Rigid Body Dynamics

From Particles to Rigid Bodies

- Particles
  - No rotations
  - Linear velocity v only
- Rigid bodies
  - Body rotations
  - Linear velocity v
  - Angular velocity ω

Rigid Bodies

Rigid bodies have both a position and orientation

Rigid bodies assume no object deformation

Rigid body motion is represented by 2 parameters

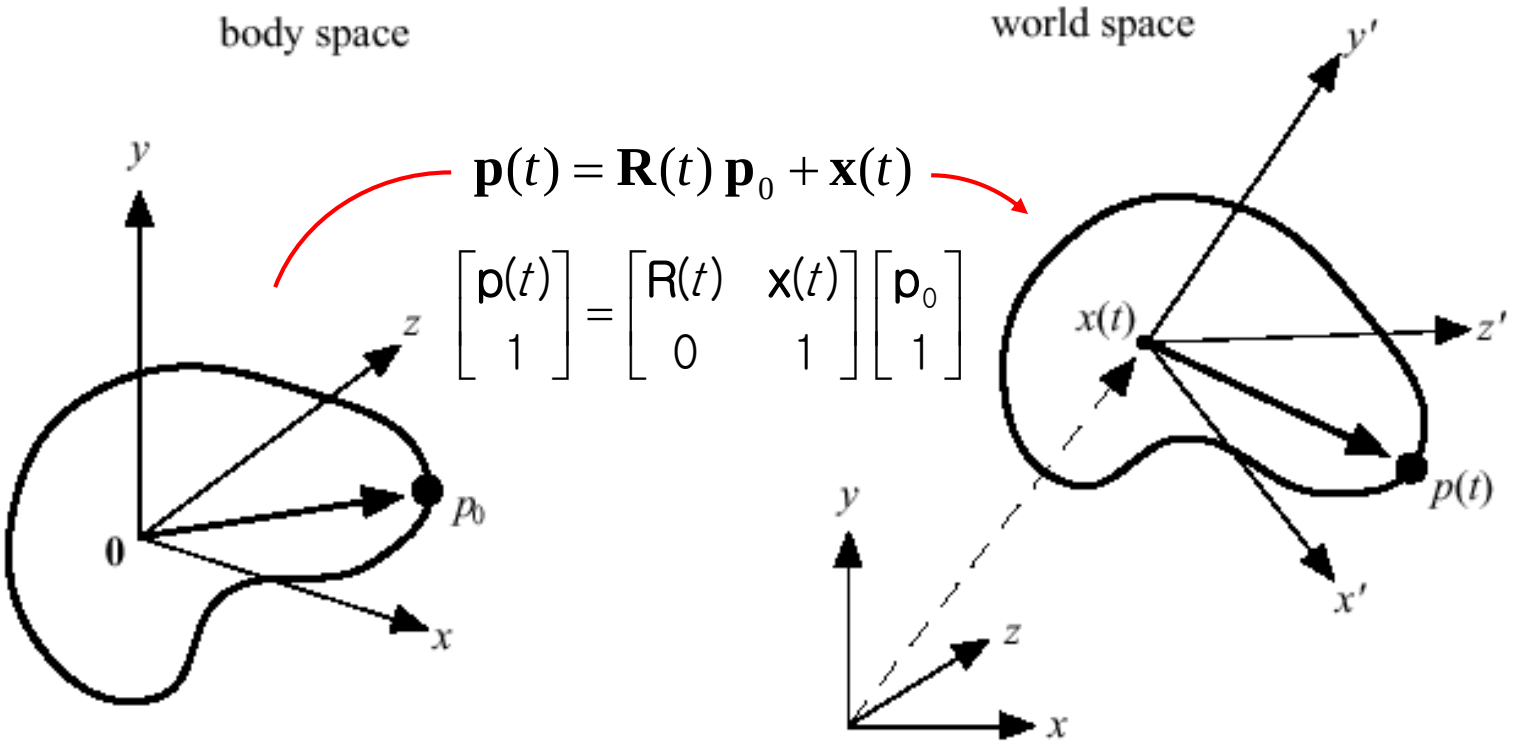
$\mathbf{x}(t)$ - center of mass

$\mathbf{R}(t)$ - orientation (rotation matrix)

Meaning of R(t): columns represent the coordinates of the body space base vectors (1,0,0), (0,1,0), (0,0,1) in world space.

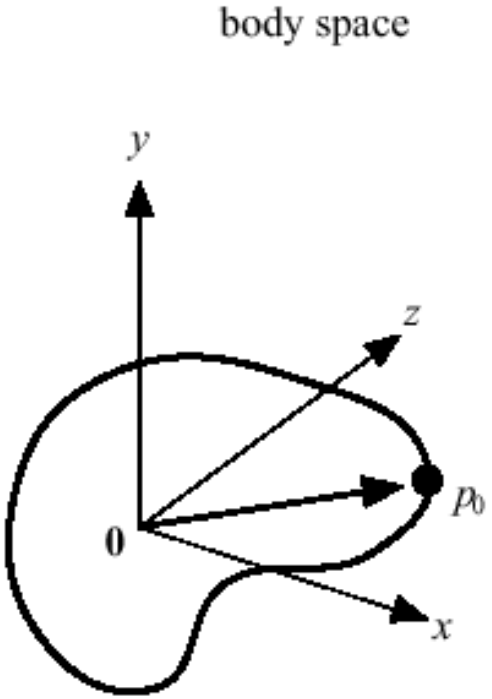
Rigid Bodies

Objects are defined in body space (local coordinate system) and transformed into world space



Body Space

Objects are defined in body space (local coordinate system) and transformed into world space



- bodies are specified relative to this system
- center of mass is the origin

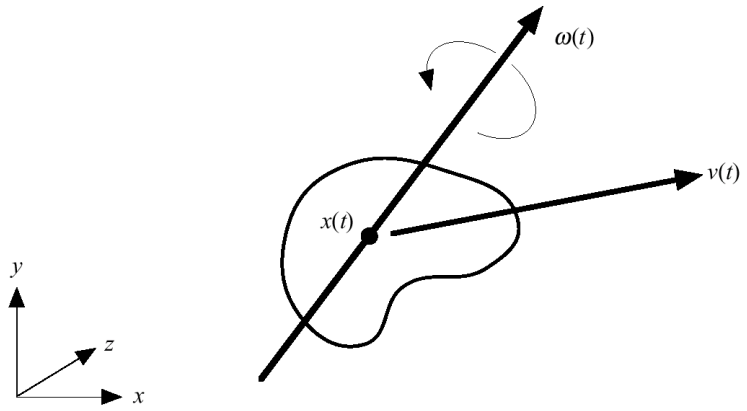
  (for convenience)
  - We will specify body-related physical properties (inertia, …) in this frame

Outline

- Rigid Body Preliminaries
  - velocity, acceleration, and inertia
- State and Evolution
- Collision Detection and Contact Determination
- Colliding Contact Response

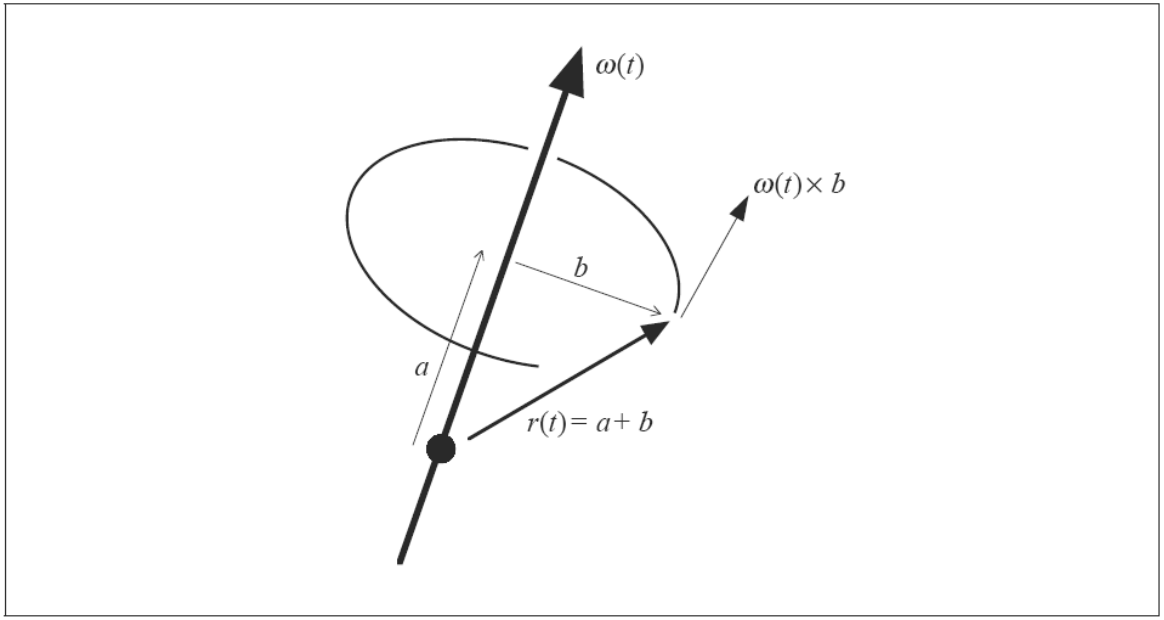
Kinematics: Velocities

- How do x(t) and R(t) change over time?
- Linear velocity v(t) = dx(t)/dt is the same:
  - Describes the velocity of the center of mass (m/s)
- Angular velocity $\omega(t)$ is new!
  - Direction is the axis of rotation
  - Magnitude is the angular velocity about the axis (rad/s)
  - There is a simple relationship between R(t) and $\omega(t)$



Kinematics: Velocities

Then



$$\dot{R} = \left(\omega(t) \times \begin{pmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{pmatrix} \quad \omega(t) \times \begin{pmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{pmatrix} \quad \omega(t) \times \begin{pmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{pmatrix} \right)$$

Angular Velocity

We can represent the cross product with a matrix

$$\mathbf{a} \times \mathbf{b} = \mathbf{a}^{*}\,\mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_x & 0 & -a_x \\ -a_y & a_z & 0 \end{bmatrix} \mathbf{b}$$

Therefore

$$\dot{\mathbf{R}}(t) = \boldsymbol{\omega}^{*}(t)\,\mathbf{R}(t)$$

Velocity of a Point

Since a point can be represented at any time by

$$\mathbf{r}(t) = \mathbf{R}(t)\,\mathbf{r}_0 + \mathbf{x}(t)$$

Total velocity can then be expressed as

$$\dot{\mathbf{r}}(t) = \dot{\mathbf{R}}(t)\,\mathbf{r}_0 + \mathbf{v}(t)$$

Which can be rewritten as

$$\dot{\mathbf{r}}(t) = \boldsymbol{\omega}^{*}\,\mathbf{R}(t)\,\mathbf{r}_0 + \mathbf{v}(t)$$

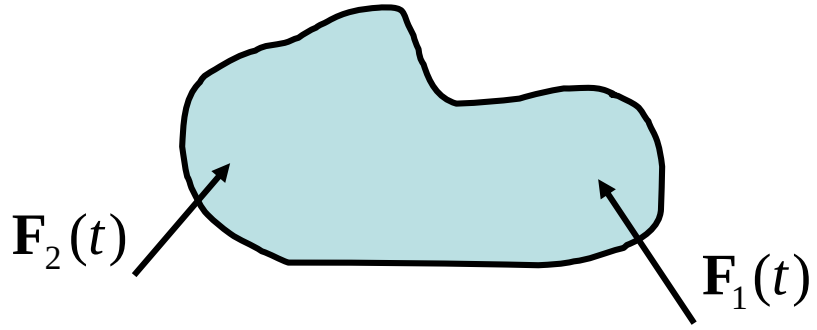
$$\boxed{\dot{\mathbf{r}}(t) = \boldsymbol{\omega} \times (\mathbf{r}(t) - \mathbf{x}(t)) + \mathbf{v}(t)}$$

Dynamics: Accelerations

- How do v(t) and dR(t)/dt change over time?
- First we need some more machinery
  - Forces and Torques
  - Momentums
  - Inertia Tensor
- Simplify equations by formulating accelerations in terms of *momentum derivatives* instead of velocity derivatives

Force

We can apply forces to the object at any point



Total force on an object is simply

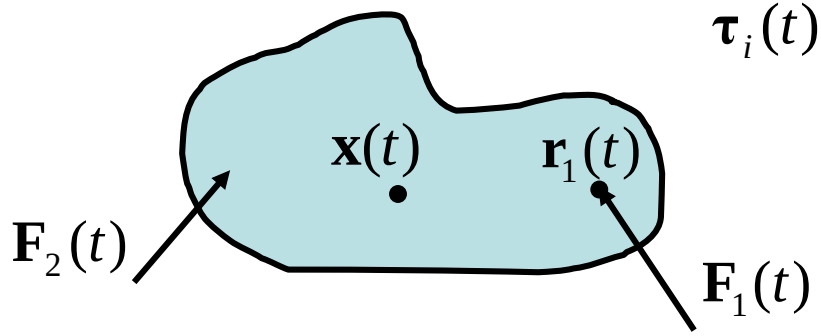
$$\mathbf{F}(t) = \sum \mathbf{F}_i(t)$$

No information about where the forces are applied

Torque

Torque describes the "rotational force"

$$\boldsymbol{\tau}_i(t) = (\mathbf{r}_i(t) - \mathbf{x}(t)) \times \mathbf{F}_i(t)$$



Total torque on an object is simply

$$\boldsymbol{\tau}(t) = \sum \boldsymbol{\tau}_i(t) = \sum (\mathbf{r}_i(t) - \mathbf{x}(t)) \times \mathbf{F}_i(t)$$

Tells us about the force distribution over the object

Forces and Torques

- External forces $F_i(t)$ act on particles
  - Total external force $F=\sum F_i(t)$
- Torques depend on distance from the center of mass:

$$\tau_i(t) = (r_i(t) - x(t)) \times F_i(t)$$

  - Total external torque

$$\tau = \sum ((r_i(t)-x(t)) \times F_i(t)$$

- F(t) doesn't convey any information about where the various forces act
- $\tau(t)$ does tell us about the *distribution* of forces

Linear Momentum

Linear momentum of a particle is

$$\mathbf{p} = m\,\mathbf{v}$$

Linear momentum of a rigid body is then

$$\mathbf{P}(t) = \int \rho\, \dot{\mathbf{r}}(t)\, dV$$

density

integration over the body

Linear Momentum

Linear momentum can be simplified as follows

$$\mathbf{P}(t) = \int \rho\, \dot{\mathbf{r}}(t)\, dV$$

$$\mathbf{P}(t) = \int \rho\, \mathbf{v}(t) + \rho\, \boldsymbol{\omega} \times (\mathbf{r}(t) - \mathbf{x}(t))\, dV$$

$$\mathbf{P}(t) = M\, \mathbf{v}(t)$$

Assuming constant mass gives

$$\dot{\mathbf{P}}(t) = M\, \dot{\mathbf{v}}(t)$$

$$\boxed{\dot{\mathbf{P}}(t) = \mathbf{F}(t)}$$

Just as if body were a particle with mass M and velocity v(t)

Linear Momentum

Linear momentum can be simplified as follows

$$\mathbf{P}(t) = \int \rho\, \dot{\mathbf{r}}(t)\, dV$$

$$\mathbf{P}(t) = \int \rho\, \mathbf{v}(t) + \underline{\rho\, \boldsymbol{\omega} \times (\mathbf{r}(t) - \mathbf{x}(t))}\, dV$$

$$\mathbf{P}(t) = M\, \mathbf{v}(t)$$

This term vanishes because of the definition of COM

Assuming constant mass gives

$$\dot{\mathbf{P}}(t) = M\, \dot{\mathbf{v}}(t)$$

$$\boxed{\dot{\mathbf{P}}(t) = \mathbf{F}(t)}$$

Angular Momentum

Angular momentum of a rigid body

$$\mathbf{L}(t) = \mathbf{I}(t)\boldsymbol{\omega}(t)$$

inertia tensor

Taking the time derivative

$$\dot{\mathbf{L}}(t) = \boldsymbol{\tau}(t)$$

Angular momentum is conserved when there is no torque

Inertia Tensor

Describes how mass is distributed in the body

$$\mathbf{I}(t) = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

$$I_{xx} = \int \rho\,(\hat{y}^2 + \hat{z}^2)\,dV$$

$$I_{xy} = -\int \rho\,(\hat{x}\hat{y})\,dV$$

$$\hat{x} = x - x_{centerOfMass}$$

Analogous to mass in linear velocity
- rotational mass

Measures the preferred axis of rotation

Expensive to compute this at every time step

Inertia Tensor

Rewrite the tensor as

$$\mathsf{I}(t) = \int \rho \, (\hat{\mathsf{r}}(t) \cdot \hat{\mathsf{r}}(t) \, \mathsf{I} - \hat{\mathsf{r}}(t) \, \hat{\mathsf{r}}(t)^T) \, dV$$

$$\mathsf{I}(t) = \int \rho \, ((\mathsf{R}(t)\mathsf{r}_0)^T \mathsf{R}(t)\mathsf{r}_0 \, \mathsf{I} - \mathsf{R}(t)\mathsf{r}_0 (\mathsf{R}(t)\mathsf{r}_0)^T) \, dV$$

$$\mathsf{I}(t) = \int \rho \, (\mathsf{R}(t)(\mathsf{r}_0^T \mathsf{r}_0 \mathsf{I} - \mathsf{r}_0 \mathsf{r}_0^T)\mathsf{R}^T(t)) \, dV$$

$$\boxed{\mathbf{I}(t) = \mathbf{R}(t) \, \mathbf{I}_{body} \, \mathbf{R}^T(t)}$$

Integrals can now be precomputed

Outline

- Rigid Body Preliminaries
- State and Evolution
  - Variables and derivatives
- Quaternions
- Collision Detection and Contact Determination
- Colliding Contact Response

New State Space

$$\mathbf{X}(t) = \begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix}$$

Spatial information

Velocity information

v(t) replaced by linear momentum P(t)

ω(t) replaced by angular momentum L(t)

Size of the vector: (3+9+3+3)N = 18N

Taking the Derivative

$$\frac{d}{dt}\mathbf{X}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ \omega(t)^* R(t) \\ F(t) \\ \tau(t) \end{pmatrix}$$

Conservation of momentum (P(t), L(t)) lets us express the accelerations in terms of forces and torques.

Discretize these continuous equations and integrate

Simulate: next state computation

- From $X(t)$ certain quantities are computed

$$\begin{cases} I^{-1}(t) = R(t)\, I_{body}^{-1}\, R^T(t) \\ v(t) = P(t) / M \\ \omega(t) = I^{-1}(t)\, L(t) \end{cases}$$

- We must be content with a finite number of discrete time points
- Use your favorite ODE solver to solve for the new state X(t), given previous state X(t-Δt) and applied forces F(t) and τ(t)

$$X(t) = \text{Solver::Step}(X(t-\Delta\, t), F(t), \tau\, (t))$$

Simple simulation algorithm

X = InitializeState()

For $t=t_0$ to t_{final} with step Δ t

  ClearForces(F(t), τ(t))

  AddExternalForces(F(t), τ(t))

  X_{new} =Solver::Step(X, F(t), τ(t))

  X =X_{new}

  t = t + Δt

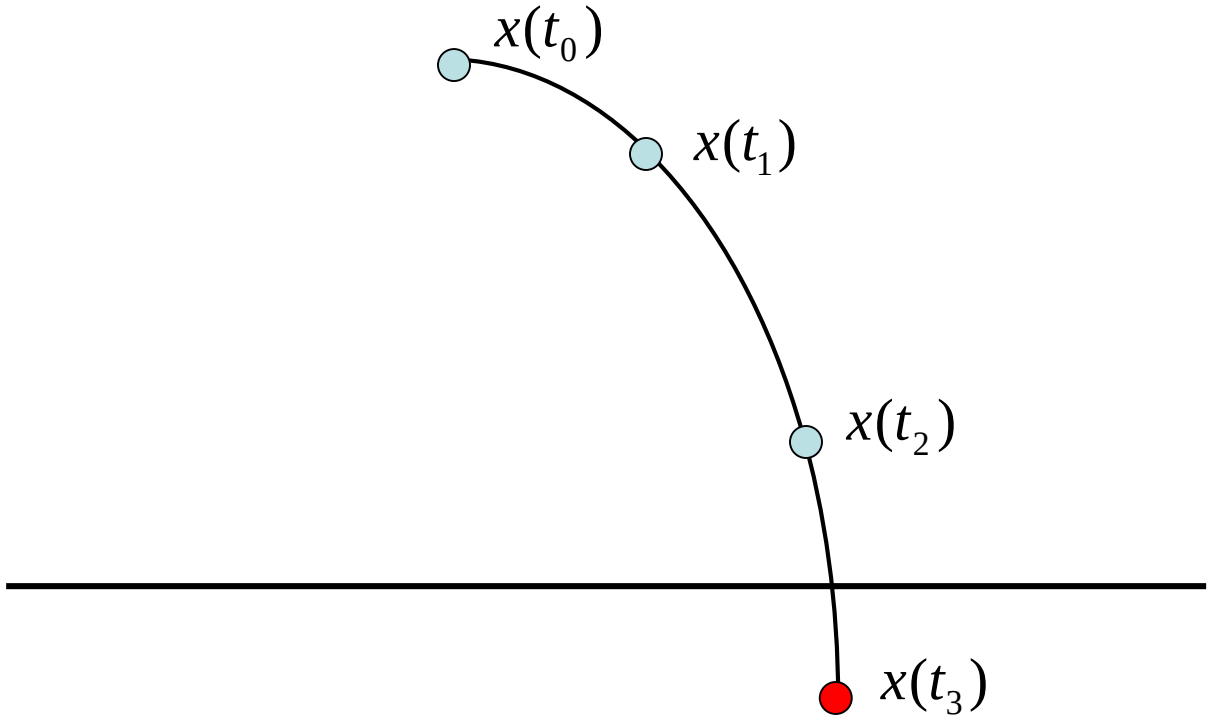
End for

Outline

- Rigid Body Preliminaries
- State and Evolution
- Collision Detection and Contact Determination
  - Contact classification
  - Intersection testing, bisection, and nearest features
- Colliding Contact Response

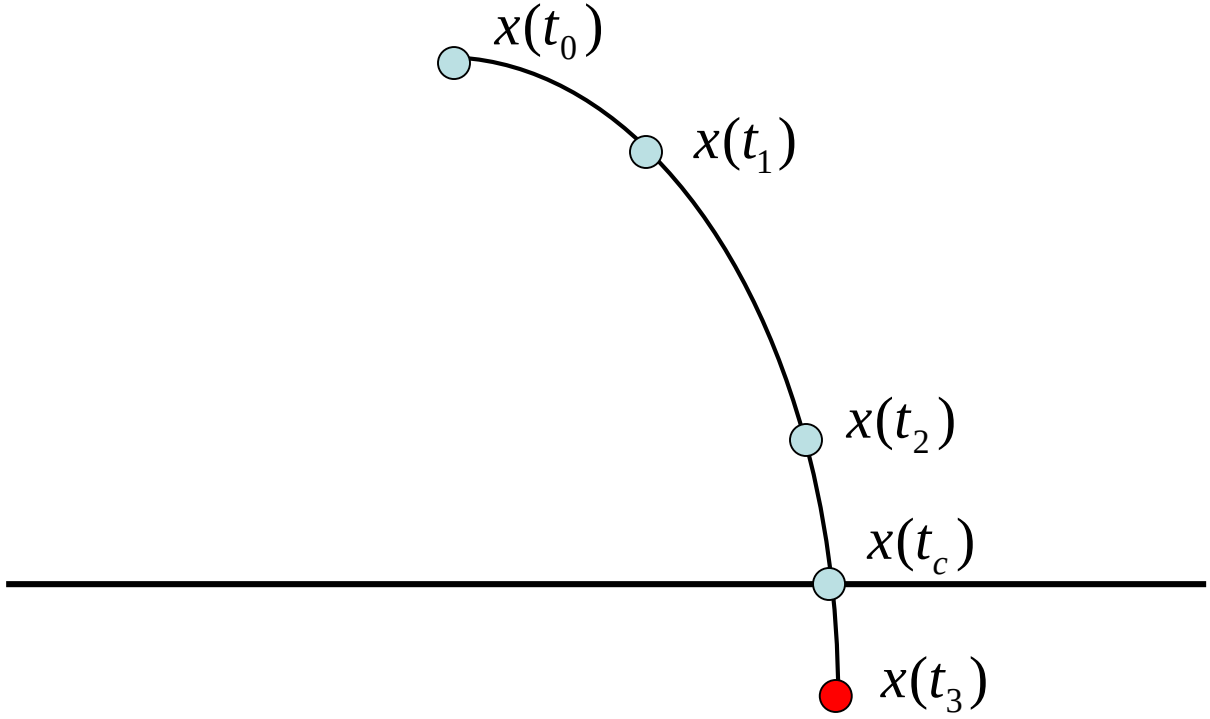
Collisions and Contact

What should we do when there is a collision?



Rolling Back the Simulation

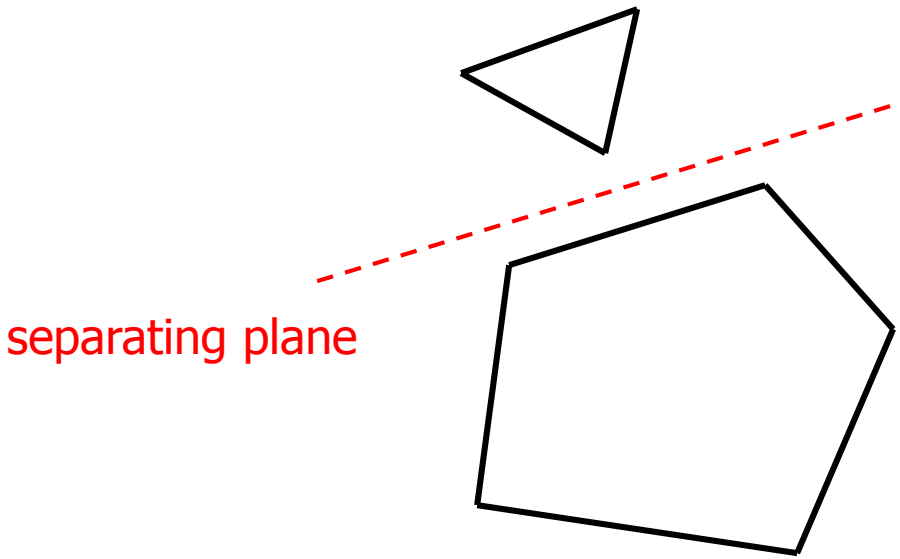
Restart the simulation at the time of the collision



Collision time can be found by bisection, etc.

Collision Detection

Exploit coherency through witnessing



Two convex objects are non-penetrating iff there exists a separating plane between them

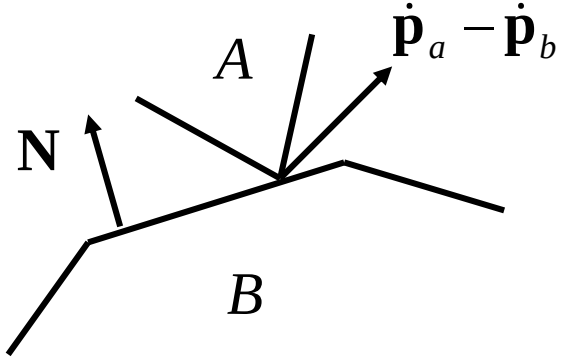
First find a separating plane and see if it is still valid after the next simulation step

Speed up with bounding boxes, grids, hierarchies, etc.

Collision Detection

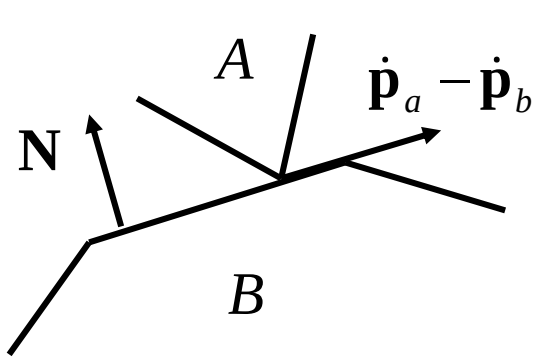
Conditions for collision

$$\dot{\mathbf{p}}_a(t) = \mathbf{v}_a(t) + \boldsymbol{\omega}_a(t) \times (\mathbf{p}_a(t) - \mathbf{x}_a(t))$$



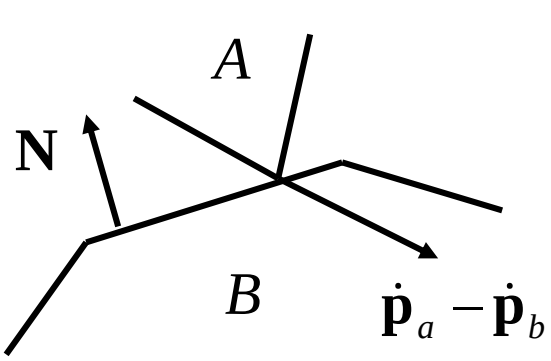
$\mathbf{N} \cdot (\dot{\mathbf{p}}_a(t) - \dot{\mathbf{p}}_b(t)) > 0$

separating



$\mathbf{N} \cdot (\dot{\mathbf{p}}_a(t) - \dot{\mathbf{p}}_b(t)) = 0$

contact

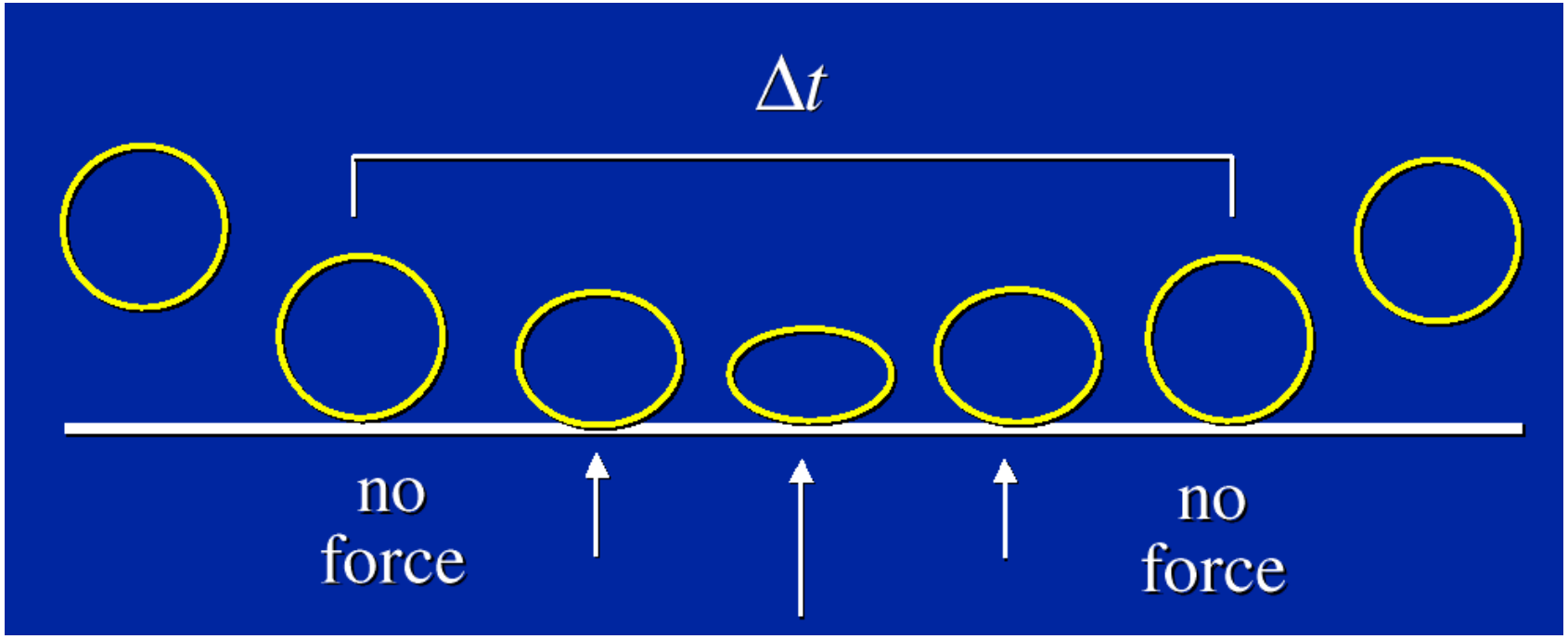


$\mathbf{N} \cdot (\dot{\mathbf{p}}_a(t) - \dot{\mathbf{p}}_b(t)) < 0$

colliding

Collision

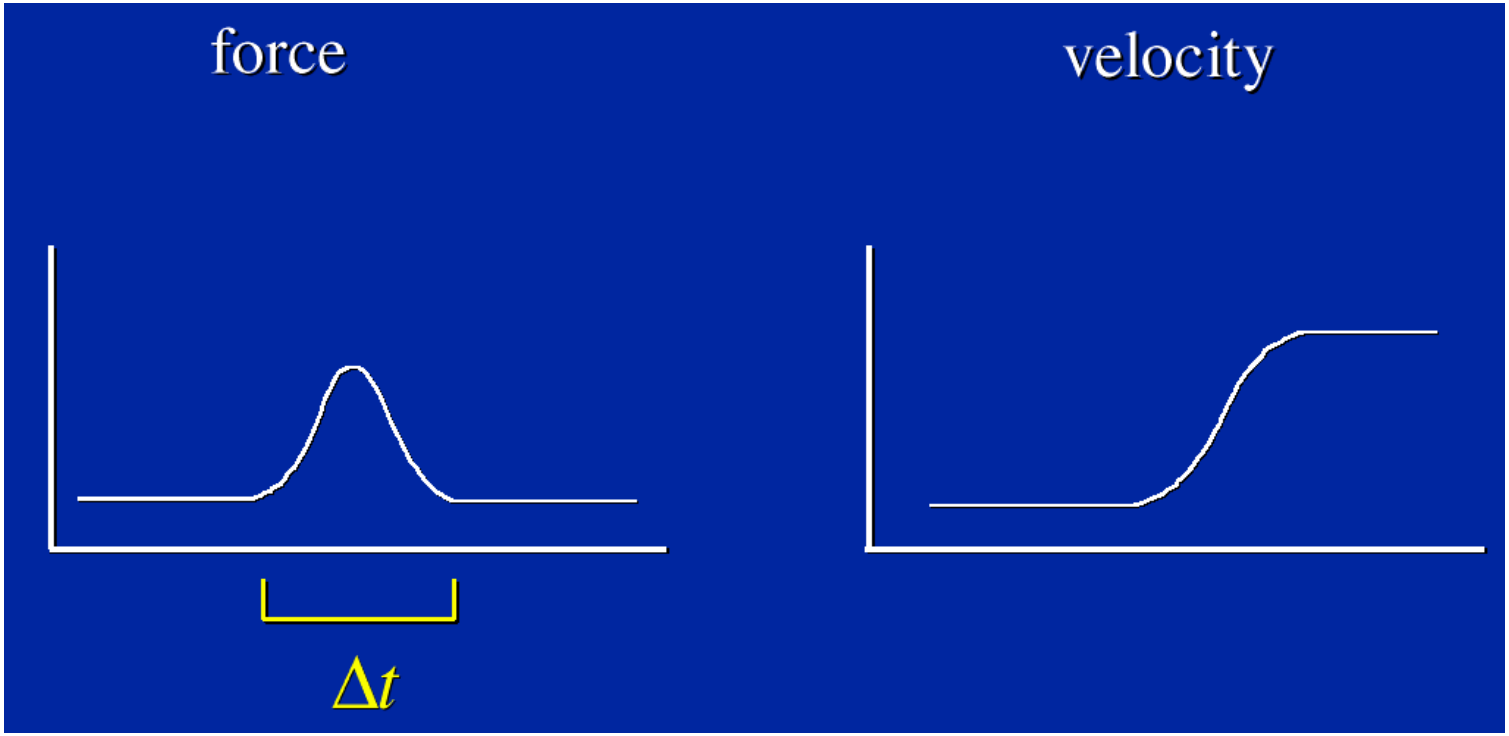
Soft Body Collision



Force is applied to prevent interpenetration

Collision

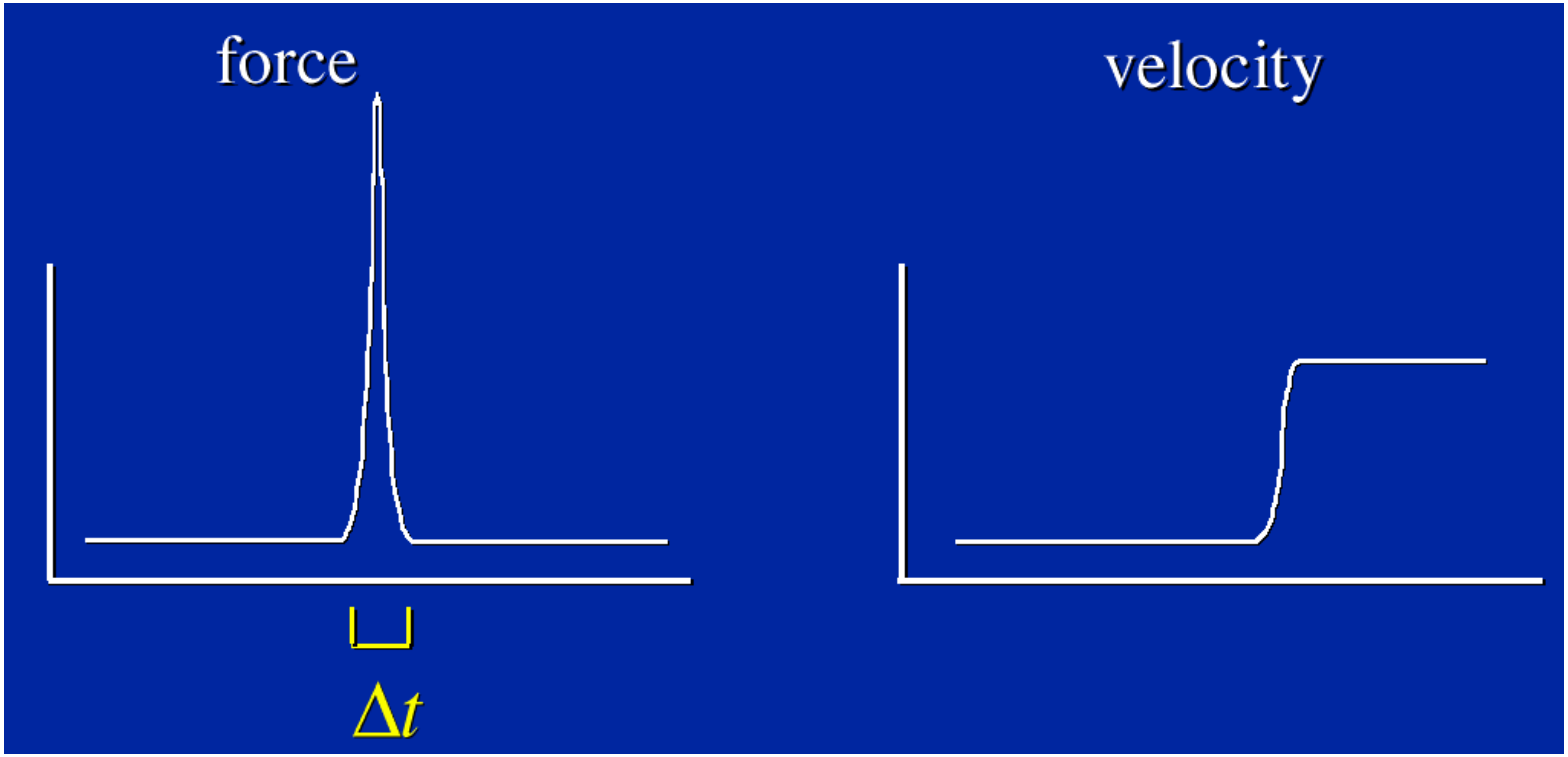
Soft Body Collision



Apply forces and change the velocity

Collision

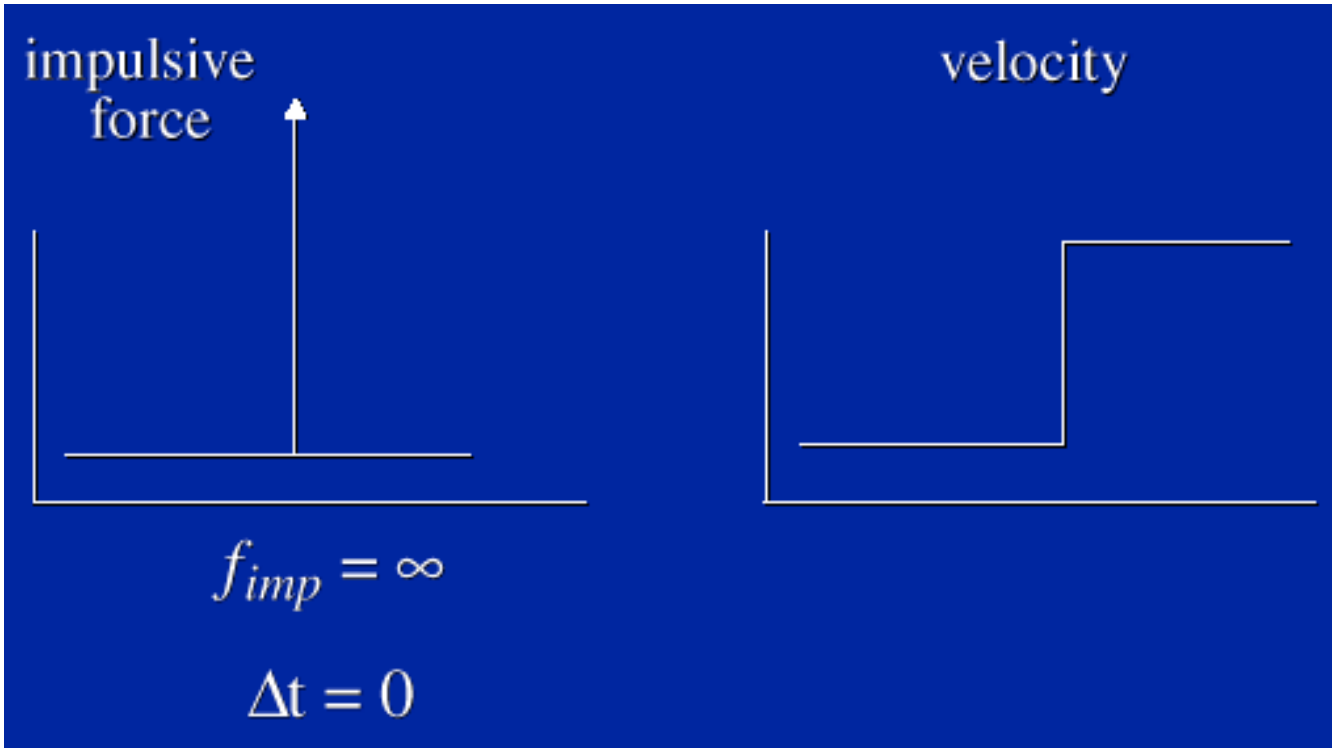
Harder Collision



Higher force over a shorter time

Collision

Rigid Body Collision



Impulsive force produces a discontinuous velocity

Impulse

We need to change velocity instantaneously

Infinite force in an infinitesimal time

$$\mathbf{J} = \mathbf{F}\,\Delta t$$

An impulse changes the velocity as

$$\Delta \mathbf{v} = \frac{\mathbf{J}}{M} \qquad \text{or} \qquad \Delta \mathbf{P} = \mathbf{J}$$

Impulse

An impulse also creates an impulsive torque

$$\boldsymbol{\tau}_{impulse} = (\mathbf{p}(t) - \mathbf{x}(t)) \times \mathbf{J}$$

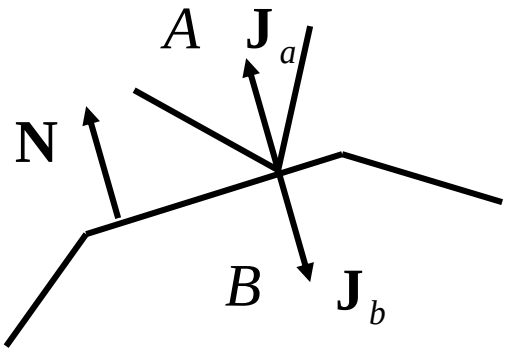
The impulsive torque changes the angular velocity

$$\Delta\boldsymbol{\omega} = \mathbf{I}^{-1}(t)\,\boldsymbol{\tau}_{impulse} \quad \text{or} \quad \Delta\mathbf{L} = \boldsymbol{\tau}_{impulse}$$

Impulse

For a frictionless collision

$$\mathbf{J} = j\,\mathbf{N}$$



But how do we calculate j ?

Impulse

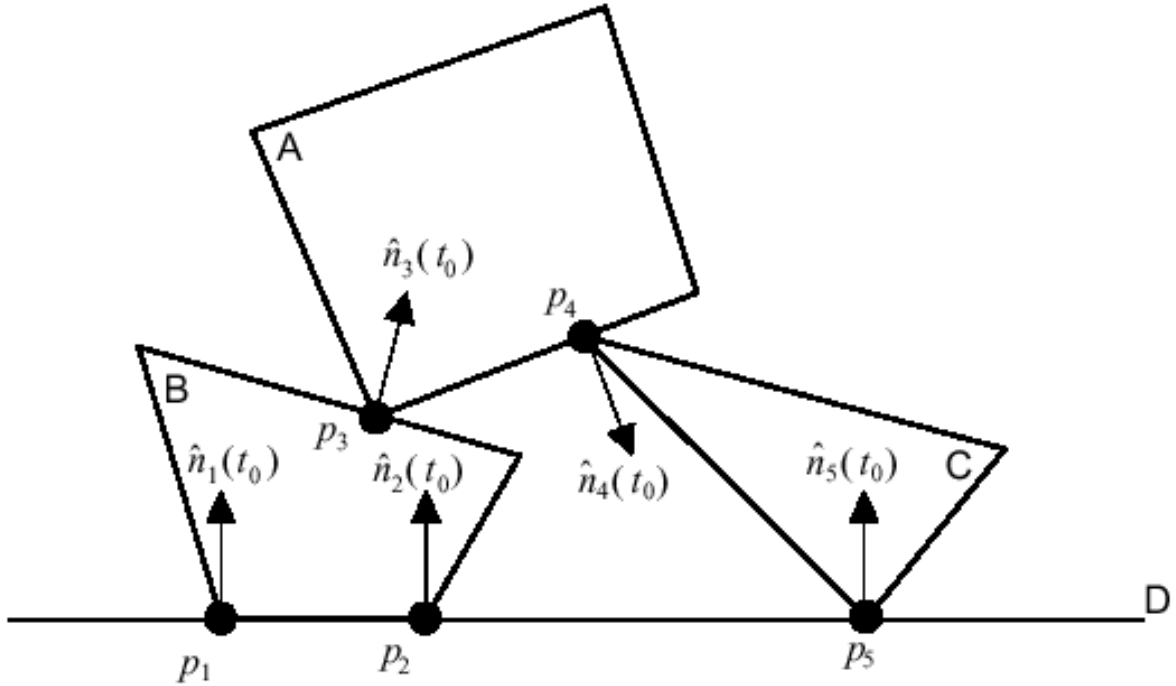
For a frictionless collision

$$\mathbf{N} \cdot (\dot{\mathbf{p}}_a^+(t) - \dot{\mathbf{p}}_b^+(t)) = -\varepsilon \, (\mathbf{N} \cdot (\dot{\mathbf{p}}_a^-(t) - \dot{\mathbf{p}}_b^-(t)))$$

Given this equation and knowing how j affects the linear and angular velocities of the two bodies, we can solve for j

Resting Contact

Bodies are neither colliding nor separating



We want a force strong enough to resist penetration but only enough to maintain contact

Resting Contact

We want to prevent interpenetration

$$d(t) = \mathbf{N} \cdot (\mathbf{p}_a(t) - \mathbf{p}_b(t)) \geq 0$$

Since $d(t_c) = 0$ we should keep it from decreasing

$$\dot{d}(t) = \dot{\mathbf{N}} \cdot (\mathbf{p}_a(t) - \mathbf{p}_b(t)) + \mathbf{N} \cdot (\dot{\mathbf{p}}_a(t) - \dot{\mathbf{p}}_b(t)) \geq 0$$

Since $\dot{d}(t_c) = 0$ we should keep it from decreasing

$$\ddot{d}(t_c) = \mathbf{N} \cdot (\ddot{\mathbf{p}}_a(t_c) - \ddot{\mathbf{p}}_b(t_c)) + 2\,\dot{\mathbf{N}} \cdot (\dot{\mathbf{p}}_a(t_c) - \dot{\mathbf{p}}_b(t_c)) \geq 0$$

Describes the objects' acceleration towards one another

Resting Contact

Contact forces only act in the normal direction

$$\mathbf{F}_c = f\ \mathbf{N}(t_c)$$

Contact forces should

- avoid interpenetration

$$\ddot{d}(t_c) \geq 0$$

- be repulsive

$$f \geq 0$$

- become zero if the bodies begin to separate

$$f \cdot \ddot{d}(t_c) = 0$$

workless force

Resting Contact

The relative accelerations can be written in terms of all of the contact forces

$$\ddot{d}_i(t_c) = a_0 f_0 + \ldots + a_n f_n + b_i$$

So we can simply solve a Quadratic Program to find the solution to all the constraints

Simulation Algorithm

Algorithm with collisions and contact