

Particle Systems

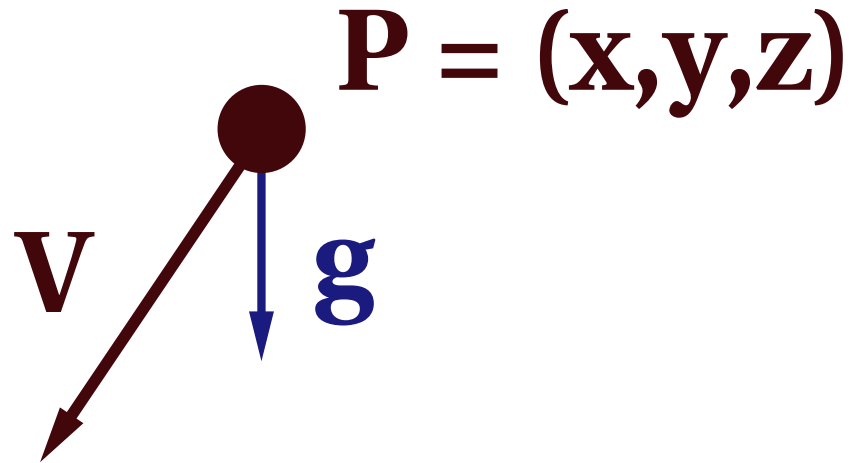
What control handles to we want/need?

Equations of Motion:

$$\mathbf{A} = \mathbf{g}$$

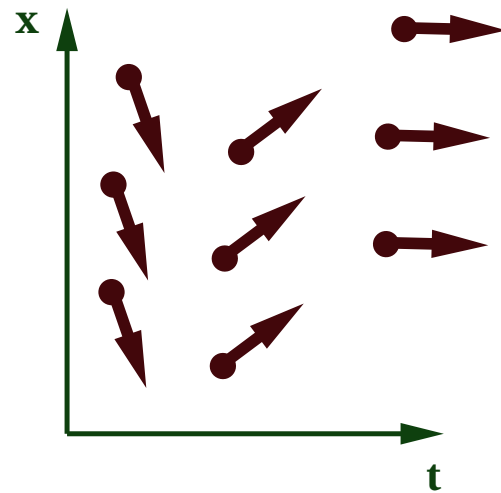
$$\mathbf{V}' = \mathbf{V} + \mathbf{A}\Delta t$$

$$\mathbf{P}' = \mathbf{P} + \frac{\mathbf{V} + \mathbf{V}'}{2} \Delta t$$



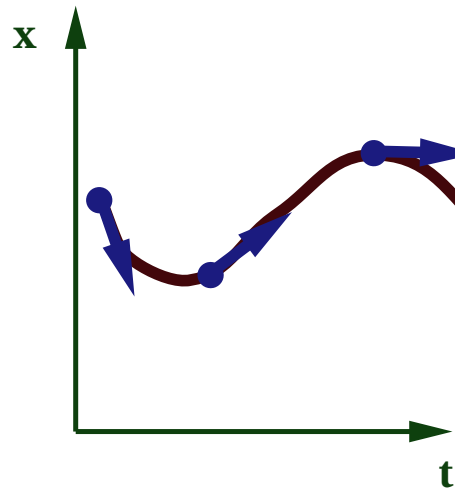
Initial Value Problem:

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



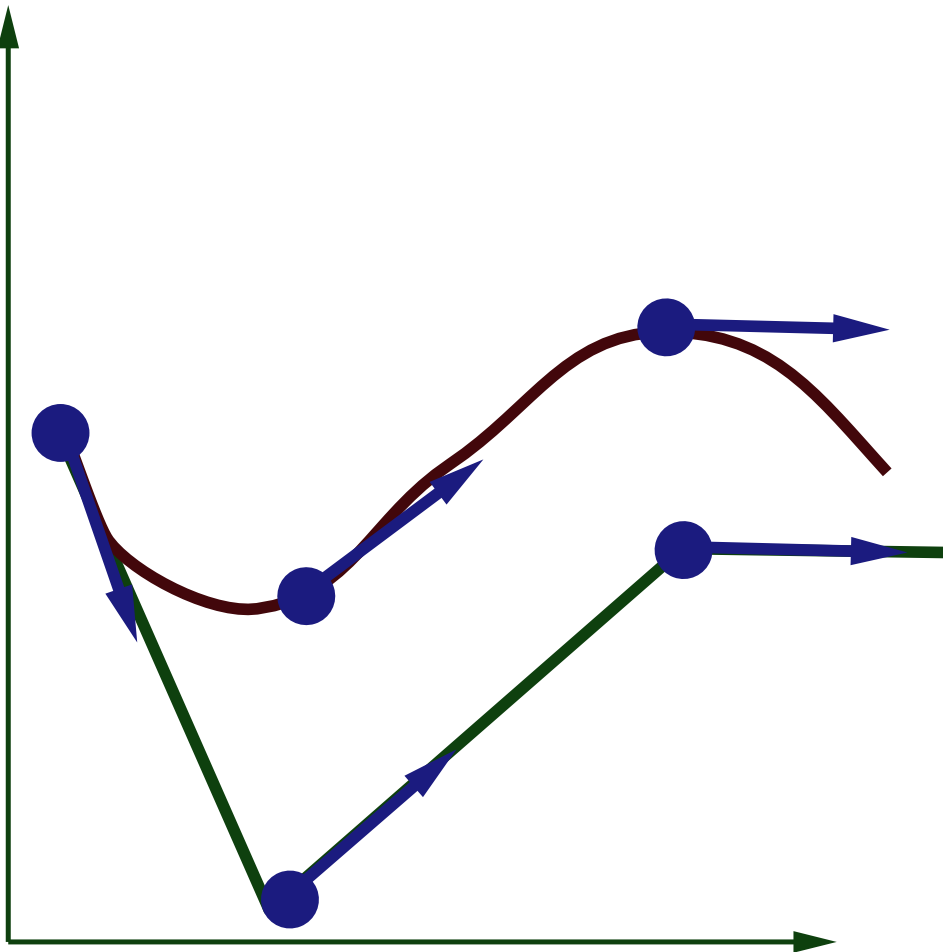
**initial conditions
determine the motion**

know x_0



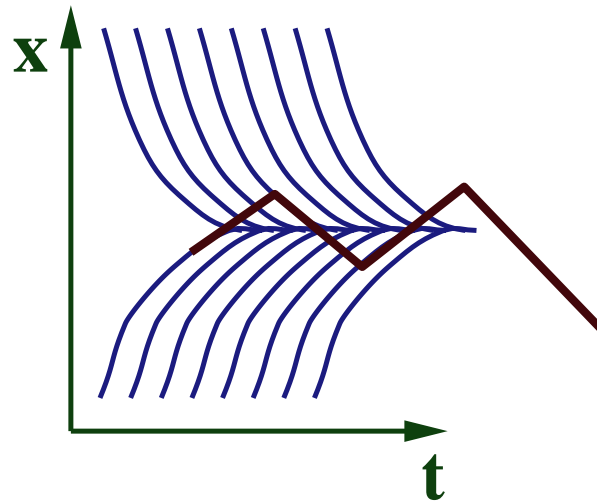
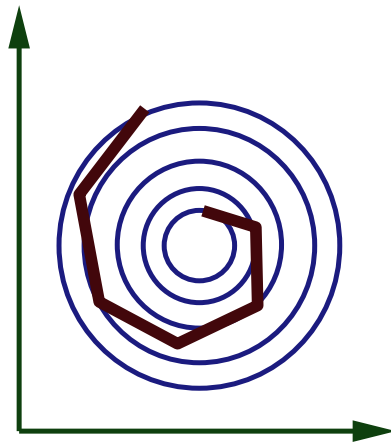
Euler Integration:

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



Euler Integration: Accuracy

as $\Delta t \Rightarrow 0$ accuracy increases



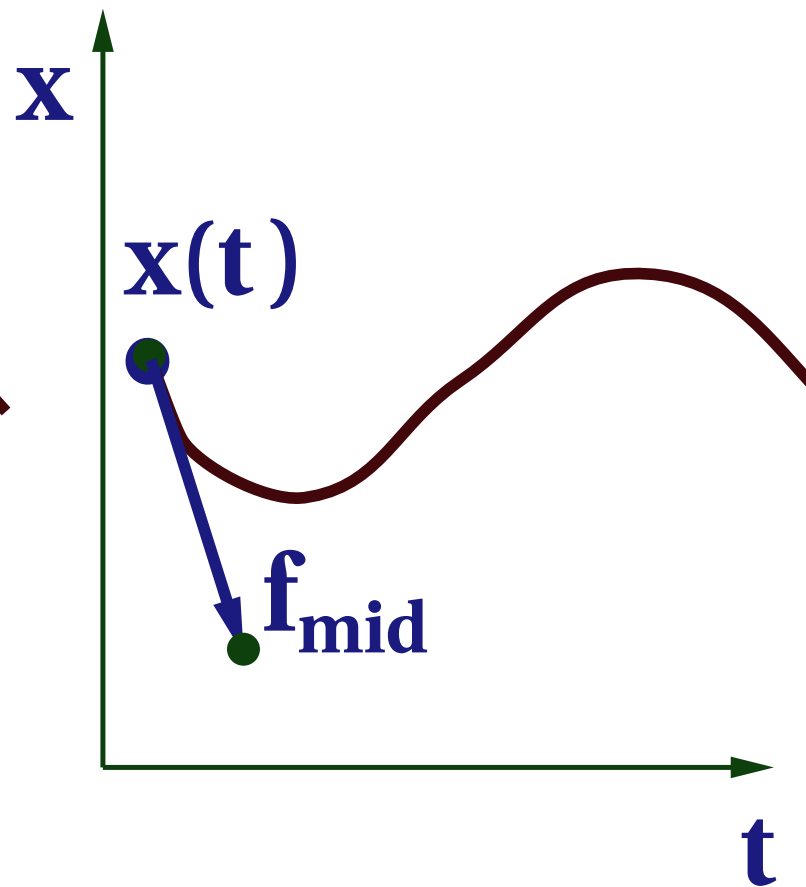
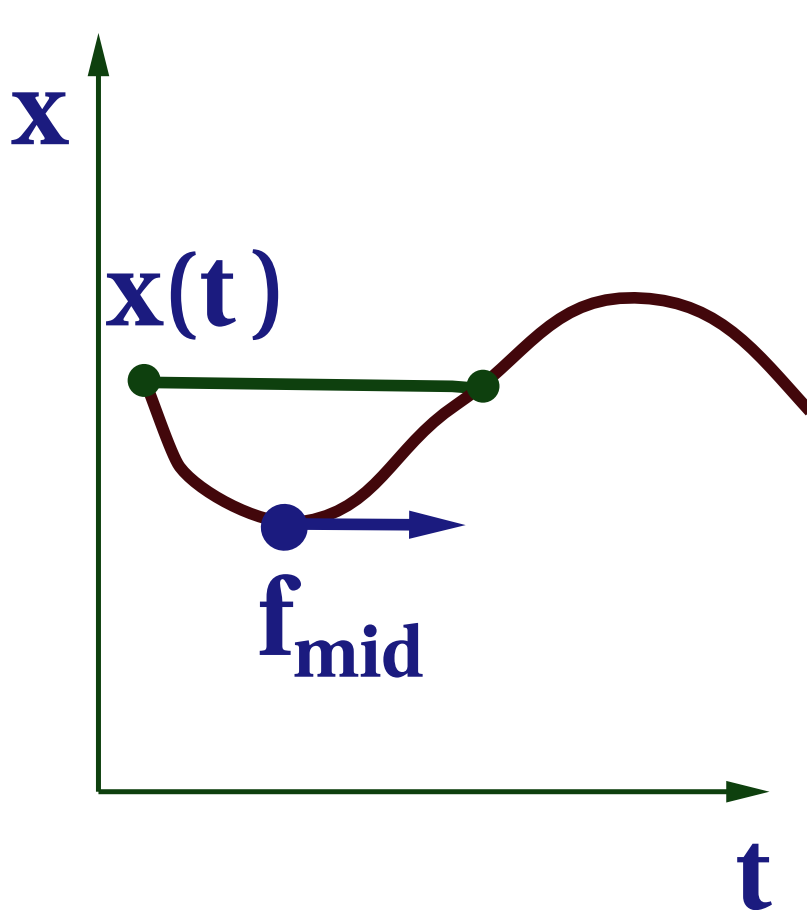
$$f = -kx$$

$\Delta t > 2/k$: unstable

Higher Order Integration Methods

midpoint method:

$$x(t + h) = x(t) + h f_{\text{mid}}$$



Adaptive Stepsizes

take advantage of the smooth parts of the space by increasing the stepsize

$$\mathbf{x}(t+0.001) = \mathbf{x}_a$$

$$\mathbf{x}(t+0.0005 + 0.0005) = \mathbf{x}_b$$

$$\mathbf{e} = |\mathbf{x}_a - \mathbf{x}_b|$$

$$\mathbf{h}' = (\mathbf{e} / \mathbf{e}_{\max})^{1/2} \mathbf{h}$$

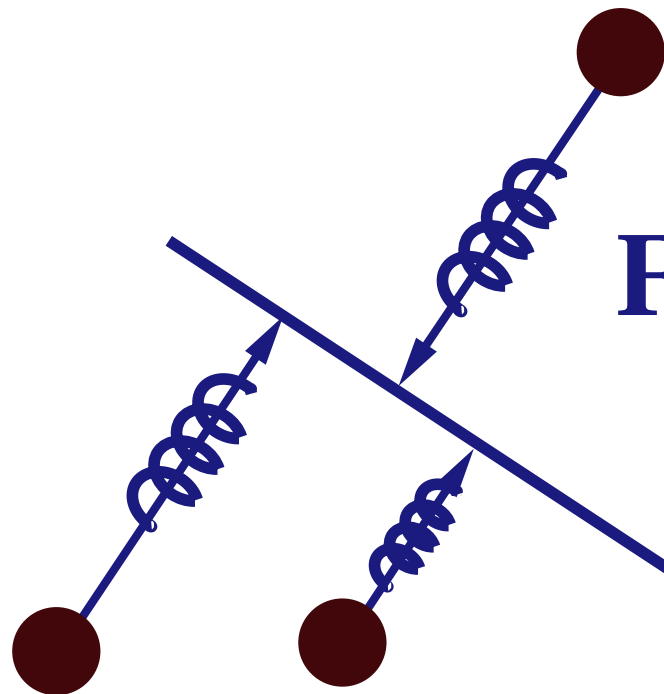
Particle Systems -- Forces

acceleration

random

$$\mathbf{F}' = \mathbf{F} + \text{random}()$$

accelerate towards point, line



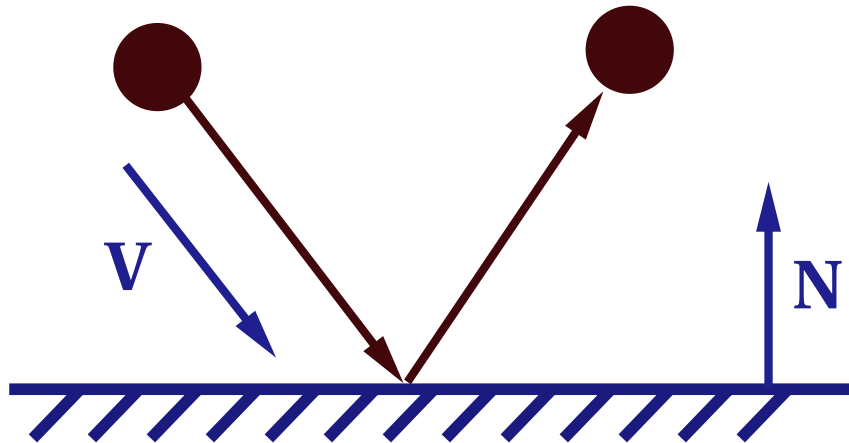
$$\mathbf{F} = -\mathbf{k} (\mathbf{P} - \mathbf{P}_d)$$

acceleration

damping $\mathbf{F} = -k'\mathbf{V}$

bouncing (coefficient of restitution)

$$\mathbf{V}' = \mathbf{V} - 2(\mathbf{V} \cdot \mathbf{N})\mathbf{N}$$



Basic Program Structure

create particles
initialize particles
for each frame

set tails to previous heads
for each simulation time step
compute forces, acceleration
compute velocity
compute position
check for collisions
adjust tails for shutter speed
render

Forces

unary

gravity: $f = mg$

viscous drag: $f = -kv$

n-ary

spring: $f = ke$, $f' = -ke$ for fixed pairs of particles

spatial interaction forces

collisions: interaction between arbitrary sets of particles and objects in the environment

Collision Detection

with a plane:

P is a point on the plane

N is normal to plane (pointing to legal side)

if $(X-P) \cdot N > 0$ outside, < 0 inside

in general:

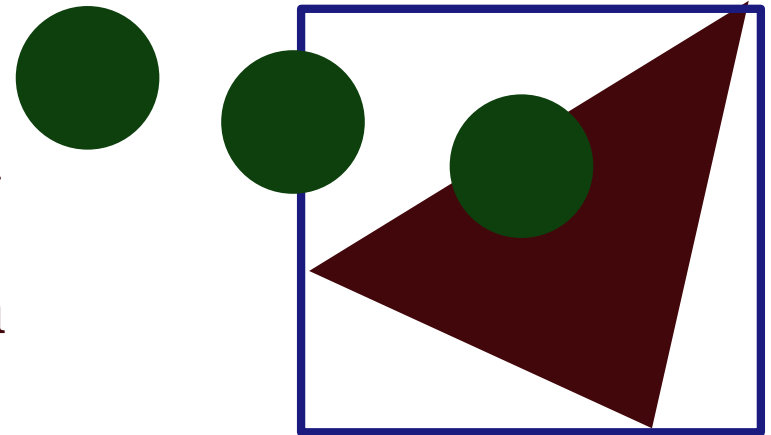
collide points with triangles

hierarchical collision detection

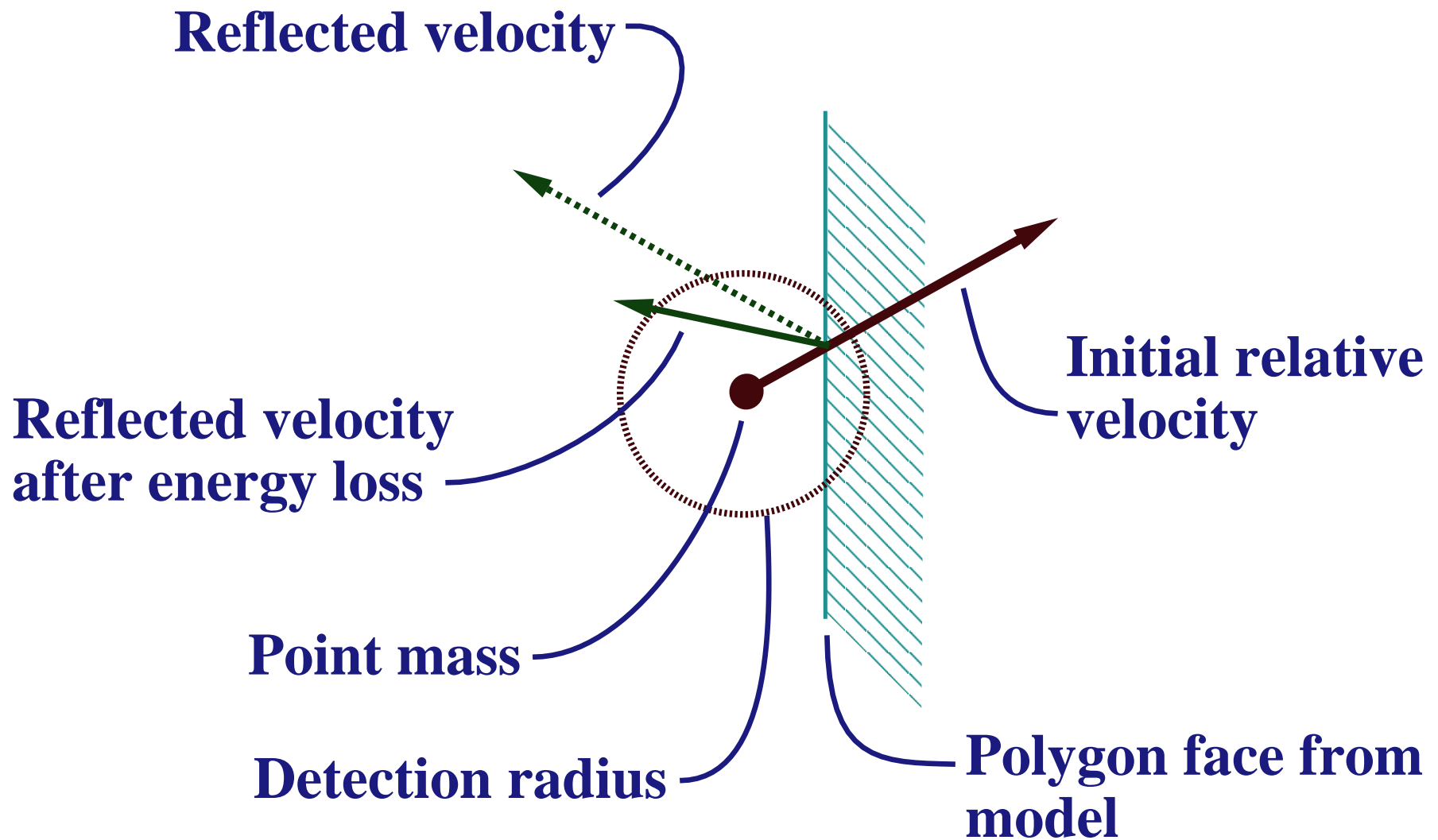
bounding boxes

incremental collision detection

tunneling



Collision Response



Collision Response

spring/damper

easy to implement

good for "soft" systems

stiff systems \rightarrow small integration timestep

constraint stabilization

stiff

analytical solutions (general. of prev. slide)

exact answers

faster integration

complex to derive and implement