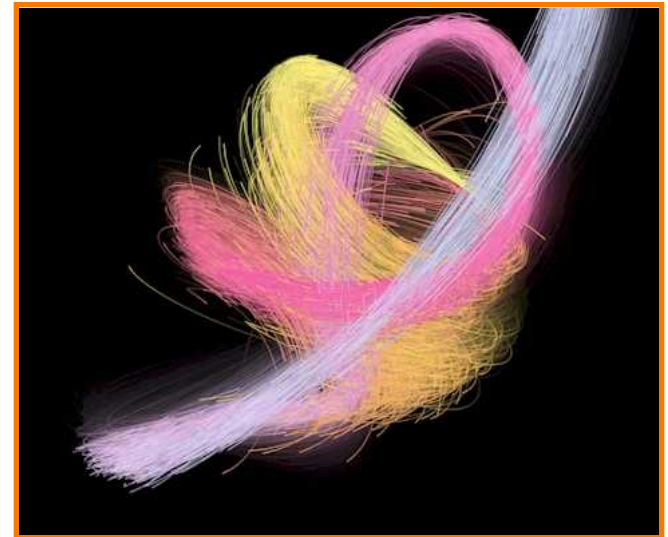
Rendering Particles



- Rendering styles
 - Points
 - Polygons
 - Shapes
 - Trails
 - etc.



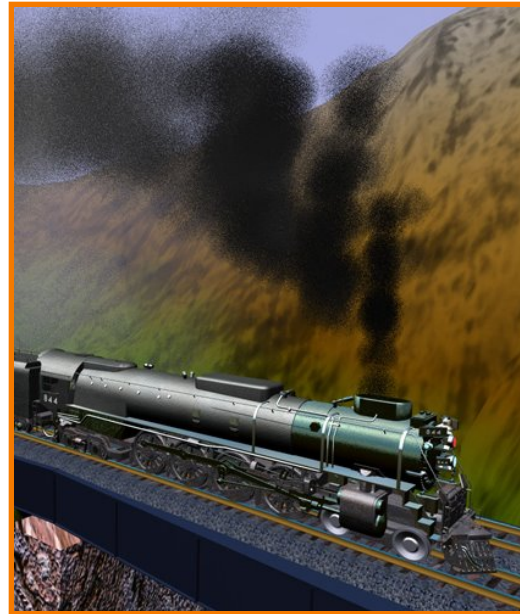
McAllister



Passive Dynamics



- Examples
 - Smoke
 - Water
 - Cloth
 - Fire
 - Fireworks
 - Dice



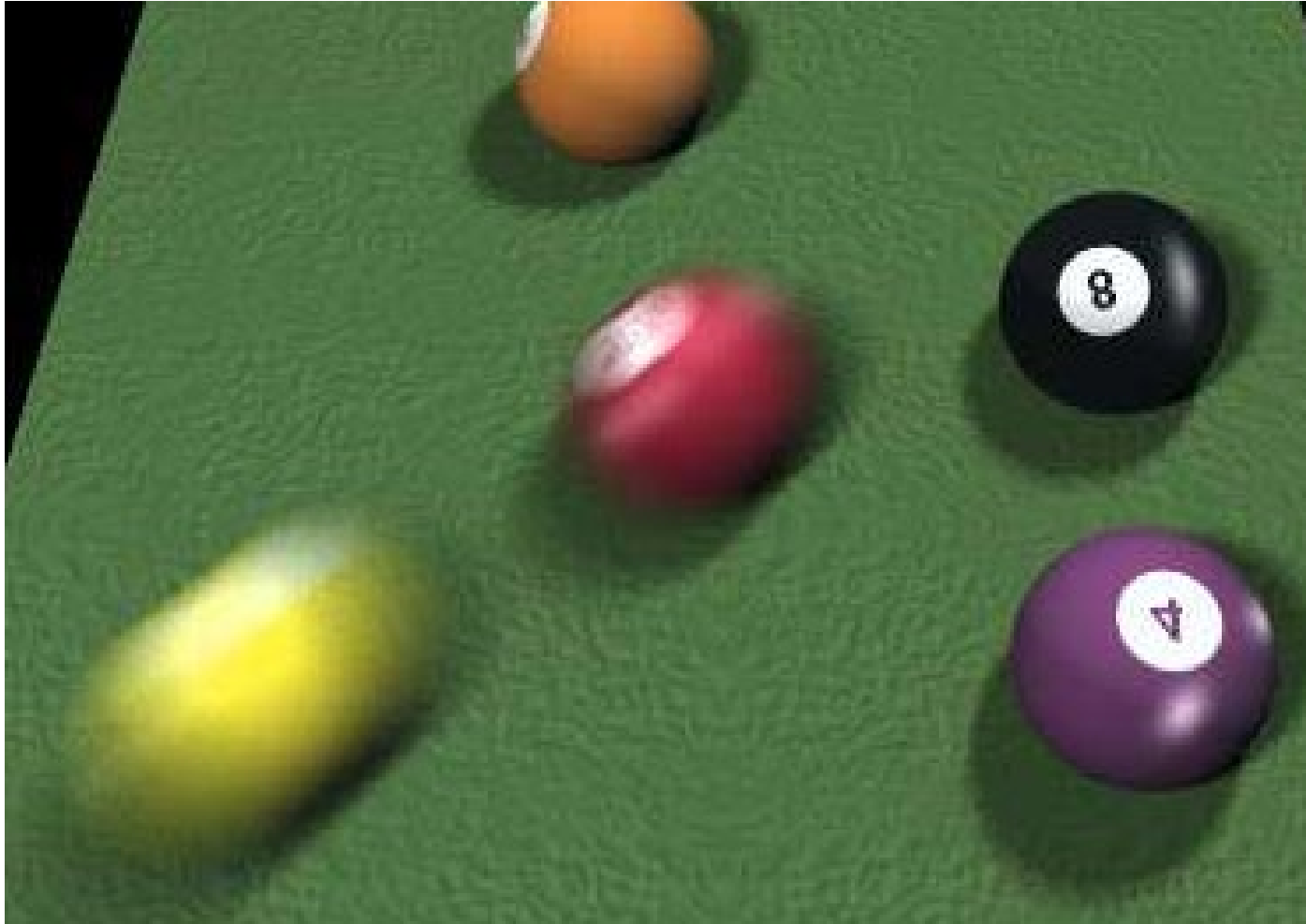
Example: Gravity



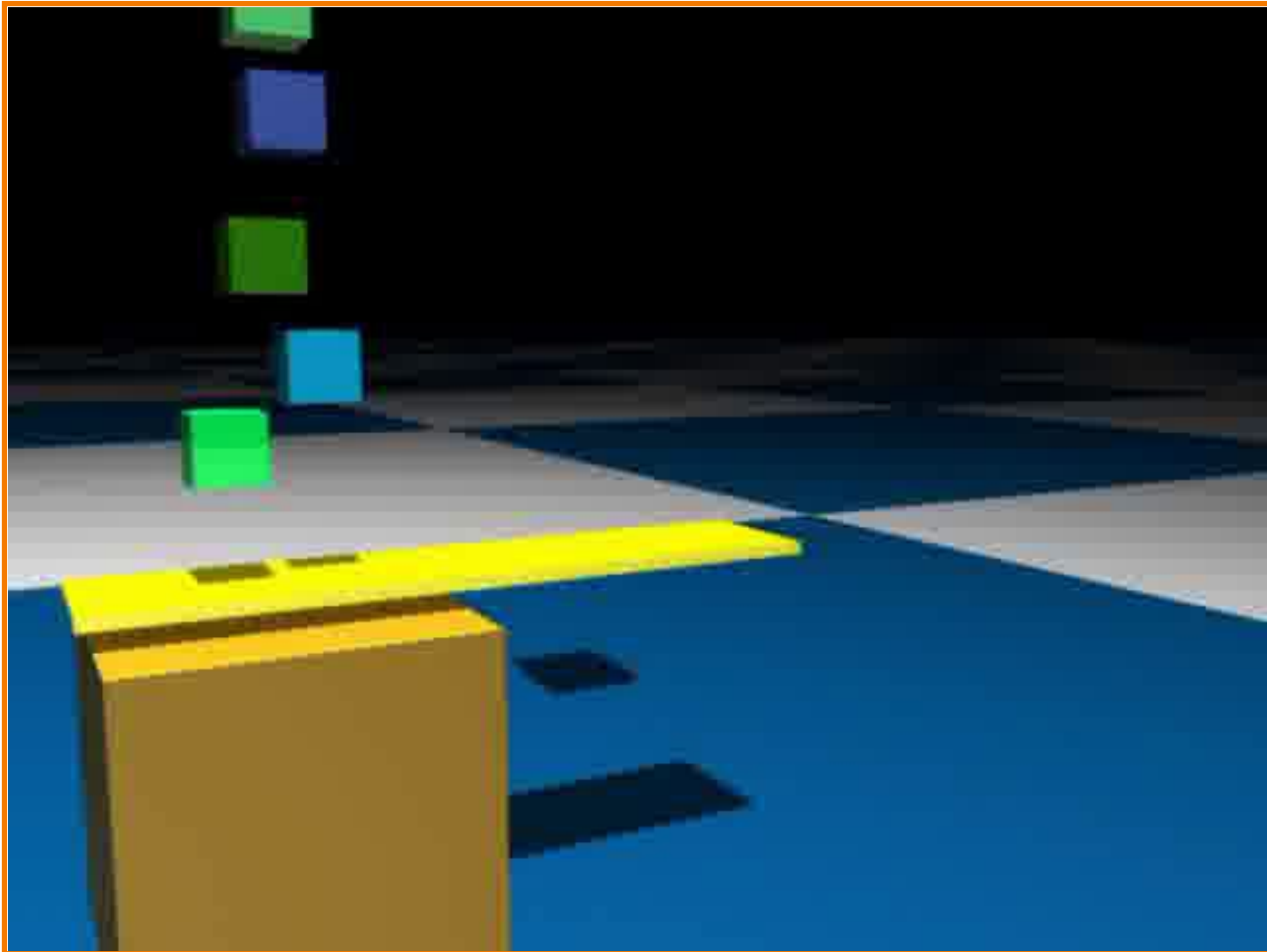
Example: Fire



Example: Bouncing Off Particles

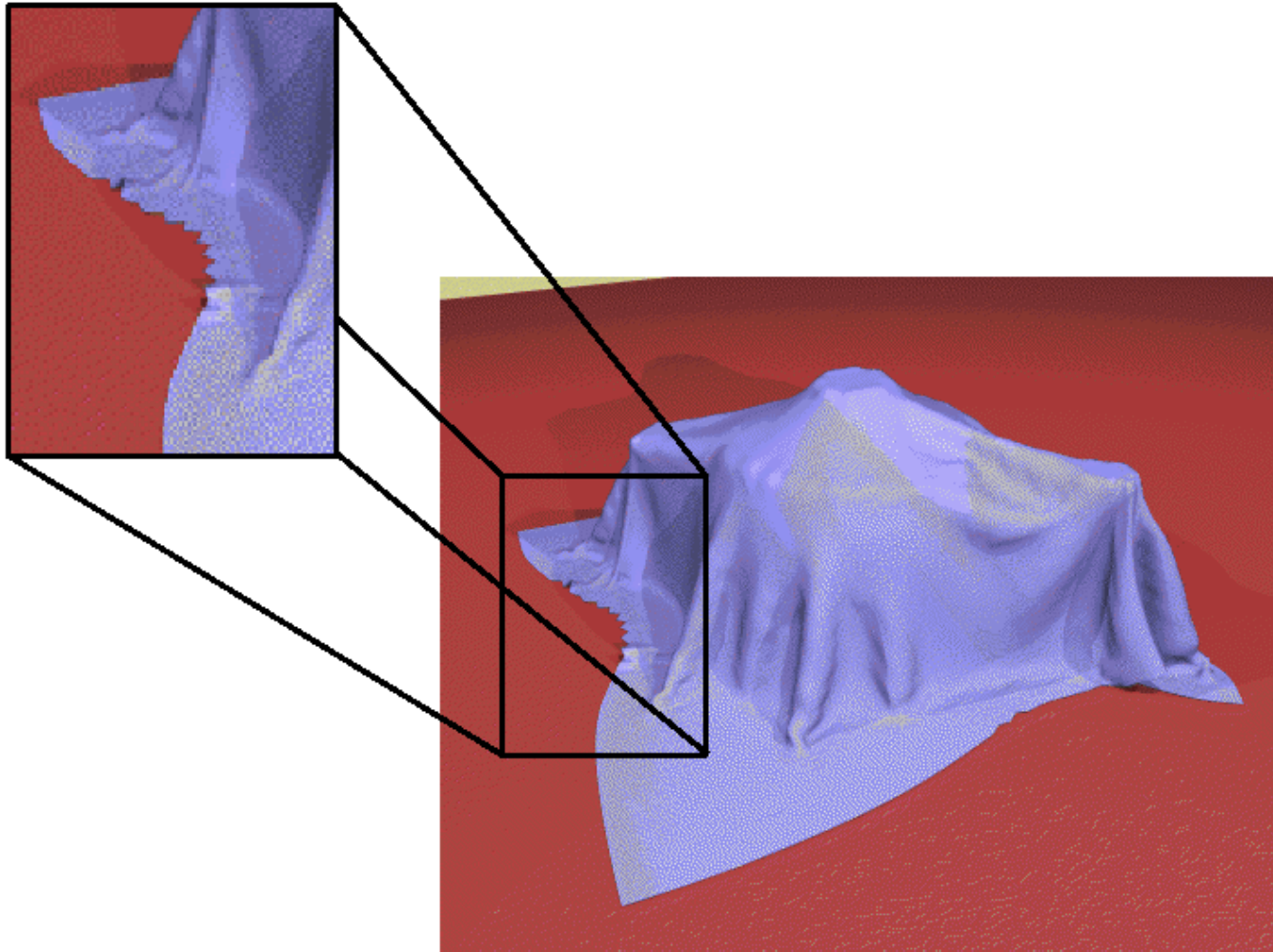


Example: More Bouncing

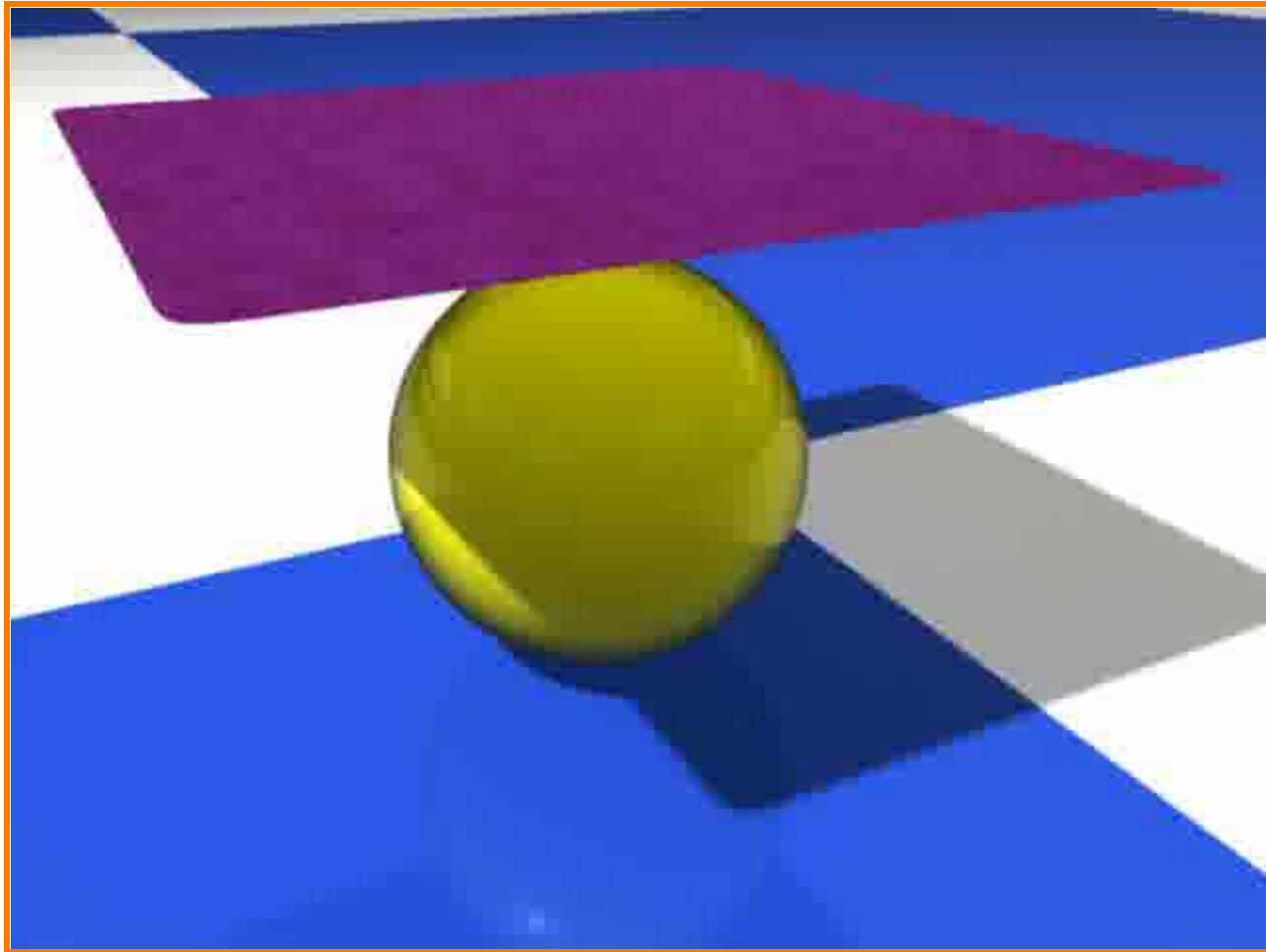


Bender

Example: Cloth

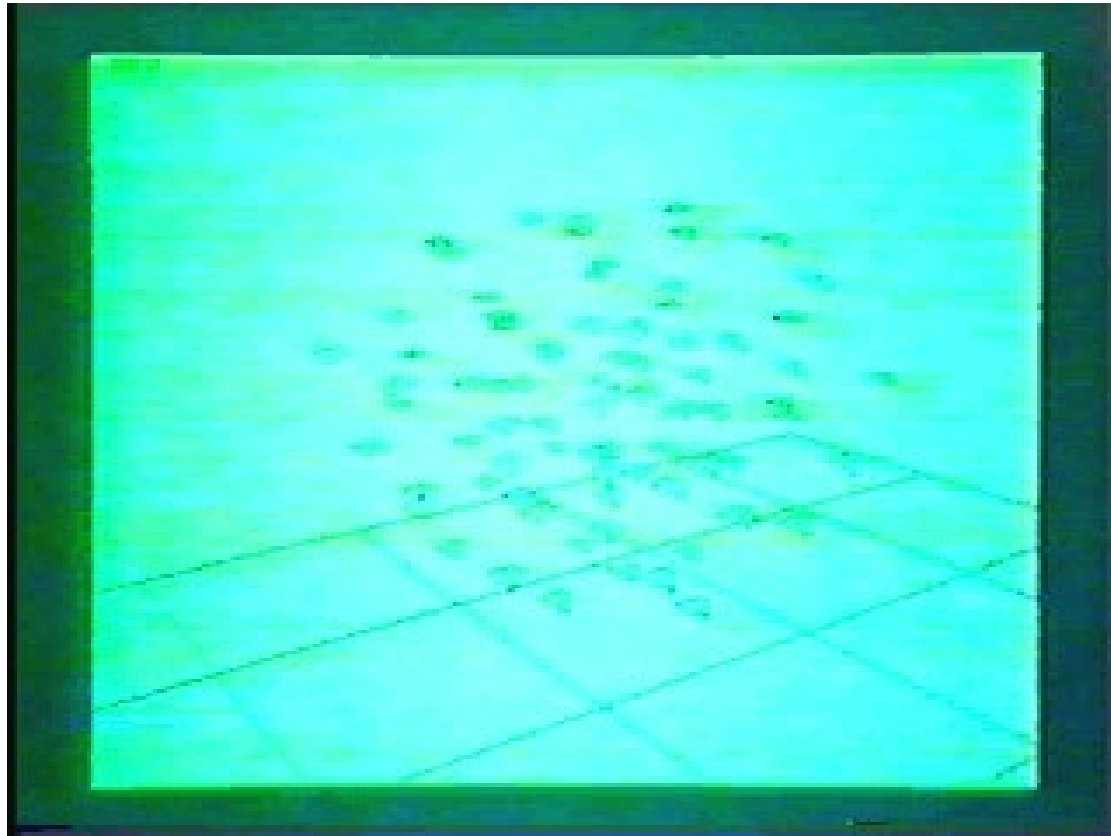


Example: Cloth



Bender

Example: Flocks & Herds



Summary



- Particle systems
 - Lots of particles
 - Simple physics
- Interesting behaviors
 - Waterfalls
 - Smoke
 - Cloth
 - Flocks
- Solving motion equations
 - For each step, first sum forces, then update position and velocity

