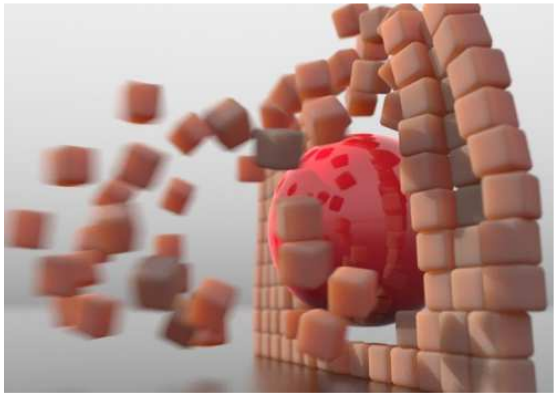
Computer Animation & Physical Simulation

Lecture 6: Rigid Body Simulation I

XIAOPEI LIU
School of Information Science and Technology
ShanghaiTech University

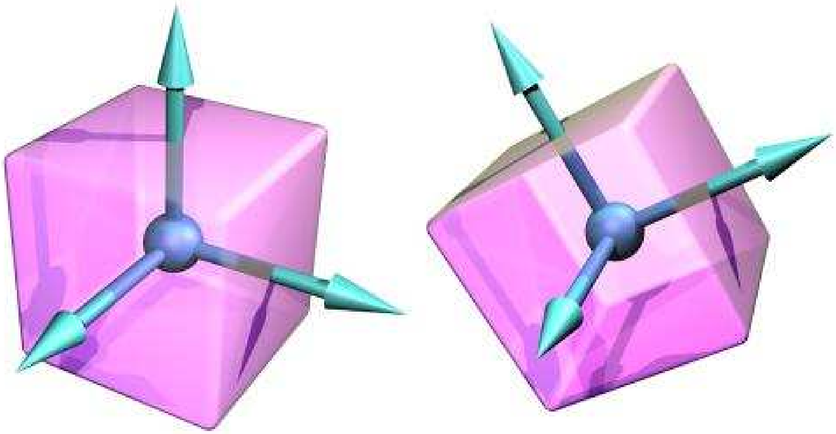
Rigid Body

- **What is a rigid body?**
  - The body never deforms (ideal)
  - The distance between any two given points of a rigid body remains constant in time regardless of any external forces



Rigid Body Motion

- **Motion due to external forces**
  - Translation
  - Rotation



I. Particle System

Particle System

- **Description of particle state**
  - Each particle is described by position and velocity

$$\mathbf{Y}(t) = \begin{pmatrix} x(t) \\ v(t) \end{pmatrix}$$

  - For a particle system with n particles

$$\mathbf{Y}(t) = \begin{pmatrix} x_1(t) \\ v_1(t) \\ \vdots \\ x_n(t) \\ v_n(t) \end{pmatrix}$$

Particle Systems

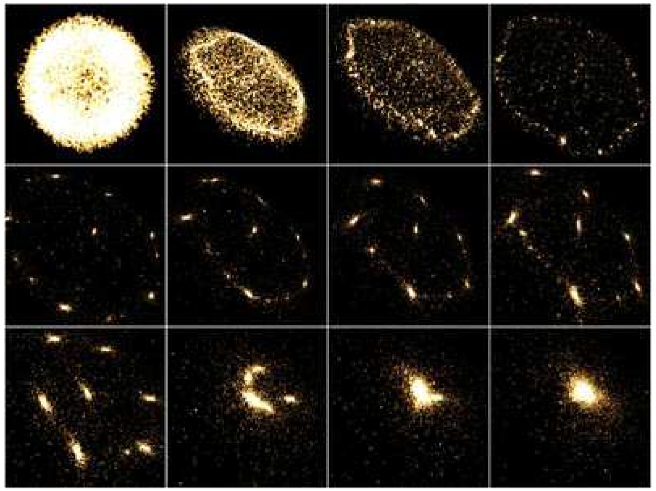
- **Dynamic system of particles**
  - For each particle
    - A force **F**(t) acting on it
    - A mass m associated with it

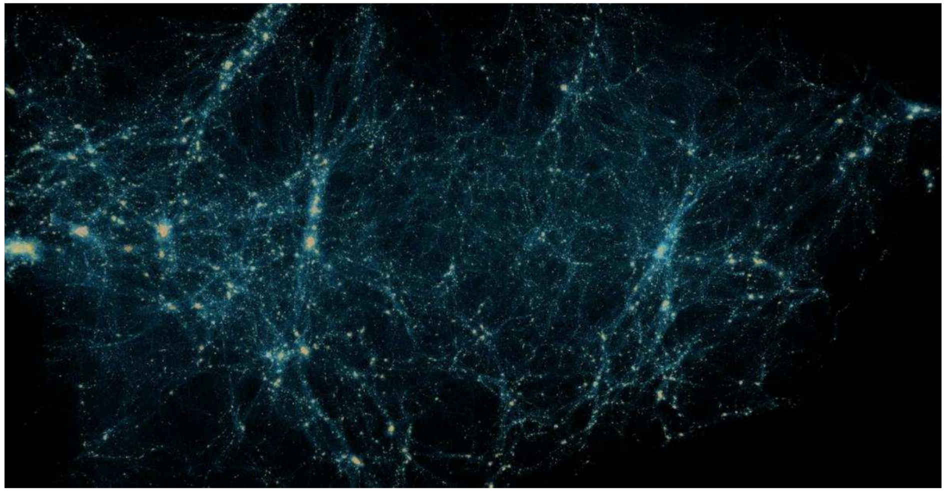
  - Dynamic equation by ordinary differential equations

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ v(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ F(t)/m \end{pmatrix}$$

N-Body Simulation

- **A simulation of a dynamical system of particles**
  - under the influence of physical forces

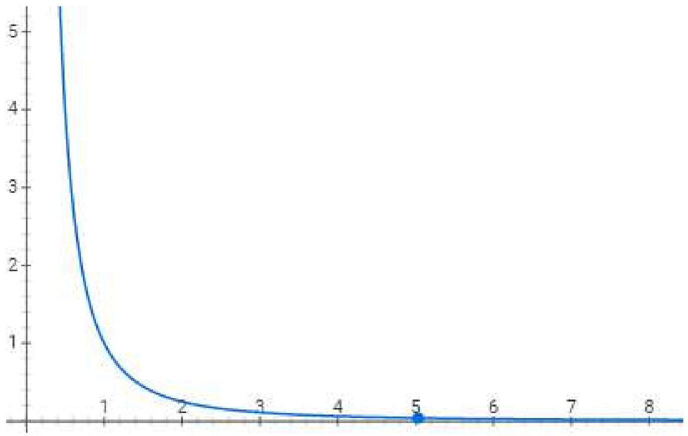




N-Body Simulation

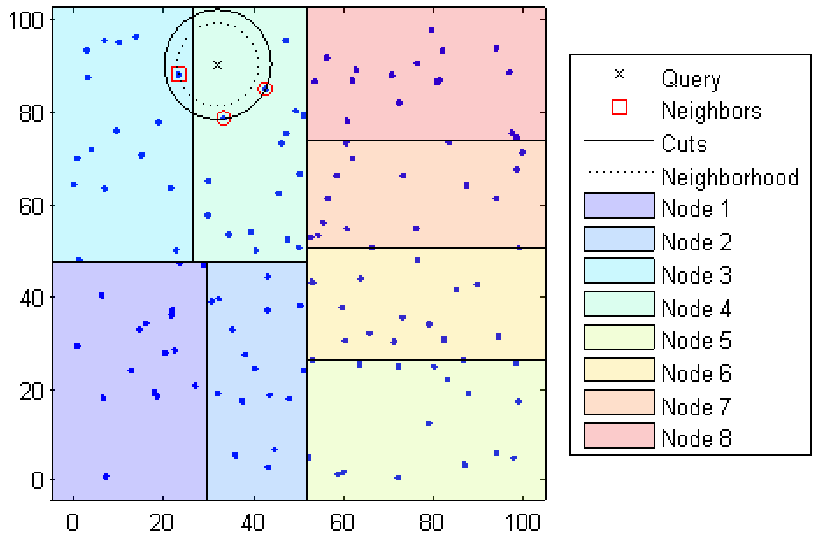
- **Formulation**

$$\ddot{\mathbf{r}}_i = -G \sum_{j=1;\, j\neq i}^{N} \frac{m_j(\mathbf{r}_i - \mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|^3}$$



N-Body Simulation

- **Fixed Radius Nearest Neighbor Search**



Solving Ordinary Differential Equations

- **Finite difference method for time evolution**
  - Ordinary set of differential equation of the form:

$$y' = f(x, y)$$

    - Seldom have closed-form solution

  - Usually with initial condition $y(x_0) = y_0$

  - Initial value problem $\mathbf{y}' = \mathbf{f}(x, \mathbf{y})\ ,\qquad \mathbf{y}(x_0) = \mathbf{y}_0$

Solving Ordinary Differential Equations

- **Numerical solution**
  - Euler's method
    - We divide this interval by the mesh-points

$$x_n = x_0 + nh,\ n = 0, \ldots, N$$

$$y' = f(x, y)$$

$$\Downarrow$$

- Integrating the differential equation

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x))\, \mathrm{d}x$$

$$\Big\Downarrow \quad \int_{x_n}^{x_{n+1}} g(x)\, \mathrm{d}x \approx h g(x_n)$$

$$y(x_{n+1}) \approx y(x_n) + h f(x_n, y(x_n))$$

Solving Ordinary Differential Equations

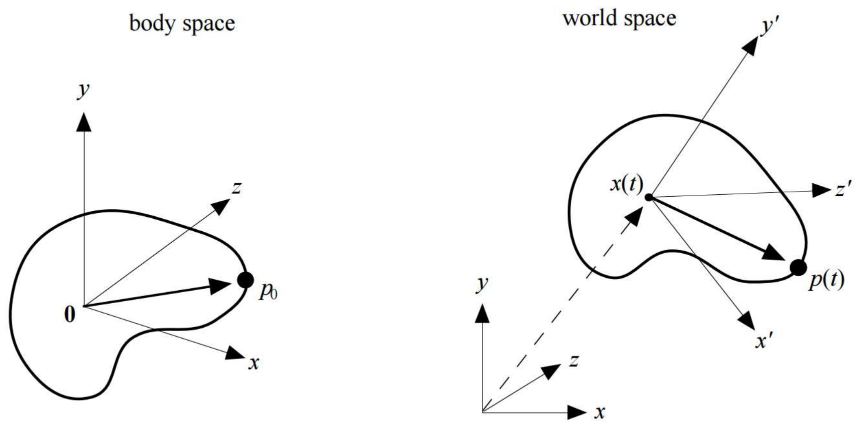
- **Runge–Kutta methods**
  - Achieve higher accuracy
  - Re-evaluate f(·, ·) at points intermediate between $(x_n, y(x_n))$ and $(x_n+1, y(x_n+1))$

$$
\begin{aligned}
y_{n+1} &= y_n + h\Phi(x_n, y_n; h) \ , \\
\Phi(x, y; h) &= \sum_{r=1}^{R} c_r k_r \ , \\
k_1 &= f(x, y) \ , \\
k_r &= f\left(x + ha_r, y + h\sum_{s=1}^{r-1} b_{rs} k_s\right) \ , \quad r = 2, \ldots, R \ , \\
a_r &= \sum_{s=1}^{r-1} b_{rs} \ , \quad r = 2, \ldots, R \ .
\end{aligned}
$$

II. Unconstrained Rigid Body Dynamics

Position and Orientation

- **World space and body space**
  - World space: a global space which does not change
  - Body space: a space relative to the body; the coordinate frame can be translated and rotated

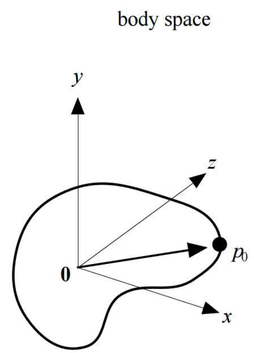


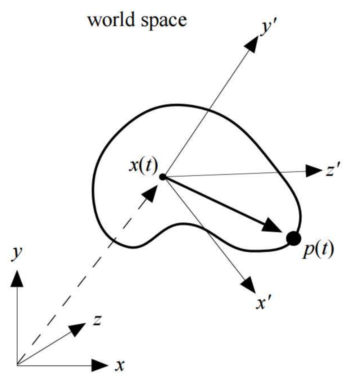
Position and Orientation

- **Connection between body space and world space**
  - Body space origin is usually defined at the center of mass
  - Transformation between body space and world space:

$$p(t) = R(t)p_0 + x(t)$$

We call *x(t)* and *R(t)* the position and orientation of the body

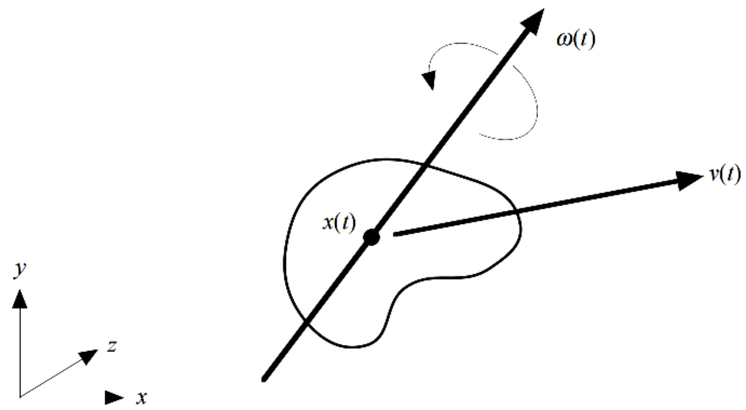




Linear & Angular Velocity

- **Definition of linear velocity**
  - The linear velocity v(t) $v(t) = \dot{x}(t)$

- **Definition of an angular velocity**
  - An axis the body rotates about
  - The speed of the rotation



Calculation of Rotation

- **Given r(t) in world coordinates**
  - Decomposition of r(t)

$$r(t) = a + b$$

  - Instant velocity

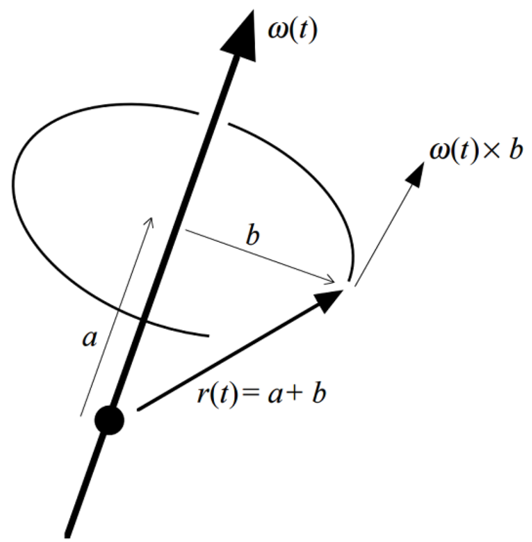
$$v(t) = \omega(t) \times b$$

$$\Downarrow$$

$$\dot{r}(t) = \omega(t) \times b = \omega(t) \times b + \omega(t) \times a = \omega(t) \times (b + a)$$

$$\Downarrow$$

$$\dot{r}(t) = \omega(t) \times r(t)$$

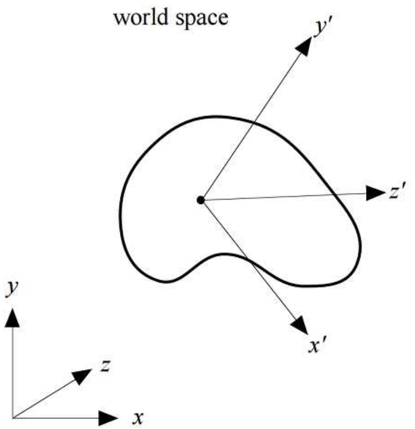


Calculation of Rotation

- **Rotating a body coordinate frame**

$$R(t) = \begin{pmatrix} r_{xx} & r_{yx} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{pmatrix} \Rightarrow R(t)\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{pmatrix}$$

$$R(t) = [\, x' \;\; y' \;\; z' \,]$$



Calculation of Rotation

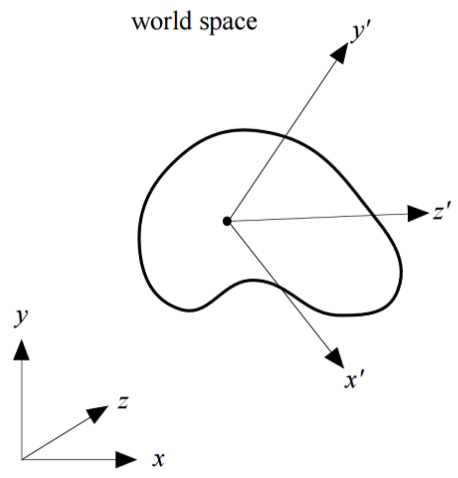
- **Apply the angular velocity to the body frame after rotation**

$$\dot{r}(t) = \omega(t) \times r(t)$$

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

$$a^* b = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix} \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} a_y b_z - b_y a_z \\ -a_x b_z + b_x a_z \\ a_x b_y - b_x a_y \end{pmatrix} = a \times b$$

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



Mass of a Body

- **Particle assumption**
  - Imagine that a rigid body is made up of a large number of small particles
  - The location of the i-th particle in world space at time t:

$$r_i(t) = R(t)r_{0i} + x(t)$$

  - The total mass of the body, M, is the sum

$$M = \sum_{i=1}^{N} m_i$$

Velocity of a Particle

- **The velocity of the i-th particle**

$$r_i(t) = R(t)r_{0i} + x(t)$$

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

$$v(t) = \dot{x}(t)$$

$$\xrightarrow{\text{differentiation}}$$

$$\dot{r}_i(t) = \omega^* R(t)r_{0i} + v(t)$$

$$\Downarrow$$

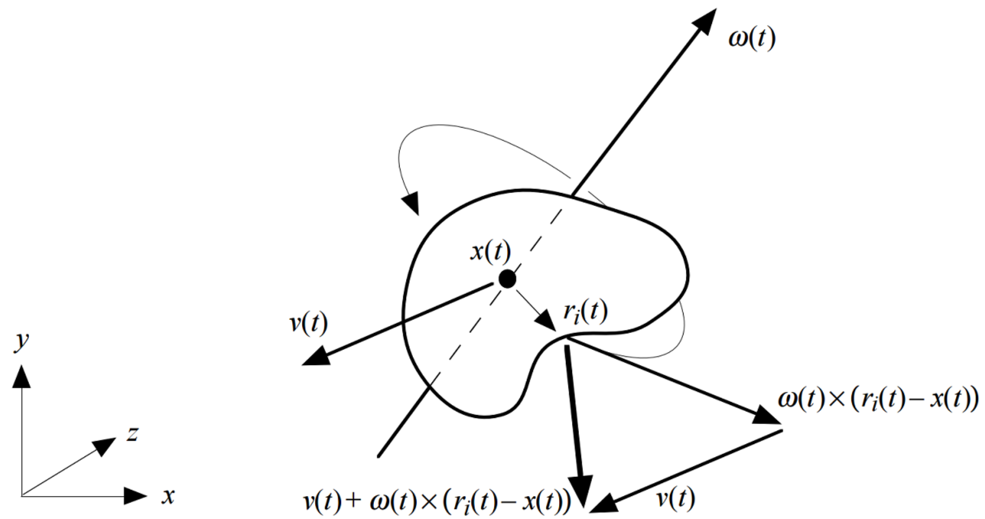
$$\begin{aligned}\dot{r}_i(t) &= \omega(t)^* R(t)r_{0i} + v(t) \\ &= \omega(t)^* \left(R(t)r_{0i} + x(t) - x(t)\right) + v(t) \\ &= \omega(t)^* \left(r_i(t) - x(t)\right) + v(t)\end{aligned}$$

$$\Downarrow$$

$$\dot{r}_i(t) = \omega(t) \times \left(r_i(t) - x(t)\right) + v(t)$$

Velocity of a Particle

- **Illustration of particle velocity**
  - The velocity can be decomposed into a linear term and an angular



Center of Mass

- **The center of mass of a body**
  - In world space (definition)

$$\frac{\sum m_i r_i(t)}{M}$$

  - In body space

$$\frac{\sum m_i r_{0i}}{M} = \mathbf{0} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Center of Mass

- **x(t) as being the location of the center of mass?**
  - Yes

$$\frac{\sum m_i r_i(t)}{M} = \frac{\sum m_i(R(t)r_{0i} + x(t))}{M} = \frac{R(t)\sum m_i r_{0i} + \sum m_i x(t)}{M} = x(t)\frac{\sum m_i}{M} = x(t)$$

  - In addition

$$\sum m_i(r_i(t) - x(t)) = \sum m_i(R(t)r_{0i} + x(t) - x(t)) = R(t)\sum m_i r_{0i} = \mathbf{0}$$

Force and Torque

- **At each particle**
  - A force $F_i(t)$ may be exerted on it
  - A torque may be generated due to $F_i(t)$

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

- **For the whole body**
  - Total force $F(t) = \sum F_i(t)$
  - Total torque $\tau(t) = \sum \tau_i(t) = \sum (r_i(t) - x(t)) \times F_i(t)$

Linear Momentum

- **The linear momentum p of a particle**
  - Defined with mass m and velocity v

$$p = mv$$

- **The total linear momentum P(t)**
  - The sum of the products of the mass and velocity of each particle

$$\dot{r}_i(t) = \omega(t) \times (r_i(t) - x(t)) + v(t)$$

$$\begin{aligned} P(t) &= \sum m_i \dot{r}_i(t) \\ &= \sum \left(m_i v(t) + m_i \omega(t) \times (r_i(t) - x(t)) \right) \\ &= \sum m_i v(t) + \omega(t) \times \sum m_i \left(r_i(t) - x(t) \right) \end{aligned}$$

$$\sum m_i (r_i(t) - x(t)) = \mathbf{0}$$

$$\Downarrow$$

$$P(t) = \sum m_i v(t) = \left(\sum m_i \right) v(t) = Mv(t)$$

Linear Momentum

- **Linear momentum is irrespective of rotation of the body**
  - Linear acceleration

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

  - Relation to total force

$$\dot{P}(t) = F(t) \qquad \dot{v}(t) = \frac{F(t)}{M}$$

Angular Momentum

- **Why consider angular momentum?**
  - Conserved unless there is external torque
  - Let you to write simpler equations
- **Analogous to linear momentum**
  - Linear momentum $P(t) = Mv(t)$
  - Angular momentum $L(t) = I(t)\omega(t)$

    Inertia tensor: 3x3 matrix
  - Relationship between angular momentum and the total torque

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

Analogous to linear momentum relation: $\dot{P}(t) = F(t)$

Inertia Tensor

- **Intrinsic property of a body**
  - Determine the torque needed for a desired angular acceleration
  - Depend on the body's mass distribution and the axis chosen
- **Definition by discrete particles**
  - Let r_i' be the displacement of the i-th particle from x(t)

$$I(t) = \sum \begin{pmatrix} m_i(r'^2_{iy} + r'^2_{iz}) & -m_i r'_{ix} r'_{iy} & -m_i r'_{ix} r'_{iz} \\ -m_i r'_{iy} r'_{ix} & m_i(r'^2_{ix} + r'^2_{iz}) & -m_i r'_{iy} r'_{iz} \\ -m_i r'_{iz} r'_{ix} & -m_i r'_{iz} r'_{iy} & m_i(r'^2_{ix} + r'^2_{iy}) \end{pmatrix} \qquad r'_i = r_i(t) - x(t)$$

  - Continuous distribution: sum to integral, mass to density

Shall we re-compute when rotated?

Inertia Tensor

- **Transformation of *I(t)***

$$I(t)=\sum\begin{pmatrix} m_i(r'^2_{iy}+r'^2_{iz}) & -m_i r'_{ix}r'_{iy} & -m_i r'_{ix}r'_{iz} \\ -m_i r'_{iy}r'_{ix} & m_i(r'^2_{ix}+r'^2_{iz}) & -m_i r'_{iy}r'_{iz} \\ -m_i r'_{iz}r'_{ix} & -m_i r'_{iz}r'_{iy} & m_i(r'^2_{ix}+r'^2_{iy}) \end{pmatrix} \quad + \quad r'^T_i r'_i = r'^2_{ix}+r'^2_{iy}+r'^2_{iz}$$

$$I(t)=\sum m_i r'^T_i r'_i \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} m_i r'^2_{ix} & m_i r'_{ix}r'_{iy} & m_i r'_{ix}r'_{iz} \\ m_i r'_{iy}r'_{ix} & m_i r'^2_{iy} & m_i r'_{iy}r'_{iz} \\ m_i r'_{iz}r'_{ix} & m_i r'_{iz}r'_{iy} & m_i r'^2_{iz} \end{pmatrix} \quad + \quad r'_i r'^T_i = \begin{pmatrix} r'^2_{ix} & r'_{ix}r'_{iy} & r'_{ix}r'_{iz} \\ r'_{iy}r'_{ix} & r'^2_{iy} & r'_{ix}r'_{iz} \\ r'_{iz}r'_{ix} & r'_{iz}r'_{iy} & r'^2_{iz} \end{pmatrix}$$

$$I(t)=\sum m_i((r'^T_i r'_i)\mathbf{1}-r'_i r'^T_i)$$

Inertia Tensor

- **Transformation of *I(t)***

$$r_i(t) = R(t)r_{0i} + x(t) \quad r_i' = R(t)r_{0i}$$

$$R(t)R(t)^T = \mathbf{1}$$

$$I(t) = \sum m_i(({r_i'}^T r_i')\mathbf{1} - r_i' {r_i'}^T)$$

+

$$\begin{aligned}
I(t) &= \sum m_i(({r_i'}^T r_i')\mathbf{1} - r_i' {r_i'}^T) \\
&= \sum m_i((R(t)r_{0i})^T (R(t)r_{0i})\mathbf{1} - (R(t)r_{0i})(R(t)r_{0i})^T) \\
&= \sum m_i({r_0}_i^T R(t)^T R(t) r_{0i}\mathbf{1} - R(t) r_{0i} {r_0}_i^T R(t)^T) \\
&= \sum m_i(({r_0}_i^T r_{0i})\mathbf{1} - R(t) r_{0i} {r_0}_i^T R(t)^T) \\
&= \sum m_i(R(t)({r_0}_i^T r_{0i}) R(t)^T \mathbf{1} - R(t) r_{0i} {r_0}_i^T R(t)^T) \\
&= R(t) \boxed{\left(\sum m_i(({r_0}_i^T r_{0i})\mathbf{1} - r_{0i} {r_0}_i^T)\right)} R(t)^T
\end{aligned}$$

Constant, can be pre-computed!

Inertia Tensor

- **General computation**
  - Define body intrinsic inertia tensor

$$I(t) = R(t)\left(\sum m_i((r_{0i}^T r_{0i})\mathbf{1} - r_{0i} r_{0i}^T)\right) R(t)^T$$

$$\Downarrow$$

$$I_{body} = \sum m_i((r_{0i}^T r_{0i})\mathbf{1} - r_{0i} r_{0i}^T)$$

$$\Downarrow$$

$$I(t) = R(t) I_{body} R(t)^T$$

Inertia Tensor

- **General computation**
  - Inverse inertia tensor

$$R(t)^T = R(t)^{-1} \qquad \left(R(t)^T\right)^T = R(t)$$

$$\Downarrow$$

$$\begin{aligned} I^{-1}(t) &= \left(R(t) I_{body} R(t)^T\right)^{-1} \\ &= \left(R(t)^T\right)^{-1} I_{body}^{-1} R(t)^{-1} \\ &= R(t) I_{body}^{-1} R(t)^T \end{aligned}$$

Rigid Body Equations of Motion

- **State variable**
  - Position and orientation
  - Linear and angular momentum

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

- **Auxiliary quantities**

$$v(t) = \frac{P(t)}{M}, \quad I(t) = R(t) I_{body} R(t)^T \quad \text{and} \quad \omega(t) = I(t)^{-1} L(t)$$

Rigid Body Equations of Motion

- **Time rate change of the state variable**

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

- **Computing order**

$$F(t) \rightarrow P(t) \qquad \tau(t) \rightarrow L(t)$$

$$P(t) \xrightarrow{P(t) = Mv(t)} v(t) \rightarrow x(t) \qquad L(t) \xrightarrow{L(t) = I(t)\omega(t)} \omega(t) \rightarrow R(t)$$

Quaternions vs. Rotation Matrices

- **Using rotation matrix is problematic**
  - Why?
    - Numerical error will accumulate on rotation matrix
    - Artificial skewing effects
    - Can be alleviated by representing rotations with unit quaternions
- **Quaternion**
  - The quaternion $s + v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$
  - Written as $[s, v]$

Quaternions vs. Rotation Matrices

- **Quaternion multiplication**

$$[s_1, v_1][s_2, v_2] = [s_1 s_2 - v_1 \cdot v_2, s_1 v_2 + s_2 v_1 + v_1 \times v_2]$$

- **From quaternion to rotation matrix**

$$\begin{pmatrix} 1-2v_y^2-2v_z^2 & 2v_xv_y-2sv_z & 2v_xv_z+2sv_y \\ 2v_xv_y+2sv_z & 1-2v_x^2-2v_z^2 & 2v_yv_z-2sv_x \\ 2v_xv_z-2sv_y & 2v_yv_z+2sv_x & 1-2v_x^2-2v_y^2 \end{pmatrix}$$

Quaternions vs. Rotation Matrices

- **Rotation of a rigid body**
  - Suppose the body to rotate with constant $\omega(t)$ for a period of time Δt :

$$[\cos\frac{|\omega(t)|\Delta t}{2}, \sin\frac{|\omega(t)|\Delta t}{2}\frac{\omega(t)}{|\omega(t)|}]$$

  - Derivation for $\dot{q}(t)$
    - Approximation at $t + t_0$:

$$q(t_0+\Delta t) = [\cos\frac{|\omega(t_0)|\Delta t}{2}, \sin\frac{|\omega(t_0)|\Delta t}{2}\frac{\omega(t_0)}{|\omega(t_0)|}]\, q(t_0)$$

Quaternions vs. Rotation Matrices

- **Rotation of a rigid body**
  - Derivation for $\dot{q}(t)$
    - Substitute $t = t_0 + \Delta t$
    - We have

$$q(t) = [\cos\frac{|\omega(t_0)|(t-t_0)}{2}, \sin\frac{|\omega(t_0)|(t-t_0)}{2}\frac{\omega(t_0)}{|\omega(t_0)|}]q(t_0)$$

    - Let us differentiate $\dot{q}(t)$

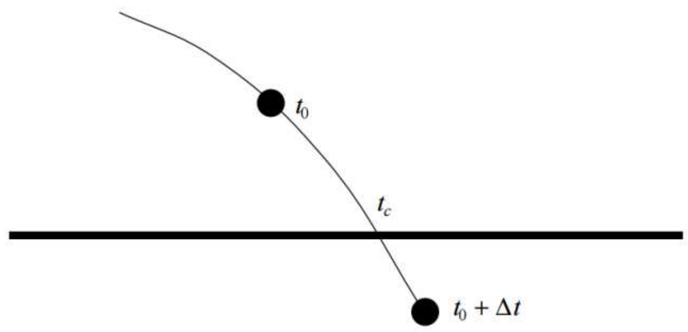
$$\begin{aligned}\dot{q}(t) &= \frac{d}{dt}\left([\cos\frac{|\omega(t_0)|(t-t_0)}{2}, \sin\frac{|\omega(t_0)|(t-t_0)}{2}\frac{\omega(t_0)}{|\omega(t_0)|}]\,q(t_0)\right)\\ &= \frac{d}{dt}\left([\cos\frac{|\omega(t_0)|(t-t_0)}{2}, \sin\frac{|\omega(t_0)|(t-t_0)}{2}\frac{\omega(t_0)}{|\omega(t_0)|}]\right)q(t_0)\\ &= [0, \frac{|\omega(t_0)|}{2}\frac{\omega(t_0)}{|\omega(t_0)|}]\,q(t_0)\\ &= [0, \tfrac{1}{2}\omega(t_0)]\,q(t_0) = \tfrac{1}{2}[0, \omega(t_0)]\,q(t_0).\end{aligned}$$

$$\Rightarrow \quad \dot{q}(t) = \tfrac{1}{2}\omega(t)q(t)$$

III. Constrained Rigid Body Dynamics

Problems of Non-penetration Constraints

- **Two types of contacts**
  - Colliding contact
    - Two bodies are in contact at some point **p**
    - They have a velocity towards each other
    - Y(t) has discontinuity
      - E.g., instantaneous change of velocity
    - How to solve?
      - Stop ODE solver at the contact
      - Compute how Y(t) changes
      - Restart ODE solver



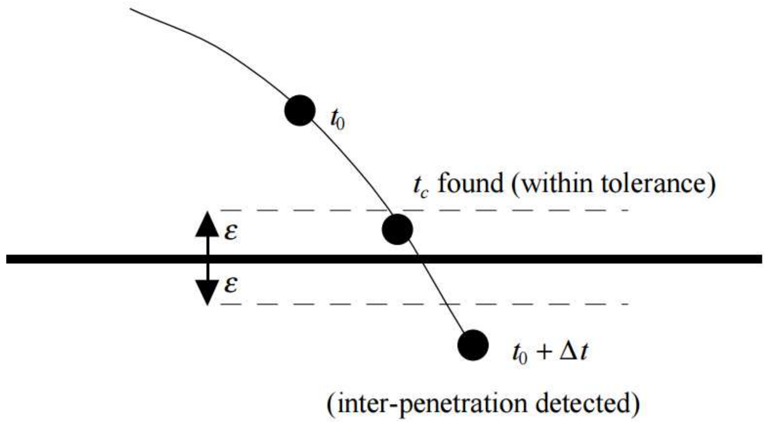
(inter-penetration detected)

Problems of Nonpenetration Constraints

- **Two types of contacts**
  - Resting contact
    - Whenever bodies are resting on one another at some point p
    - We compute a force that prevents the particle from accelerating
    - Contact force
      - A force that acts at the point of contact between two objects
- **Two problems to solve**
  - Compute velocity changes for colliding contact
  - Compute the contact forces that prevent inter-penetration

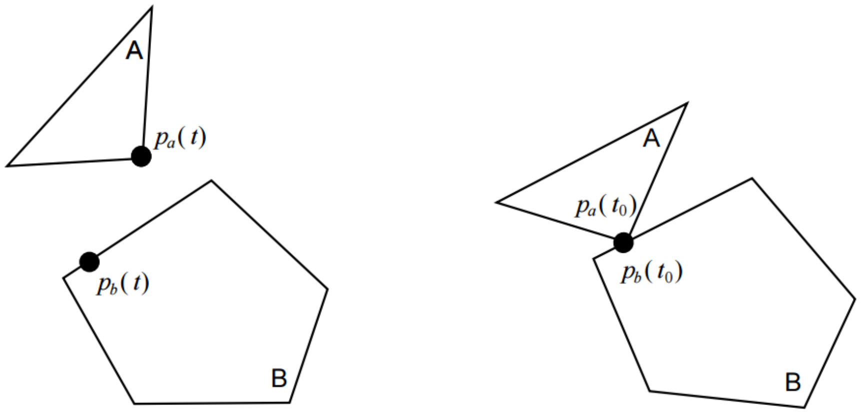
Bisection

- **When inter-penetration is detected**
  - We inform the ODE solver that we wish to restart back at time *t*
  - Simulate forward to time $t_0 + \Delta t/2$, and repeat until some tolerance is met



Colliding Contact

- **Description of a colliding contact**
  - Illustration



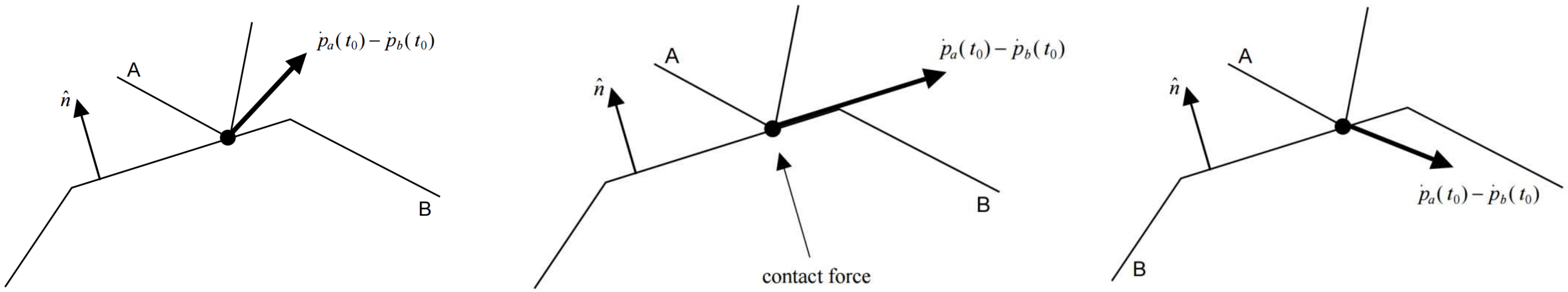
  - Formula

$$\dot{p}_a(t_0) = v_a(t_0) + \omega_a(t_0) \times (p_a(t_0) - x_a(t_0))$$

$$\dot{p}_b(t_0) = v_b(t_0) + \omega_b(t_0) \times (p_b(t_0) - x_b(t_0))$$

Colliding Contact

- **Examine the relative velocity**



$$v_{rel} = \hat{n}(t_0) \cdot (\dot{p}_a(t_0) - \dot{p}_b(t_0))$$

- v_{rel} >0: two bodies leaving apart, not interested
- v_{rel} =0: resting contact
- v_{rel} <0: a colliding contact
  - How do we compute the change in velocity?

Colliding Contact

- **Definition of an impulse**
  - Force exerted over a time period $F\Delta t = J$
  - Apply an impulse *J* to a rigid body with mass M $\Delta P = J$ $\Delta v = \frac{J}{M}$
  - Impulsive torque $\tau_{impulse} = (p - x(t)) \times J$
  - Change in angular momentum $\Delta L = \tau_{impulse}$
  - Change in angular velocity $I^{-1}(t_0)\tau_{impulse}$

Colliding Contact

- **How to compute the impulse?**
  - Force F is unknown
  - For frictionless bodies, the direction of the impulse will be in the normal direction

$$J = j\hat{n}(t_0)$$

  - How to compute *j*?
    - We compute *j* by using an empirical law for collisions
  - Some definitions

$\dot{p}_a^-(t_0)$ ⟵ velocity of the contact vertex of A prior to the impulse being applied

$\dot{p}_a^+(t_0)$ ⟵ velocity after we apply the impulse *J*

Colliding Contact

- **Definition of relative velocities**
  - Initial relative velocity in the normal direction

$$v_{rel}^{-} = \hat{n}(t_0) \cdot (\dot{p}_a^{-}(t_0) - \dot{p}_b^{-}(t_0))$$

  - After the application of the impulse

$$v_{rel}^{+} = \hat{n}(t_0) \cdot (\dot{p}_a^{+}(t_0) - \dot{p}_b^{+}(t_0))$$

  - Empirical law for frictionless collisions

$$v_{rel}^{+} = -\epsilon v_{rel}^{-} \qquad 0 \leq \epsilon \leq 1$$

Coefficient of restitution

Colliding Contact

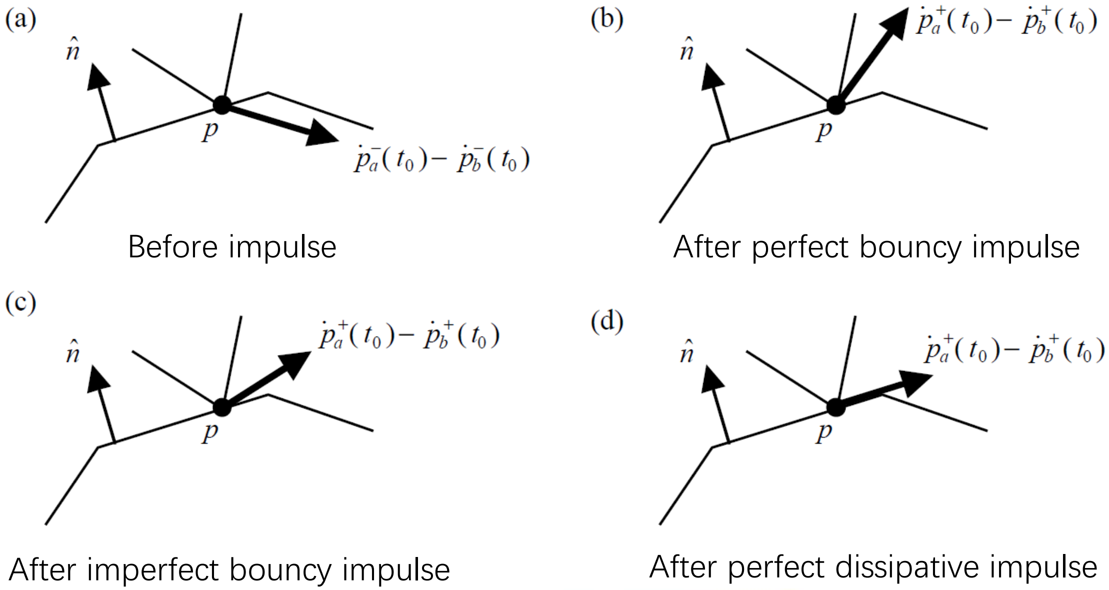
- **Physical meaning for coefficient of restitution**
  - Perfect bouncing
    - No kinetic energy is lost

$$\epsilon = 1 \qquad\qquad v_{rel}^{+} = -v_{rel}^{-}$$

  - Perfect dissipative
    - A maximum of kinetic energy is lost $\epsilon = 0$
      - After this collision, the two bodies will be in rest contact $v_{rel}^{+} = 0$

Colliding Contact

- **Physical meaning for coefficient of restitution**
  - Illustration



Colliding Contact

- **Derivation**

$$\dot{p}_a^+(t_0) = v_a^+(t_0) + \omega_a^+(t_0) \times r_a \qquad v_a^+(t_0) = v_a^-(t_0) + \frac{j\hat{n}(t_0)}{M_a} \qquad \omega_a^+(t_0) = \omega_a^-(t_0) + I_a^{-1}(t_0)\left(r_a \times j\hat{n}(t_0)\right)$$

$$\downarrow$$

$$\begin{aligned}
\dot{p}_a^+(t_0) &= \left(v_a^-(t_0) + \frac{j\hat{n}(t_0)}{M_a}\right) + \left(\omega_a^-(t_0) + I_a^{-1}(t_0)\left(r_a \times j\hat{n}(t_0)\right)\right) \times r_a \\
&= v_a^-(t_0) + \omega_a^-(t_0) \times r_a + \left(\frac{j\hat{n}(t_0)}{M_a}\right) + \left(I_a^{-1}(t_0)\left(r_a \times j\hat{n}(t_0)\right)\right) \times r_a \\
&= \dot{p}_a^- + j\left(\frac{\hat{n}(t_0)}{M_a} + I_a^{-1}(t_0)\left(r_a \times \hat{n}(t_0)\right)\right) \times r_a
\end{aligned}$$

$$\downarrow$$

$$\dot{p}_b^+(t_0) = \dot{p}_b^- - j\left(\frac{\hat{n}(t_0)}{M_b} + I_b^{-1}(t_0)\left(r_b \times \hat{n}(t_0)\right)\right) \times r_b$$

Colliding Contact

- **This yields**

$$\dot{p}_a^+(t_0) - \dot{p}_b^+ = (\dot{p}_a^-(t_0) - \dot{p}_b^-) + j\Bigg(\frac{\hat{n}(t_0)}{M_a} + \frac{\hat{n}(t_0)}{M_b} +$$
$$\big(I_a^{-1}(t_0)\big(r_a \times \hat{n}(t_0)\big)\big) \times r_a + \big(I_b^{-1}(t_0)\big(r_b \times \hat{n}(t_0)\big)\big) \times r_b\Bigg)$$

$$\Downarrow$$

$$\begin{aligned} v_{rel}^+ &= \hat{n}(t_0) \cdot (\dot{p}_a^+(t_0) - \dot{p}_b^+) \\ &= \hat{n}(t_0) \cdot (\dot{p}_a^-(t_0) - \dot{p}_b^-) + j\Bigg(\frac{1}{M_a} + \frac{1}{M_b} + \\ &\quad \hat{n}(t_0) \cdot \big(I_a^{-1}(t_0)\big(r_a \times \hat{n}(t_0)\big)\big) \times r_a + \hat{n}(t_0) \cdot \big(I_b^{-1}(t_0)\big(r_b \times \hat{n}(t_0)\big)\big) \times r_b\Bigg) \\ &= v_{rel}^- + j\Bigg(\frac{1}{M_a} + \frac{1}{M_b} + \\ &\quad \hat{n}(t_0) \cdot \big(I_a^{-1}(t_0)\big(r_a \times \hat{n}(t_0)\big)\big) \times r_a + \hat{n}(t_0) \cdot \big(I_b^{-1}(t_0)\big(r_b \times \hat{n}(t_0)\big)\big) \times r_b\Bigg) \end{aligned}$$

Colliding Contact

- **This yields**
  - Empirical law for frictionless collision

$$v_{rel}^{+} = -\epsilon v_{rel}^{-} \qquad 0 \leq \epsilon \leq 1$$

$$\Downarrow$$

$$v_{rel}^{-} + j\left(\frac{1}{M_a} + \frac{1}{M_b} + \hat{n}(t_0) \cdot \left(I_a^{-1}(t_0)\left(r_a \times \hat{n}(t_0)\right)\right) \times r_a + \hat{n}(t_0) \cdot \left(I_b^{-1}(t_0)\left(r_b \times \hat{n}(t_0)\right)\right) \times r_b\right) = -\epsilon v_{rel}^{-}$$

$$\Downarrow$$

$$j = \frac{-(1+\epsilon)v_{rel}^{-}}{\frac{1}{M_a} + \frac{1}{M_b} + \hat{n}(t_0) \cdot \left(I_a^{-1}(t_0)\left(r_a \times \hat{n}(t_0)\right)\right) \times r_a + \hat{n}(t_0) \cdot \left(I_b^{-1}(t_0)\left(r_b \times \hat{n}(t_0)\right)\right) \times r_b}$$

Colliding Contact

- **Handling fixed bodies**
  - Some bodies cannot be moved
    - Floors, walls, etc.
  - Look at the formulation again

$$j = \frac{-(1+\epsilon)v_{rel}^{-}}{\frac{1}{M_a} + \frac{1}{M_b} + \hat{n}(t_0) \cdot \left(I_a^{-1}(t_0)\left(r_a \times \hat{n}(t_0)\right)\right) \times r_a + \hat{n}(t_0) \cdot \left(I_b^{-1}(t_0)\left(r_b \times \hat{n}(t_0)\right)\right) \times r_b}$$

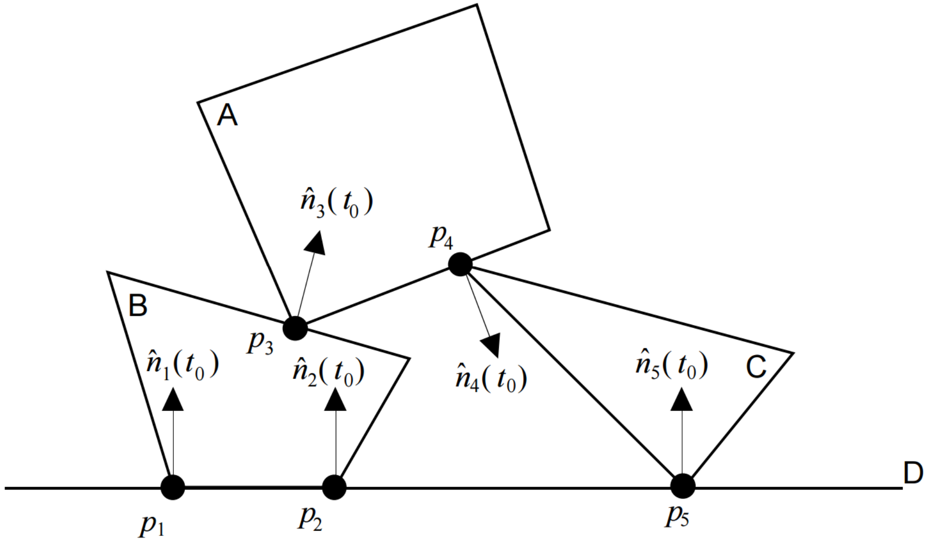
  - What we need
    - Inverse of mass and inertia tensor
  - Tricks
    - Set inverse mass to be zero
    - Set inverse inertia tensor to be zero matrix

Resting Contact

- **Condition of resting contact**
  - Relative velocity v_{rel} is zero (within numerical threshold)
  - Contact force
    - At each contact point, there is a contact force $f_i \hat{n}_i(t_0)$ where f_i is an unknown scalar
    - Our goal
      - Determine each f_i at the same time
      - To maintain contact between bodies

Resting Contact

- **Condition of resting contact**
  - Computing contact forces



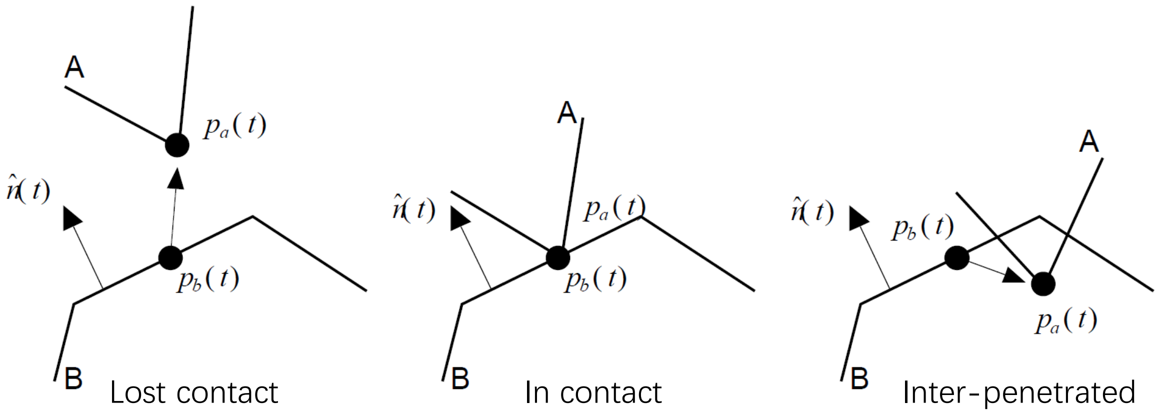
Resting Contact

- **Derivation**
  - Contact force subject to three conditions
    - 1. Must prevent inter-penetration
    - 2. Must be repulsive
      - Never act like a “glue” and hold bodies together
    - 3. Be zero if the bodies begin to separate

Resting Contact

- **Derivation**
  - Preventing inter-penetration
    - Construction of separation distance

$$d_i(t) = \hat{n}_i(t) \cdot (p_a(t) - p_b(t))$$



Resting Contact

- **Derivation**
  - Preventing inter-penetration
    - Consider $d_i(t_0) = 0$
    - We have to keep the two bodies from accelerating towards each other
      - Taking the second derivative of $d_i(t_0)$

$$\ddot{d}(t_0) = \hat{n}_i(t_0) \cdot (\ddot{p}_a(t_0) - \ddot{p}_b(t_0)) + 2\dot{\hat{n}}_i(t_0) \cdot (\dot{p}_a(t_0) - \dot{p}_b(t_0))$$

      - $\ddot{d}_i(t_0) > 0$: contact will break immediately
      - $\ddot{d}_i(t_0) = 0$: contact remains
      - $\ddot{d}_i(t_0) < 0$: must be avoided

Resting Contact

- **Expression for three conditions**
  - Non-interpenetration

$$\ddot{d}_i(t_0) \geq 0$$

  - Repulsiveness

$$f_i \geq 0$$

  - Contact breaking

$$f_i \ddot{d}_i(t_0) = 0$$

Resting Contact

- **Computing contact force**
  - Force contribution
    - Consider the expression

$$\ddot{d}(t_0) = \hat{n}_i(t_0) \cdot (\ddot{p}_a(t_0) - \ddot{p}_b(t_0)) + 2\dot{\hat{n}}_i(t_0) \cdot (\dot{p}_a(t_0) - \dot{p}_b(t_0))$$

$$\ddot{p}_a(t) = \dot{v}_a(t) + \dot{\omega}_a(t) \times r_a(t) + \omega_a(t) \times (\omega_a(t) \times r_a(t))$$

    - Forces contribute to
      - Linear acceleration $\dot{v}_a(t)$ $\quad \frac{f_j \hat{n}_j(t_0)}{m_a} = f_j \frac{\hat{n}_j(t_0)}{m_a}$
      - Angular acceleration $\dot{\omega}_a(t) = I_a^{-1}(t)\tau_a(t) + I_a^{-1}(t)(L_a(t) \times \omega_a(t))$ $\quad (p_j - x_a(t_0)) \times f_j \hat{n}_j(t_0)$

Resting Contact

- **Computing contact force**
  - Express separating distance acceleration in terms of all associated forces

$$\ddot{d}_i(t_0) = a_{i1} f_1 + a_{i2} f_2 + \cdots + a_{in} f_n + b_i$$

  - Write for all contact points

$$\begin{pmatrix} \ddot{d}_1(t_0) \\ \vdots \\ \ddot{d}_n(t_0) \end{pmatrix} = \mathbf{A} \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} + \begin{pmatrix} b_i \\ \vdots \\ b_n \end{pmatrix}$$

Resting Contact

- **Computing contact force**
  - Contribution of a_{ij}
    - Consider linear and angular acceleration

$$\frac{\hat{n}_j(t_0)}{m_a} + \left(I_a^{-1}(t_0)\left((p_j - x_a(t_0)) \times \hat{n}_j(t_0)\right)\right) \times r_a$$

  - Contribution of b_i
    - Collect the force independent part

$$\frac{F_a(t_0)}{m_a} + \left(I_a^{-1}(t_0)\tau_a(t_0)\right) \times r_a + \omega_a(t_0) \times (\omega_a(t_0) \times r_a) + \left(I_a^{-1}(t_0)(\text{Ł}_a(t_0) \times \omega_a(t_0))\right) \times r_a$$

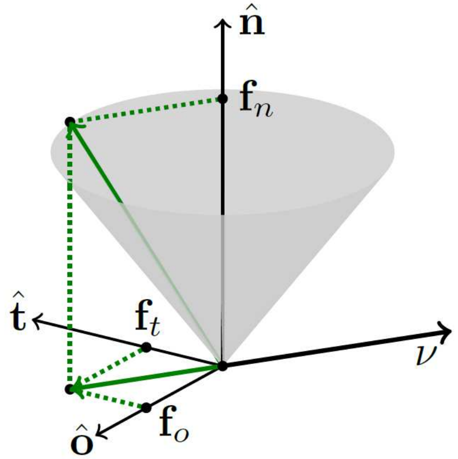
Frictional Force at Contact

- **Friction forces**
  - Dissipative
  - In contact interfaces to halt sliding at sliding contacts
  - Prevent sliding at sticking and rolling contacts

- **Dry (static) friction**
  - Assume to act at contacts between body surfaces
  - Allow bodies to stick together
  - Require a non-zero tangential force to initiate sliding

Frictional Force at Contact

- **Isotropic Coulomb friction model**
  - Assumption
    - Contact occurs at a single point with a uniquely defined tangent plane
    - A friction cone



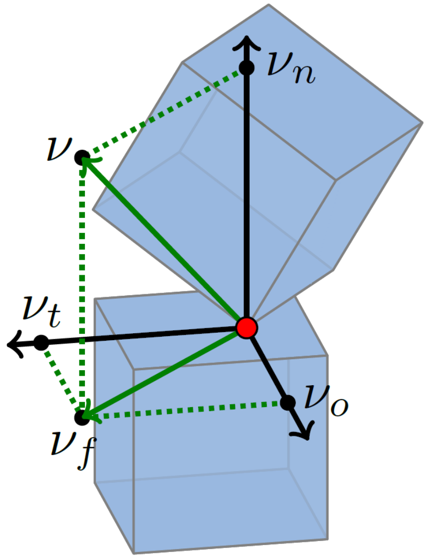
normal force is unilateral

$$\mathbf{f}_n \geq 0$$

Tangential force is determined by normal force

Frictional Force at Contact

- **Isotropic Coulomb friction model**
  - The relative velocity between the touching points



The contact is sliding

$$\nu_n = 0$$

The contact is separating

$$\nu_n > 0$$

Frictional Force at Contact

- **Isotropic Coulomb friction model**
  - Two conditions
    - The net contact force must lie in a quadratic friction cone
    - When the bodies are slipping, the friction force must be the one on the boundary of the cone
      - Directly oppose the sliding motion
      - The cone is defined as

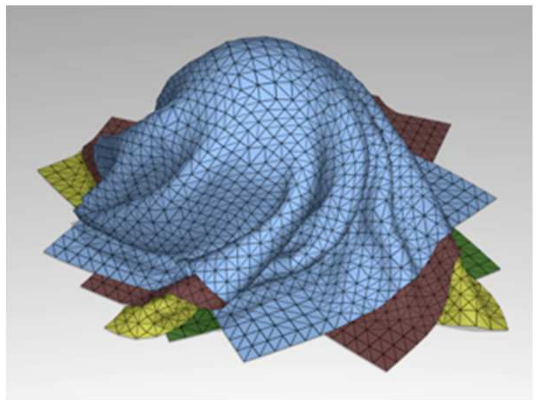
$$\mathcal{F}(\mathbf{f}_n, \mu) = \{\mu^2 \mathbf{f}_n^2 - \mathbf{f}_t^2 - \mathbf{f}_o^2 \geq 0, \mathbf{f}_n \geq 0\}$$

  - Sliding friction force
    - Maximize friction dissipation $\mathbf{f}_t = -\mu \mathbf{f}_n \frac{\nu_t}{\beta}, \quad \mathbf{f}_o = -\mu \mathbf{f}_n \frac{\nu_o}{\beta}, \quad \beta = \sqrt{\nu_t^2 + \nu_o^2}$

IV. Collision Detection

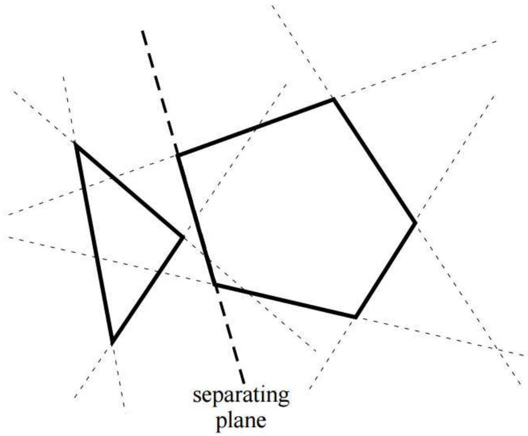
Collision Detection

- **Problem formulation**
  - The computational problem of detecting the intersection of two or more objects



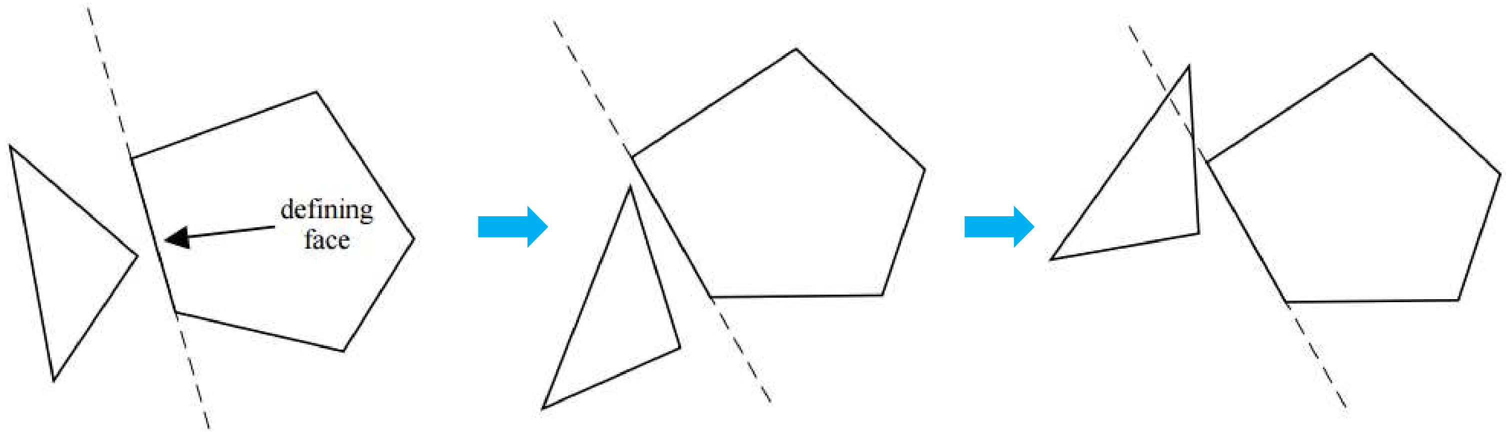
Collision Detection

- **How to detect inter-penetration?**
  - Convex polyhedron
    - Two polyhedra do no inter-penetrate if and only if a separating plane between them exists
    - Finding the separating plane



Collision Detection

- **How to detect inter-penetration?**
  - Convex polyhedra
    - Progress with defining face



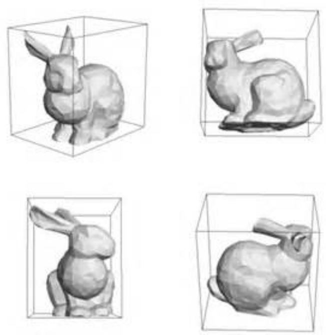
Bounding volumes

- **Why bounding volume**
  - Directly collision testing of two objects is often very expensive
  - Especially when objects consist of hundreds or even thousands of polygons

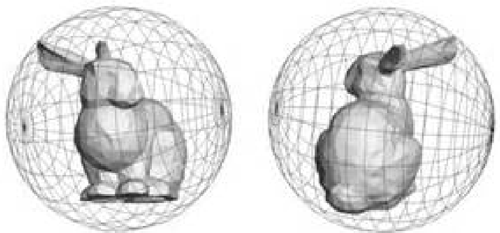
- **What is a bounding volume**
  - A bounding volume (BV) is a single simple volume encapsulating one or more objects of more complex nature

Bounding volumes

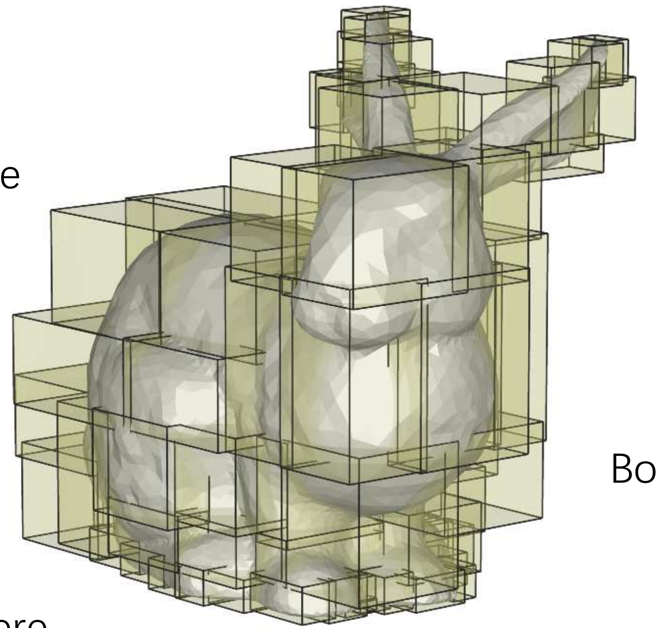
- **Example of Bounding Volume**



Bounding Cube



Bounding sphere



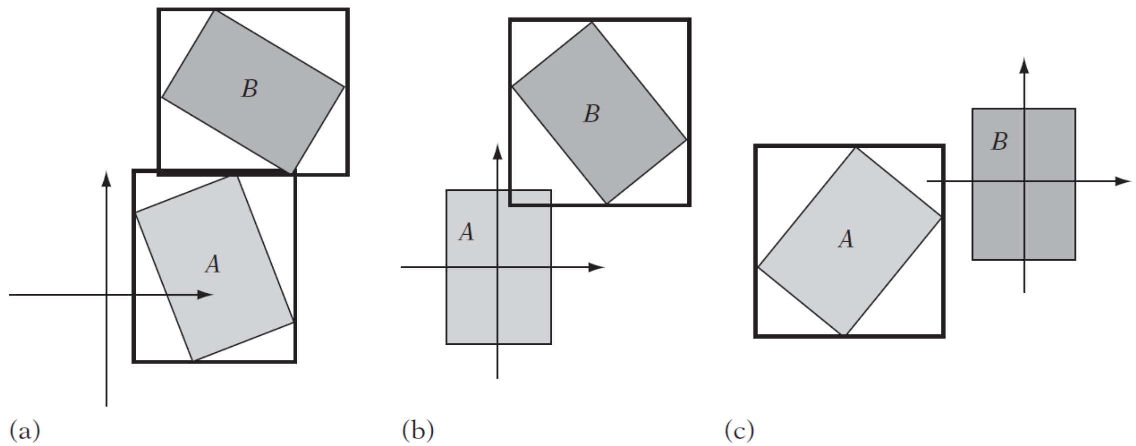
Bounding poly-cube

Bounding Volumes

- **Desirable BV characteristics**
  - Not all geometric objects serve as effective bounding volumes
  - Desirable properties
    - Inexpensive intersection tests
    - Tight fitting
    - Inexpensive to compute
    - Easy to rotate and transform
    - Use little memory

Bounding volumes

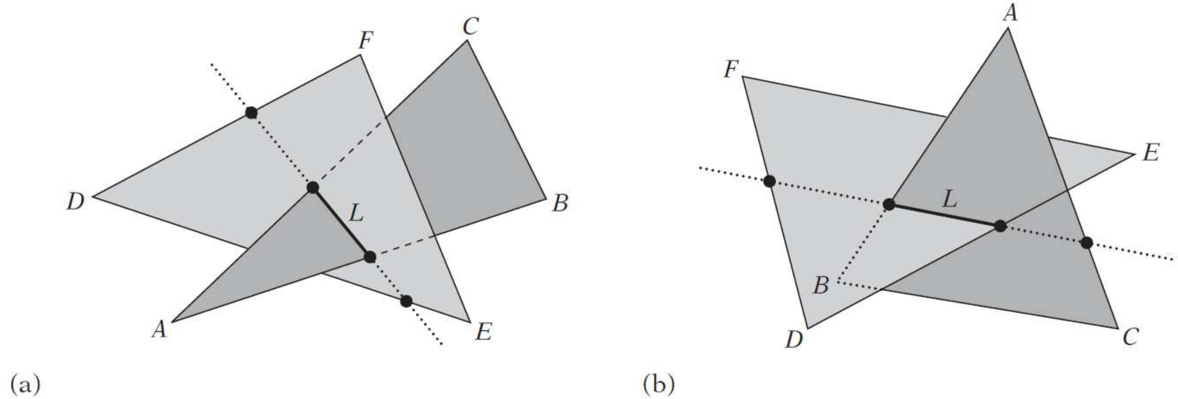
- **Axis-aligned bounding boxes (AABBs)**
  - AABBs in terms of different coordinate system



(a) AABBs in world space (b) AABBs in the local space of A (c) AABBs in the local space of B

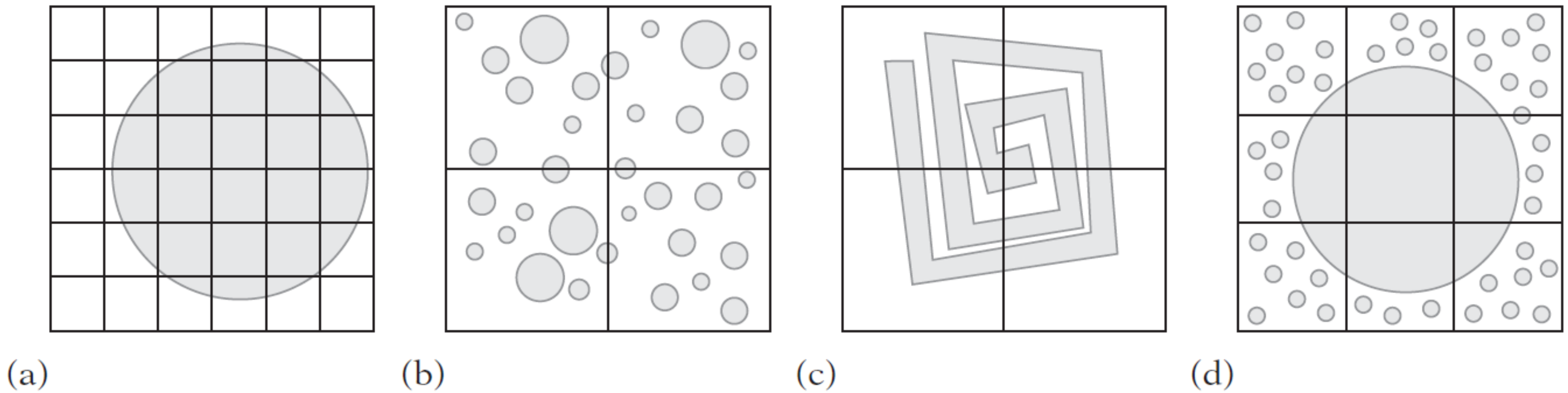
Basic Primitive Tests

- **Testing primitives**
  - Testing triangle against triangle
    - Detecting the intersection of two triangles ABC and DEF
    - When two triangles intersect
      - Two edges of one triangle pierce the interior of the other
      - One edge from each triangle pierces the interior of the other



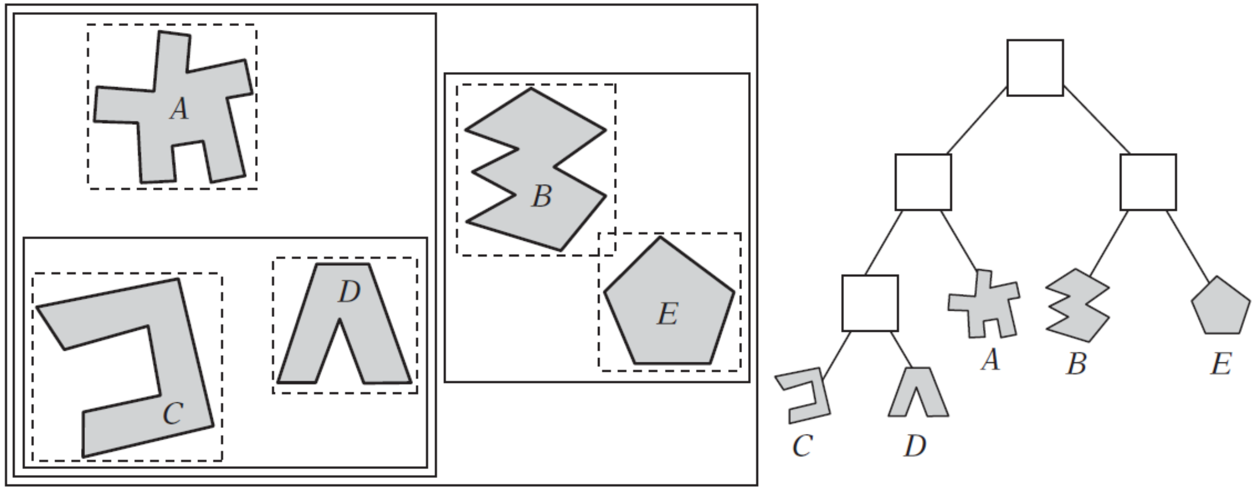
Spatial Partitioning

- **Uniform grids**
  - In-depth tests are only performed against those found sharing cells



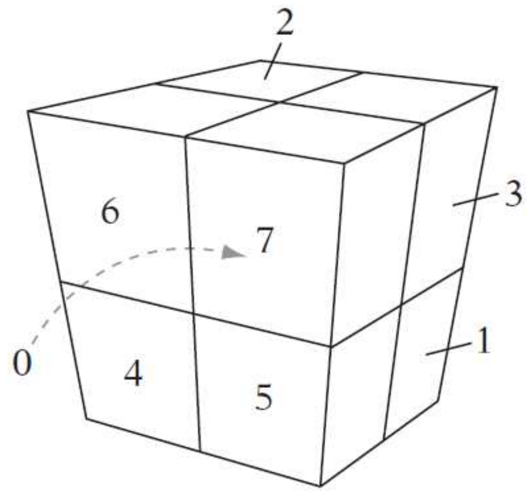
Spatial Partitioning

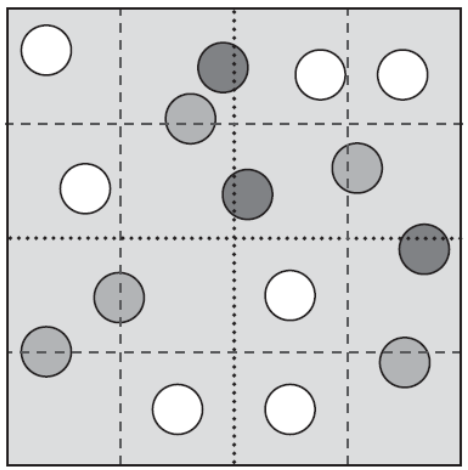
- **Bounding volume hierarchy (BVH)**
  - Time complexity can be reduced to logarithmic in the number of tests performed



Spatial Partitioning

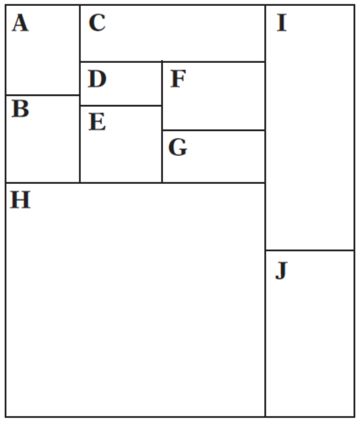
- **Octree (quadtree for 2D)**
  - An axis-aligned hierarchical partitioning of a volume of 3D world space



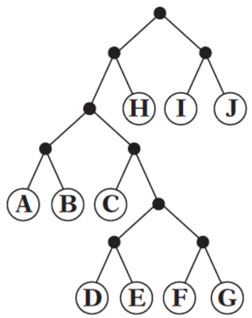


Spatial Partitioning

- **K-d trees**
  - A generalization of octrees and quadtrees
    - The k-d tree divides space along one dimension at a time



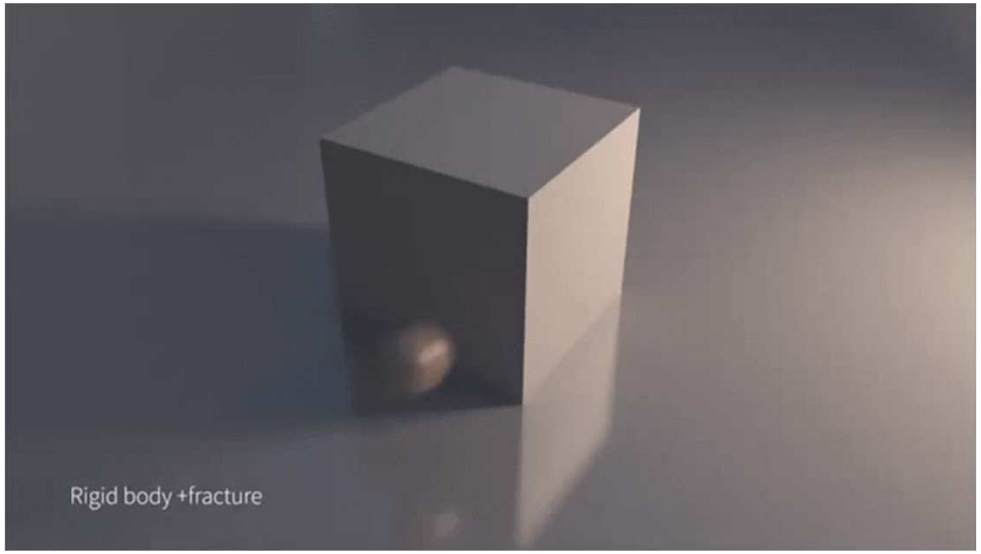
(a)



(b)

What you will get finally?

- **An example of a system of rigid body motion**



Next Lecture: Rigid Body Simulation II