Computer Graphics I

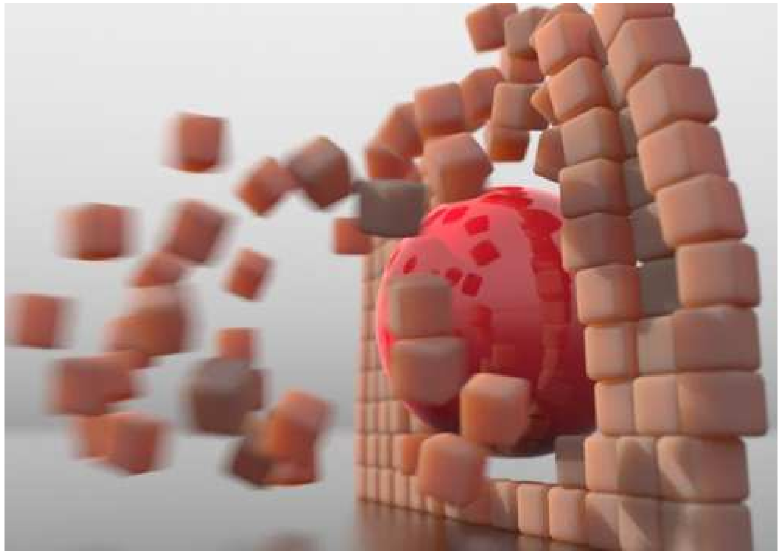
Lecture 18: Computer animation 2

Xiaopei LIU

School of Information Science and Technology
ShanghaiTech University

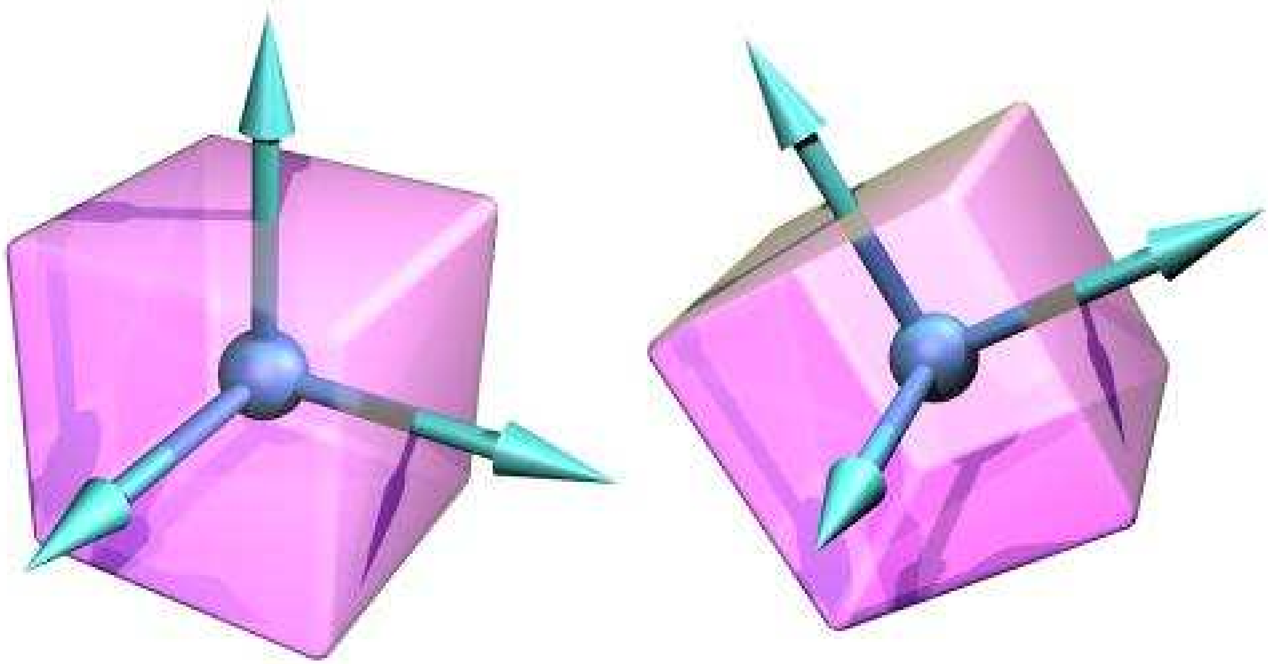
Rigid body

- **What is a rigid body?**
  - The body never deforms nonlinearly (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



1. Rigid body

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

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

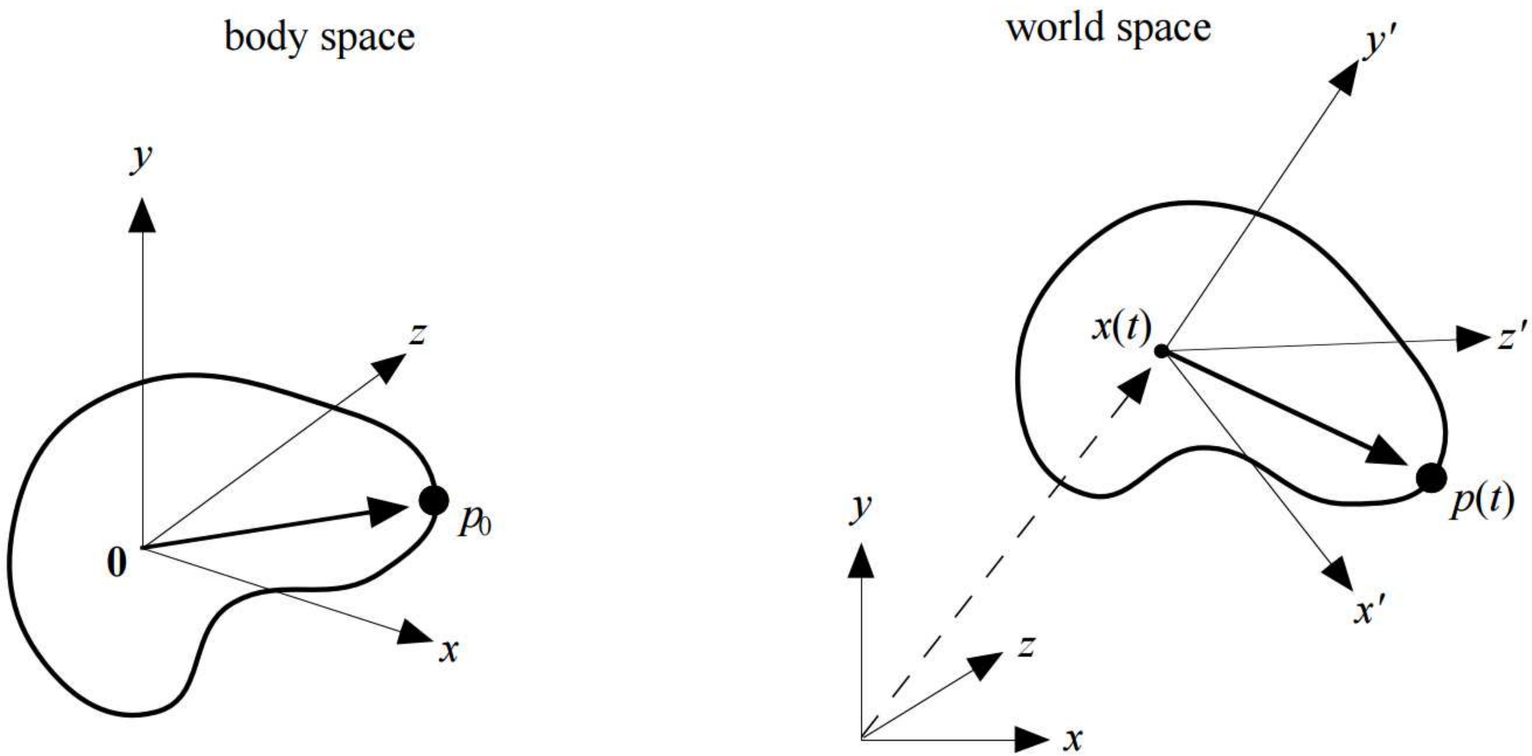
Particle system

- **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

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

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

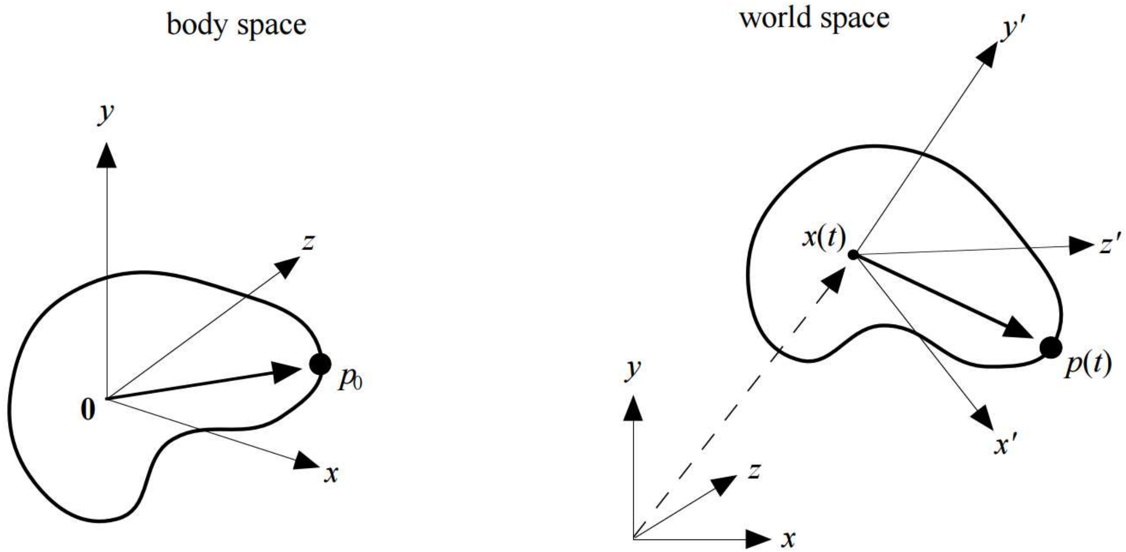


Position and orientation

- **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

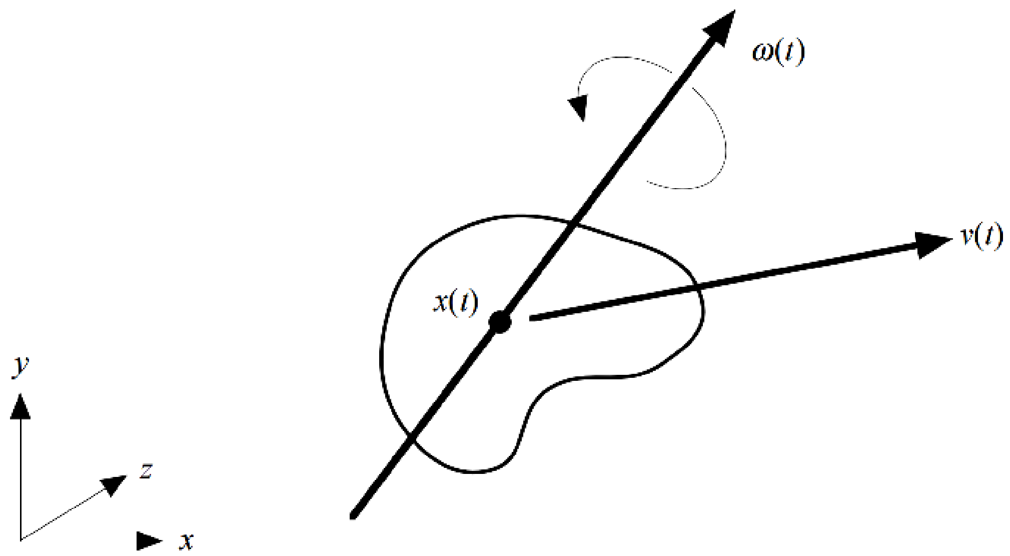


Linear & angular velocity

- **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)

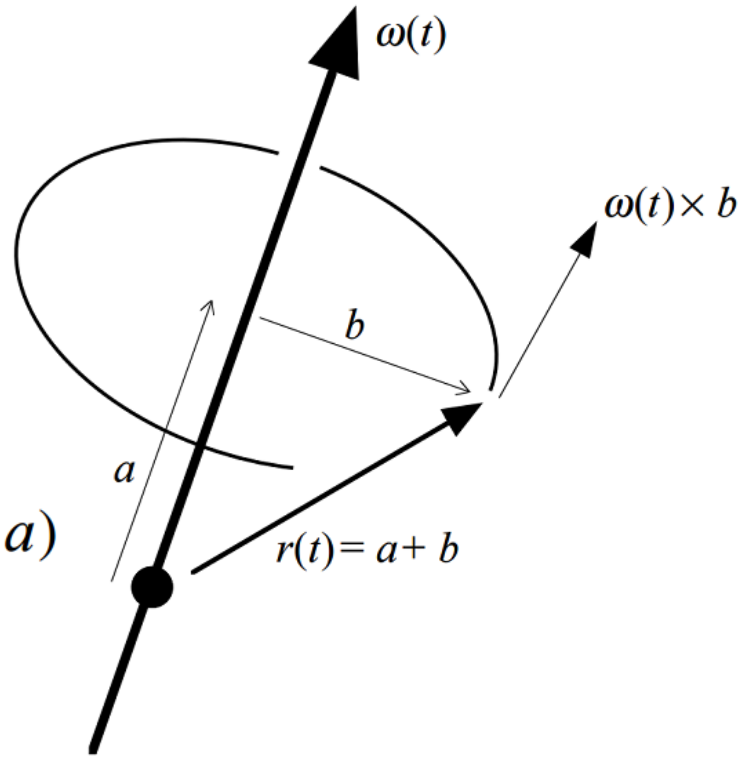
$$r(t) = a + b$$

  - Instant velocity

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

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

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

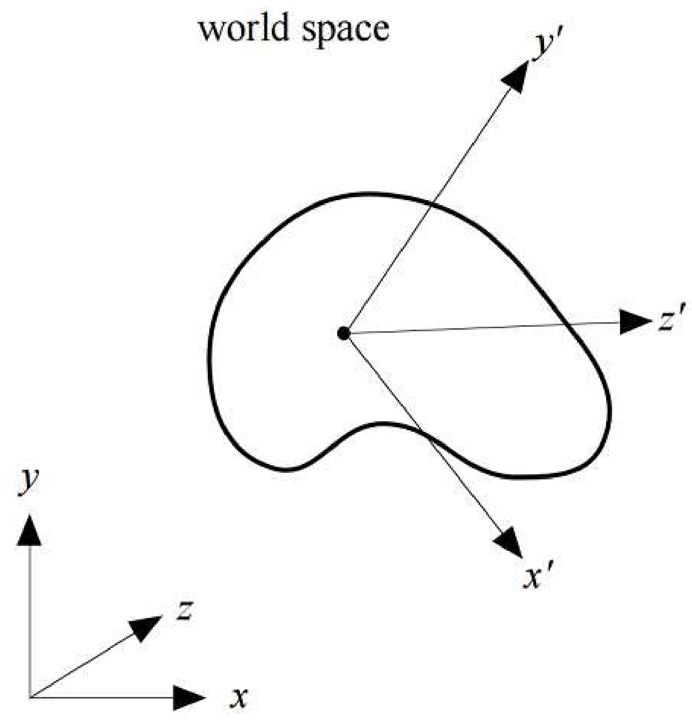


Calculation of rotation

- **Rotating a body coordinate frame**

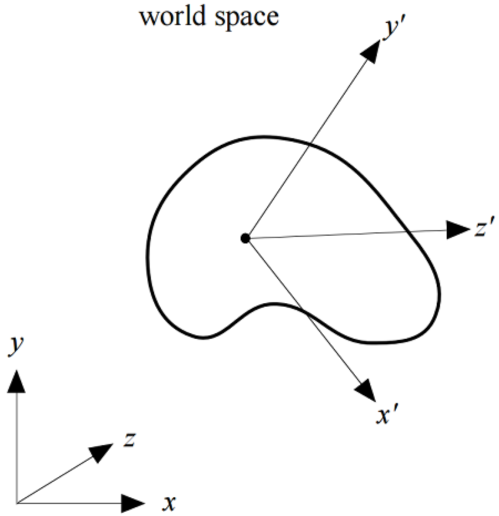
$$R(t) = \begin{pmatrix} r_{xx} & r_{yx} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{pmatrix} \Rightarrow R(t)\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{pmatrix}$$

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



Calculation of rotation

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



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

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

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

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

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

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

Center of mass

- **The center of mass in 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

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

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

Velocity of a particle

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

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

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

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

differentiation

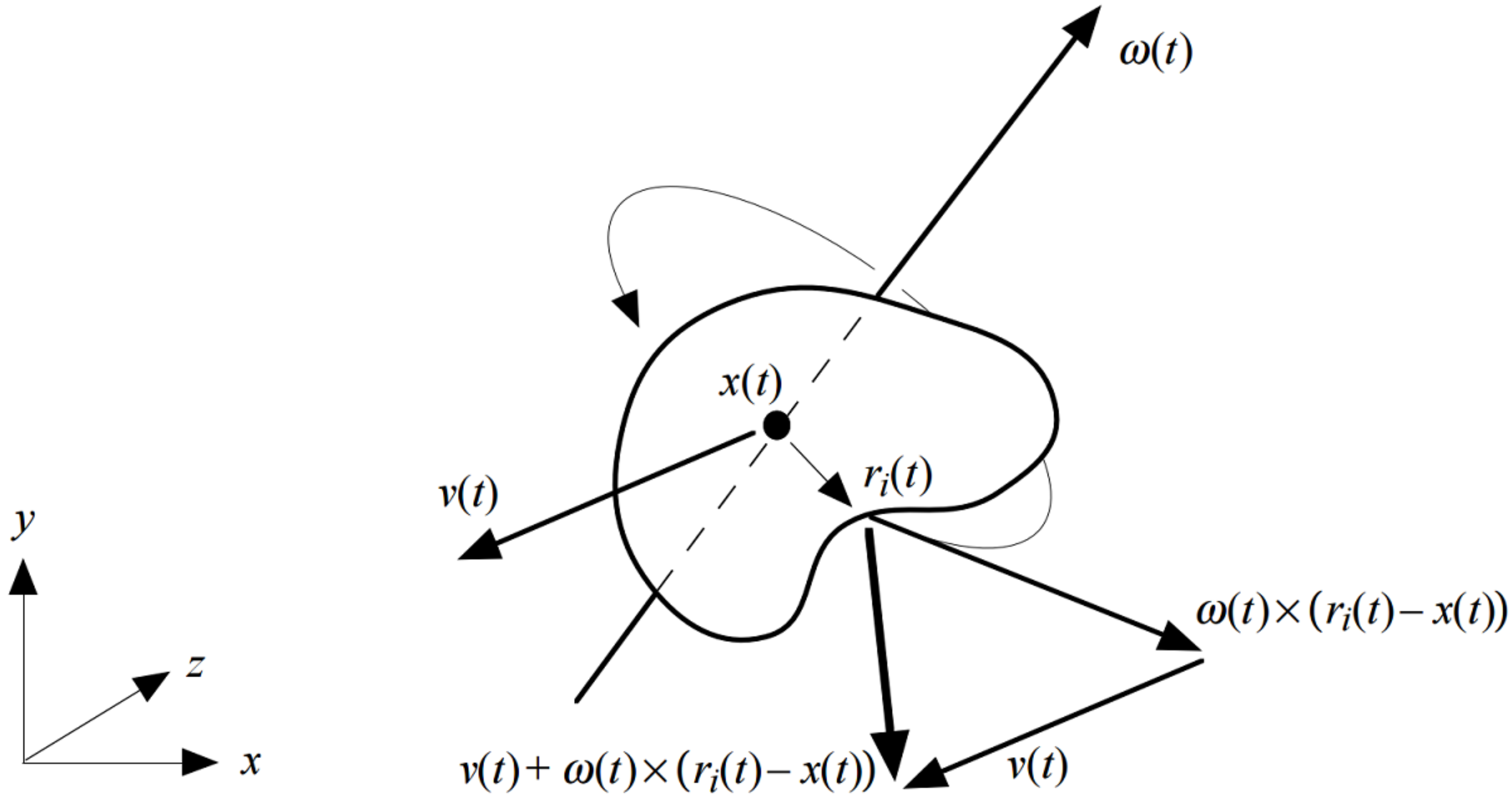
$$\dot{r}_i(t) = \omega^* R(t)r_{0i} + v(t)$$

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

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

Velocity of a particle

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



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

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

$$\begin{aligned} P(t) &= \sum m_i \dot{r}_i(t) \\ &= \sum \Big(m_i v(t) + m_i \omega(t) \times (r_i(t) - x(t)) \Big) \qquad \sum m_i (r_i(t) - x(t)) = \mathbf{0} \\ &= \sum m_i v(t) + \omega(t) \times \sum m_i \left(r_i(t) - x(t) \right) \end{aligned}$$

$$\Downarrow$$

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

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 write simpler equations
- **Analogous to linear momentum**
  - Linear momentum

$$P(t) = Mv(t)$$

  - Angular momentum

Inertia tensor: 3x3 matrix

$$L(t) = I(t)\omega(t)$$

  - 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**
  - Determines the torque needed for a desired angular acceleration
  - Depends 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)

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

$$r_i' = r_i(t) - x(t)$$

Shall we re-compute when rotated?

  - Continuous distribution: sum to integral, mass to density

Inertia tensor

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

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

$+$

$$r'^T_i r'_i = r'^2_{ix} + r'^2_{iy} + r'^2_{iz}$$

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

$+$

$$r'_i r'^T_i = \begin{pmatrix} r'^2_{ix} & r'_{ix} r'_{iy} & r'_{ix} r'_{iz} \\ r'_{iy} r'_{ix} & r'^2_{iy} & r'_{ix} r'_{iz} \\ r'_{iz} r'_{ix} & r'_{iz} r'_{iy} & r'^2_{iz} \end{pmatrix}$$

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

Inertia tensor

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

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

+

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

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

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

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

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

$$\Downarrow$$

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

2. Rigid body dynamics

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**

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

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

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**

$$\begin{pmatrix} 1-2v_y^2-2v_z^2 & 2v_x v_y - 2s v_z & 2v_x v_z + 2s v_y \\ 2v_x v_y + 2s v_z & 1-2v_x^2-2v_z^2 & 2v_y v_z - 2s v_x \\ 2v_x v_z - 2s v_y & 2v_y v_z + 2s v_x & 1-2v_x^2-2v_y^2 \end{pmatrix}$$

Quaternions vs. rotation matrices

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

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

  - Equation for $\dot{q}(t)$

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

Solving ordinary differential equations

- **The rigid body dynamics results in**
  - 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

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

    - Integrating the differential equation

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

$$\Downarrow$$

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

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

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

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))

Now pick a step-size $h > 0$ and define

$$
\begin{aligned}
y_{n+1} &= y_n + \tfrac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \\
t_{n+1} &= t_n + h
\end{aligned}
$$

for $n = 0, 1, 2, 3, \ldots$, using

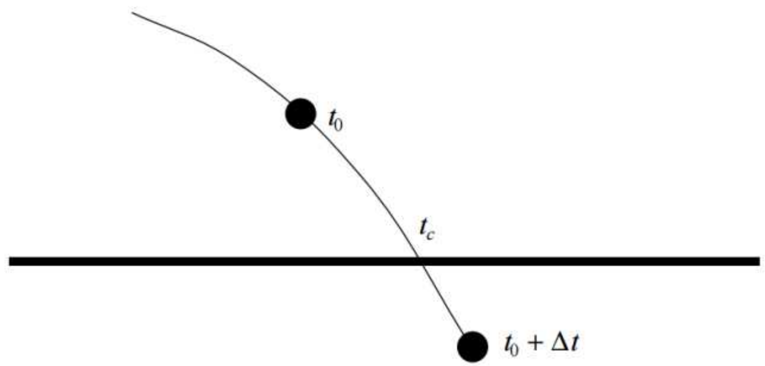
$$
\begin{aligned}
k_1 &= h\, f(t_n, y_n), \\
k_2 &= h\, f\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right), \\
k_3 &= h\, f\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right), \\
k_4 &= h\, f\left(t_n + h, y_n + k_3\right).
\end{aligned}
$$

3. Non-penetration constraints

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



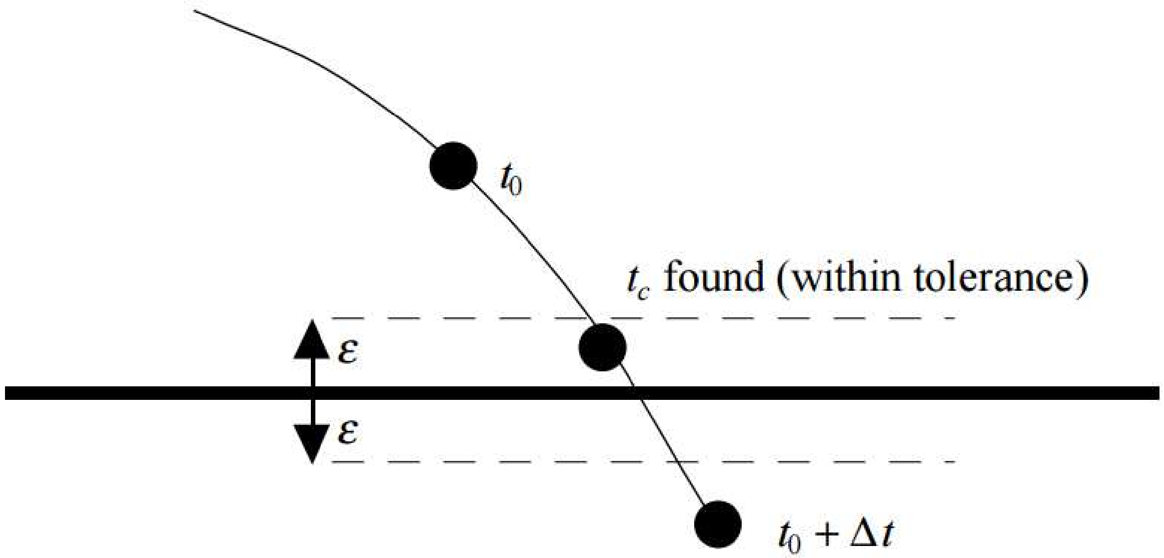
(inter-penetration detected)

Problems of non-penetration 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

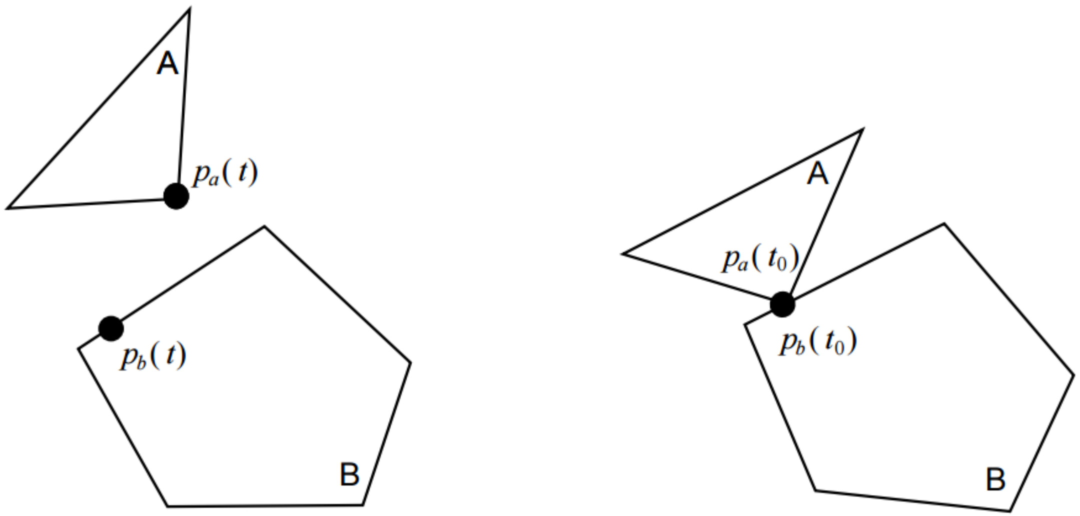
- **When inter-penetration is detected**
  - We inform the ODE solver that we wish to restart back at time t
  - Simulate forward to time $t_0 + \Delta t/2$, and repeat until some tolerance is met



(inter-penetration detected)

Colliding contact

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



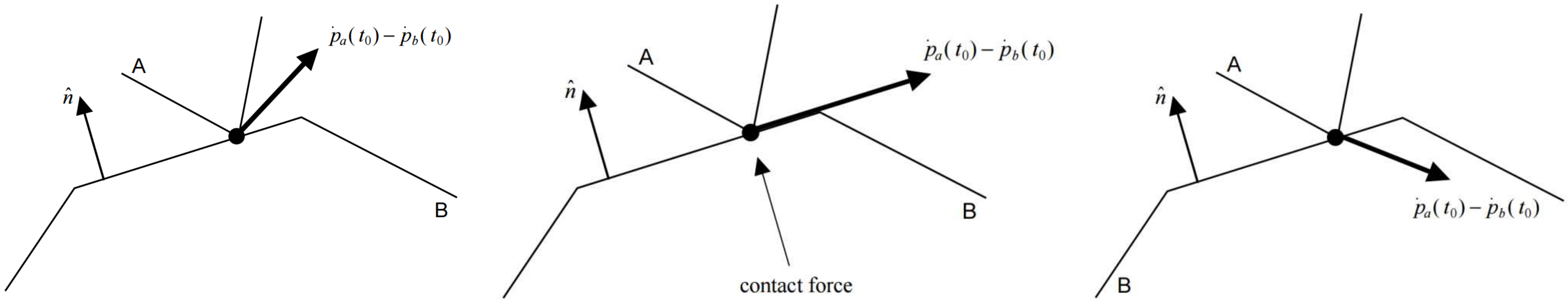
  - Formula

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

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

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

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

  - 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

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

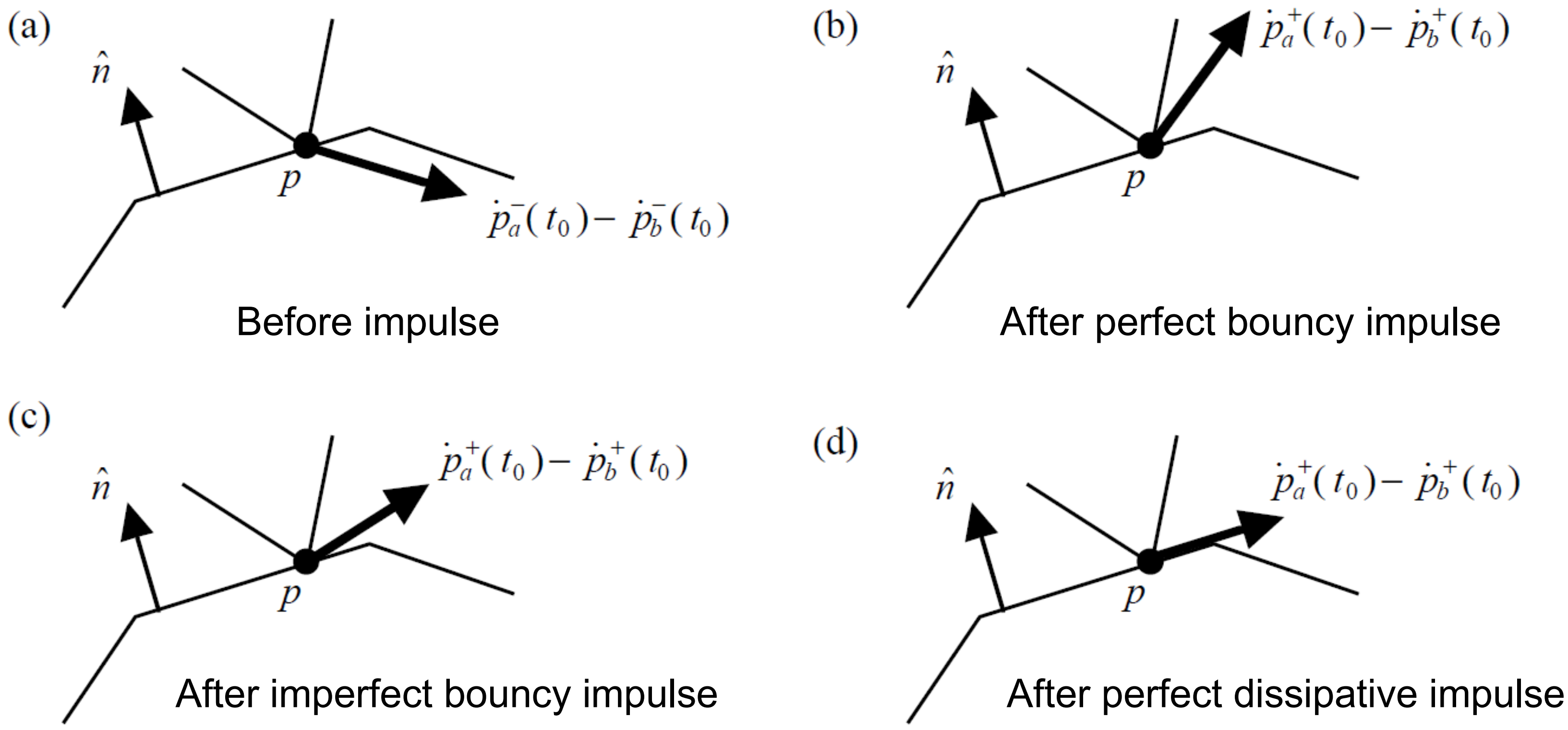
  - Perfect dissipative
    - A maximum of kinetic energy is lost

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

      - After this collision, the two bodies will be in rest contact

Colliding contact

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



Before impulse

After perfect bouncy impulse

After imperfect bouncy impulse

After perfect dissipative impulse

Colliding contact

- **Derivation**

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

$$\Downarrow$$

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

$$\Downarrow$$

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

Colliding contact

- **This yields**

$$\dot{p}_a^+(t_0) - \dot{p}_b^+ = (\dot{p}_a^-(t_0) - \dot{p}_b^-) + j\Bigg(\frac{\hat{n}(t_0)}{M_a} + \frac{\hat{n}(t_0)}{M_b} +$$

$$\left(I_a^{-1}(t_0)\left(r_a \times \hat{n}(t_0)\right)\right) \times r_a + \left(I_b^{-1}(t_0)\left(r_b \times \hat{n}(t_0)\right)\right) \times r_b\Bigg)$$

$$\Downarrow$$

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

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

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

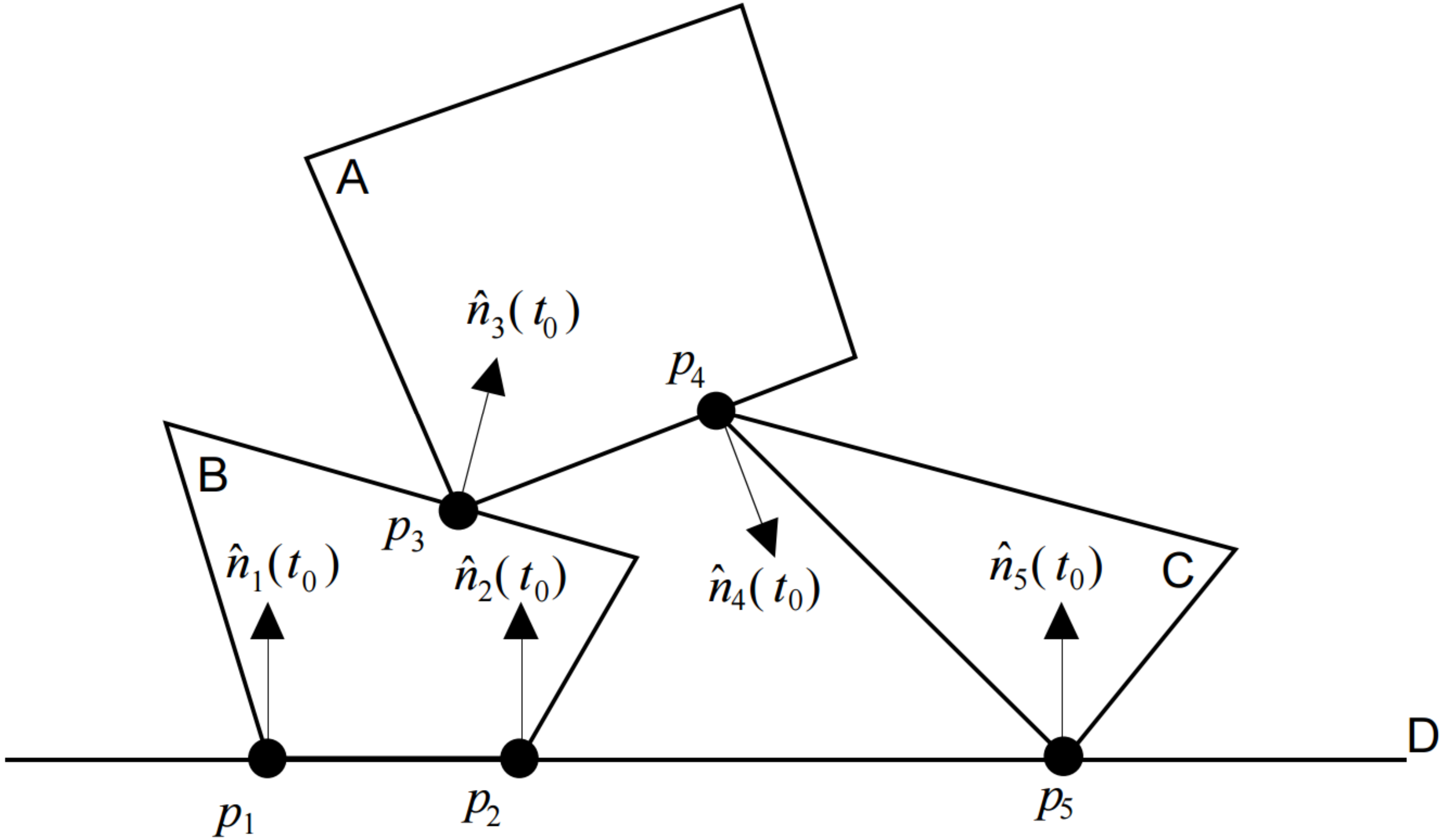
  - What we need
    - Inverse of mass and inertia tensor
  - Tricks
    - Set inverse mass to be zero
    - Set inverse inertia tensor to be zero matrix

Resting contact

- **Condition of resting contact**
  - Relative velocity v_{rel} is zero (within numerical threshold)
  - Contact force $f_i \hat{n}_i(t_0)$
    - At each contact point, there is a contact force 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



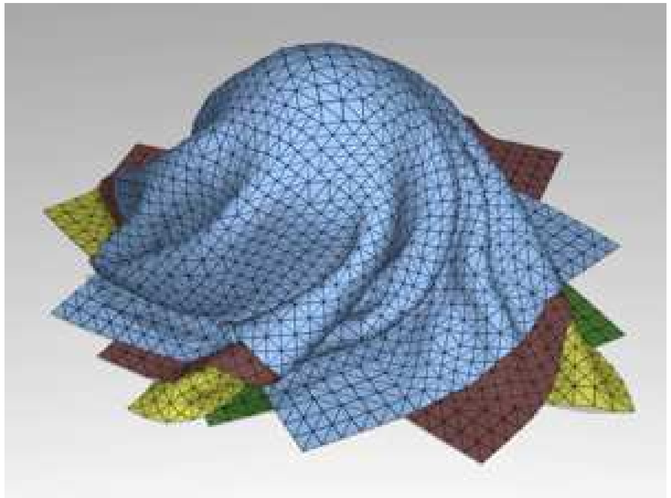
Resting contact

- **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

4. Collision detection

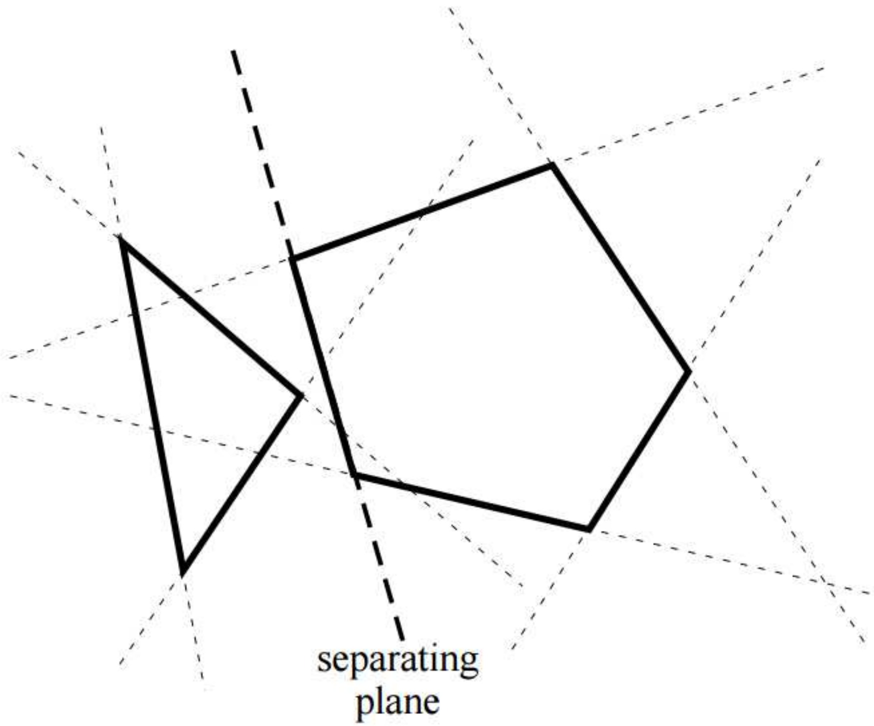
Collision Detection Problem

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



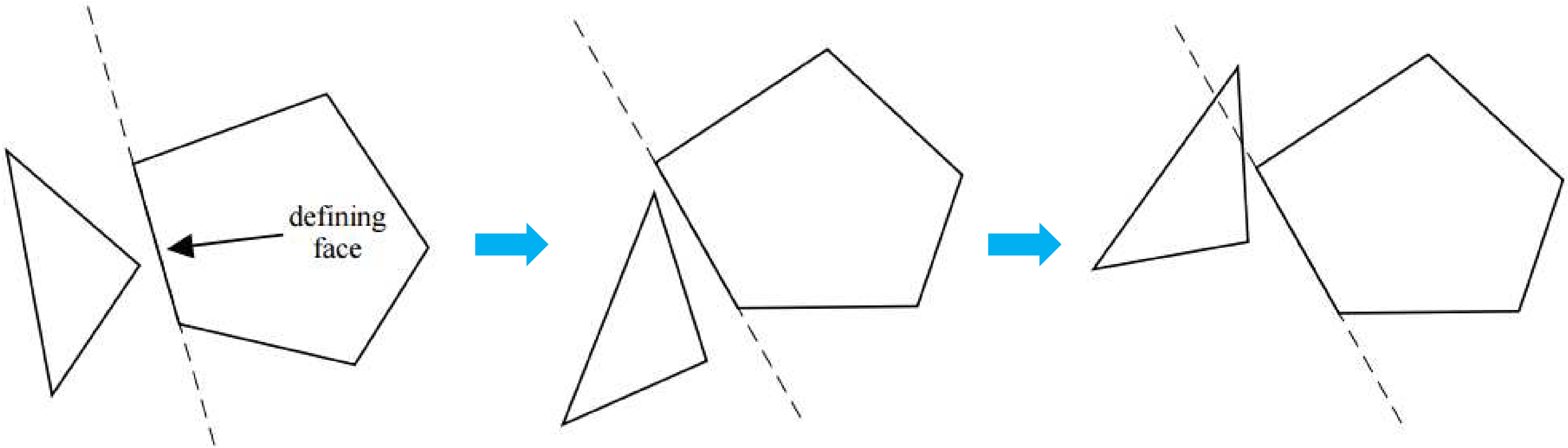
Collision detection

- **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



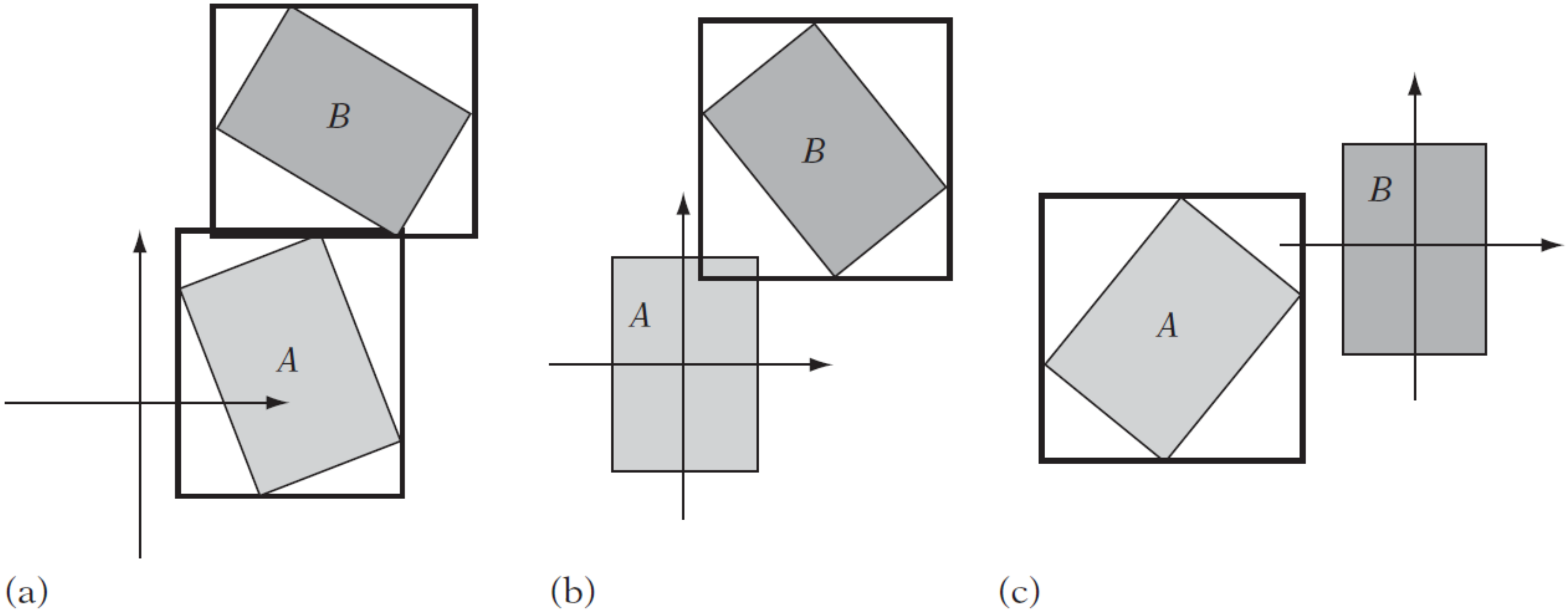
Collision detection

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



Bounding volumes

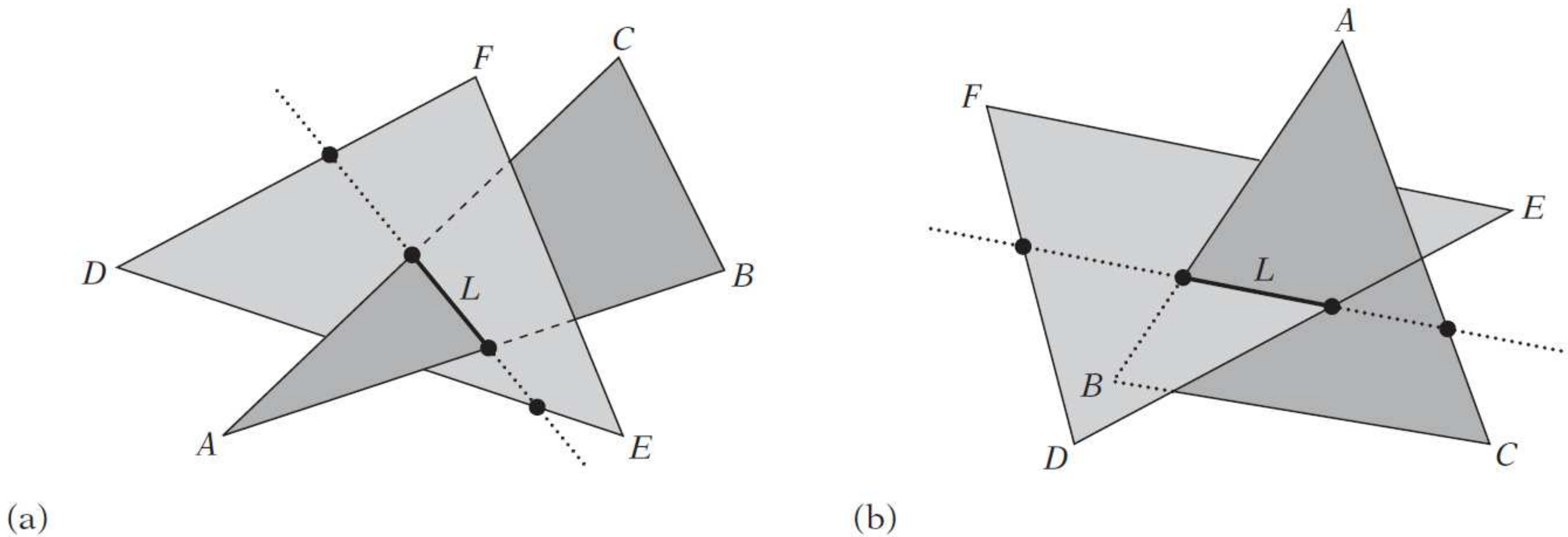
- **Axis-aligned bounding boxes (AABBs)**
  - AABBs in terms of different coordinate system



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

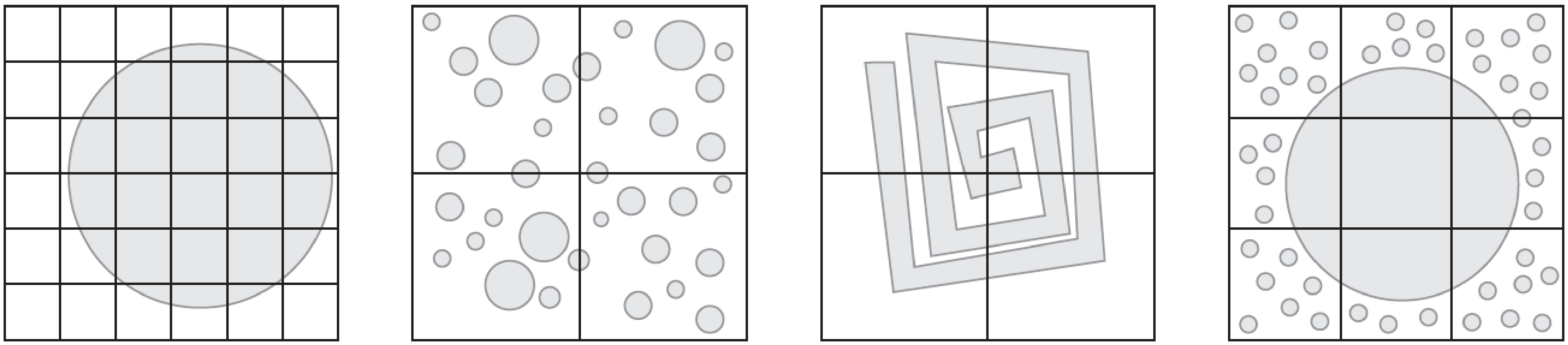
Basic primitive tests

- **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

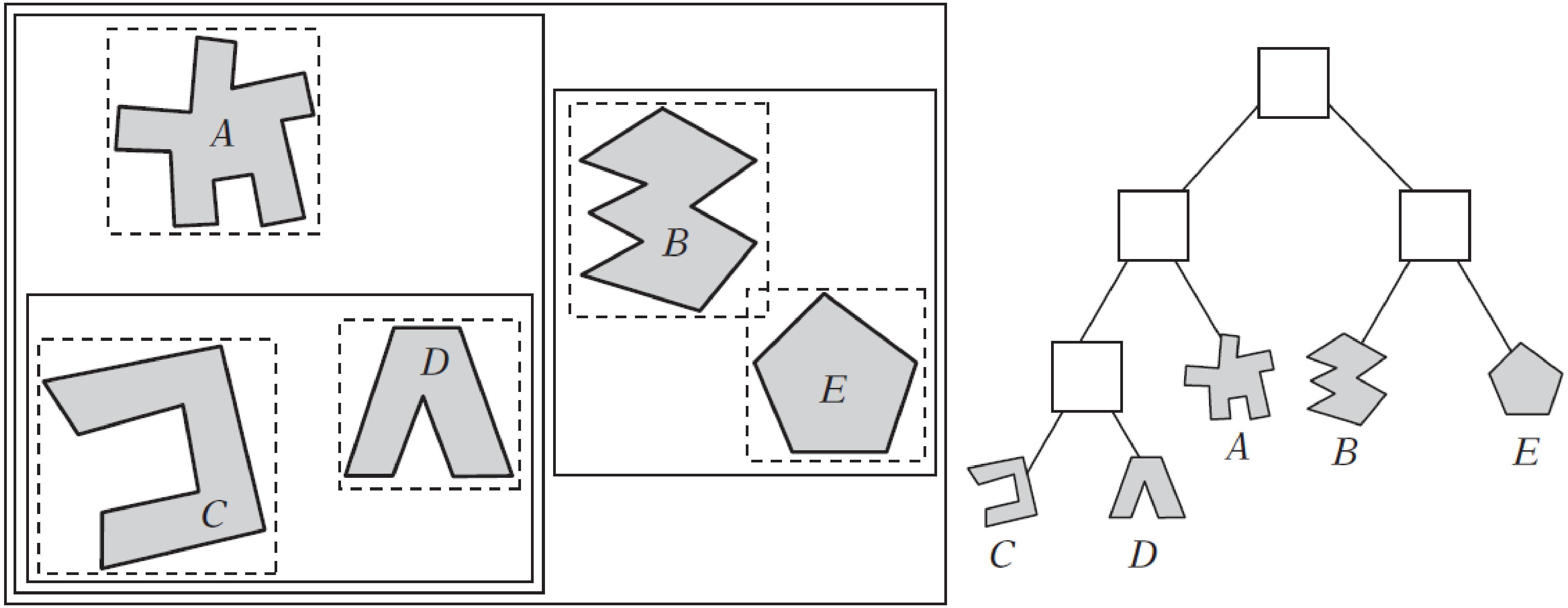
- **Uniform grids**
  - In-depth tests are only performed against those found sharing cells



(a) (b) (c) (d)

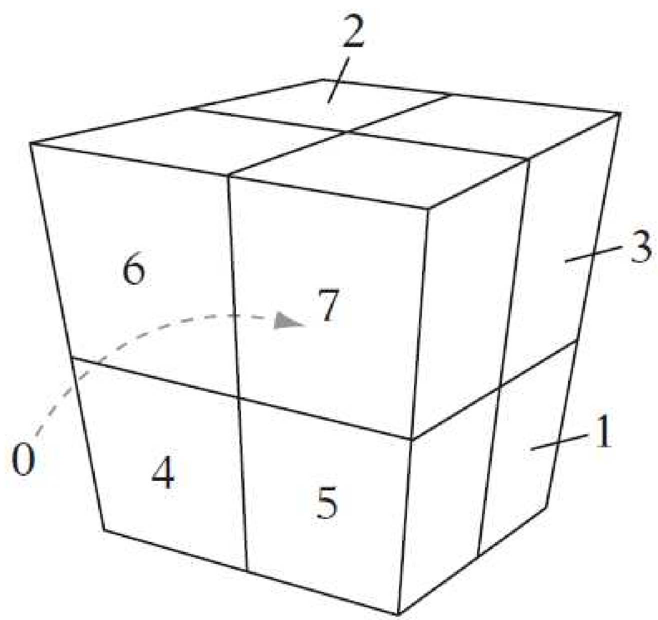
Bounding volume hierarchies

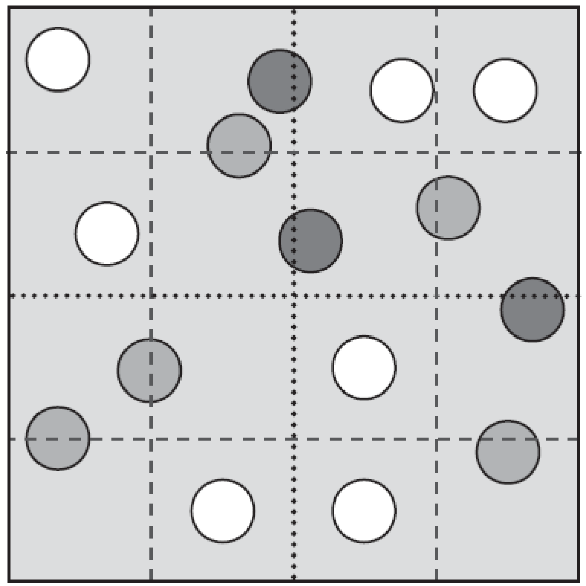
- **Bounding volume hierarchy (BVH)**
  - Time complexity can be reduced to logarithmic in the number of tests performed



Spatial partitioning

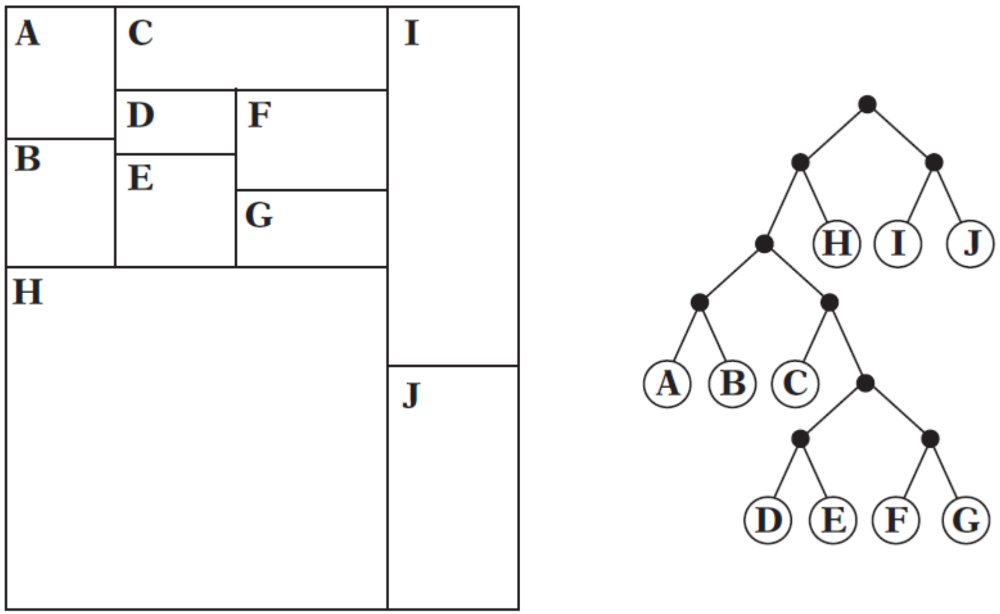
- **Trees**
  - Octree (quadtree for 2D)
    - An axis-aligned hierarchical partitioning of a volume of 3D world space





Spatial partitioning

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

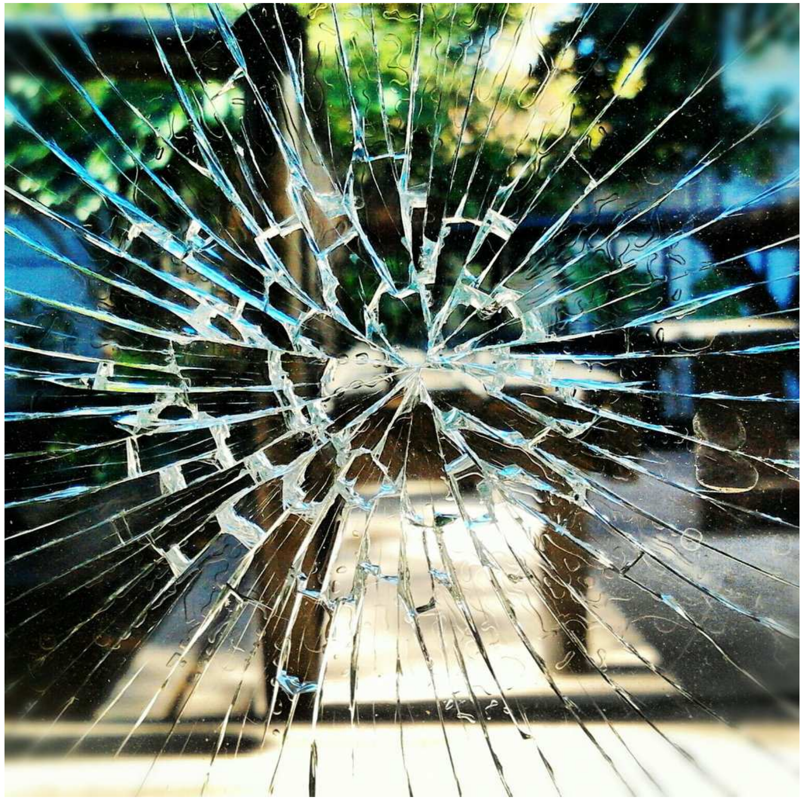


(a) (b)

5. Rigid body fracture

What is a fracture?

- **A fracture**
  - The separation of an object or material into two or more pieces under the action of stress



How to model fracture?

- **Consider material deformations**
  - Even rigid body objects have small deformations
  - Deformation causes change of internal stress
  - Fracture arises when internal stress exceed the material toughness (strength)

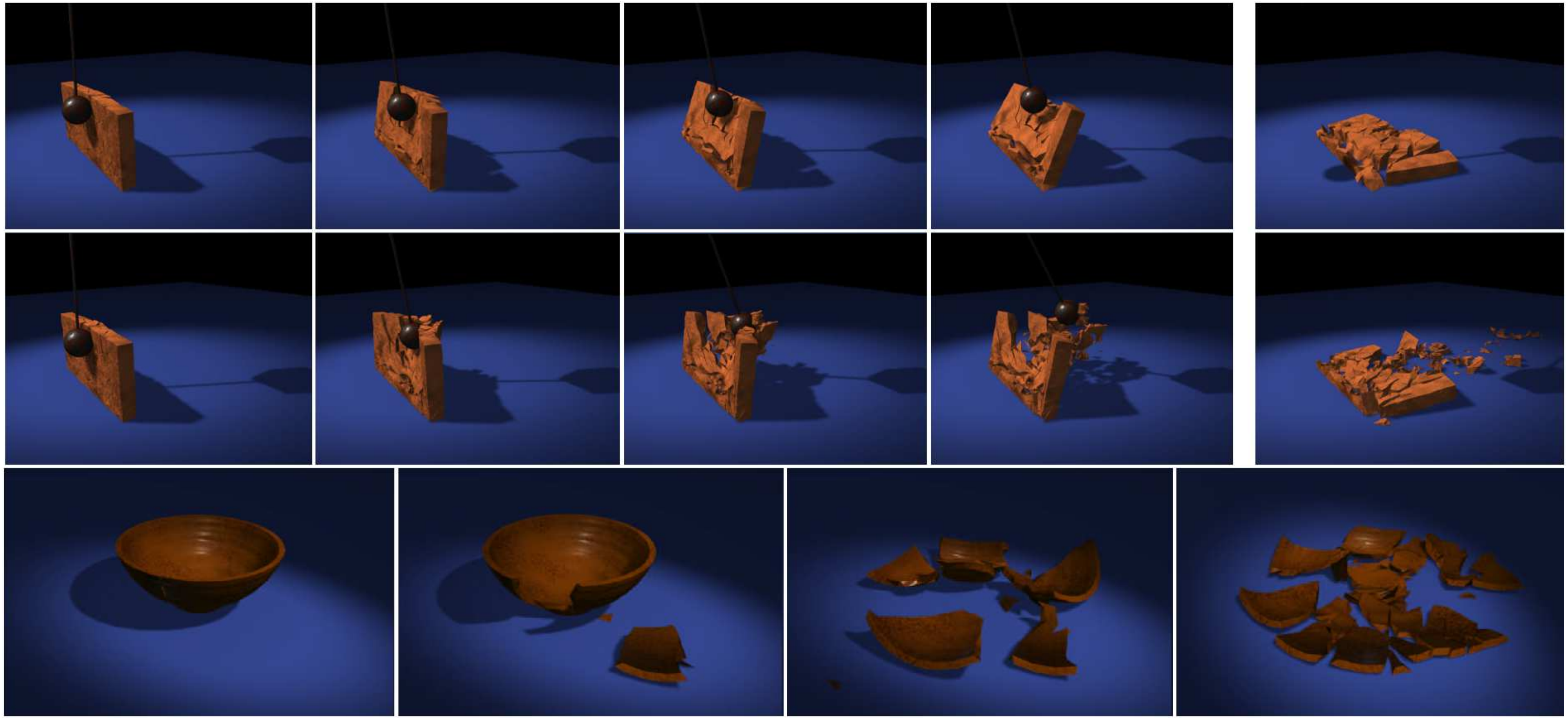
- **Modeling**
  - Computation of internal stress distribution
  - Determine the fracture point and fracture geometry

Continuum mechanics

- **A branch of mechanics**
  - Modeled as a continuous mass rather than as discrete particles
  - The matter in the body is continuously distributed
  - A continuum is a body that can be continually sub-divided into infinitesimal elements
    - Derivatives are available to compute
  - Deal with deformable bodies
    - As opposed to ideal rigid bodies
    - Analyzing internal force of rigid bodies should consider deformation (very small)

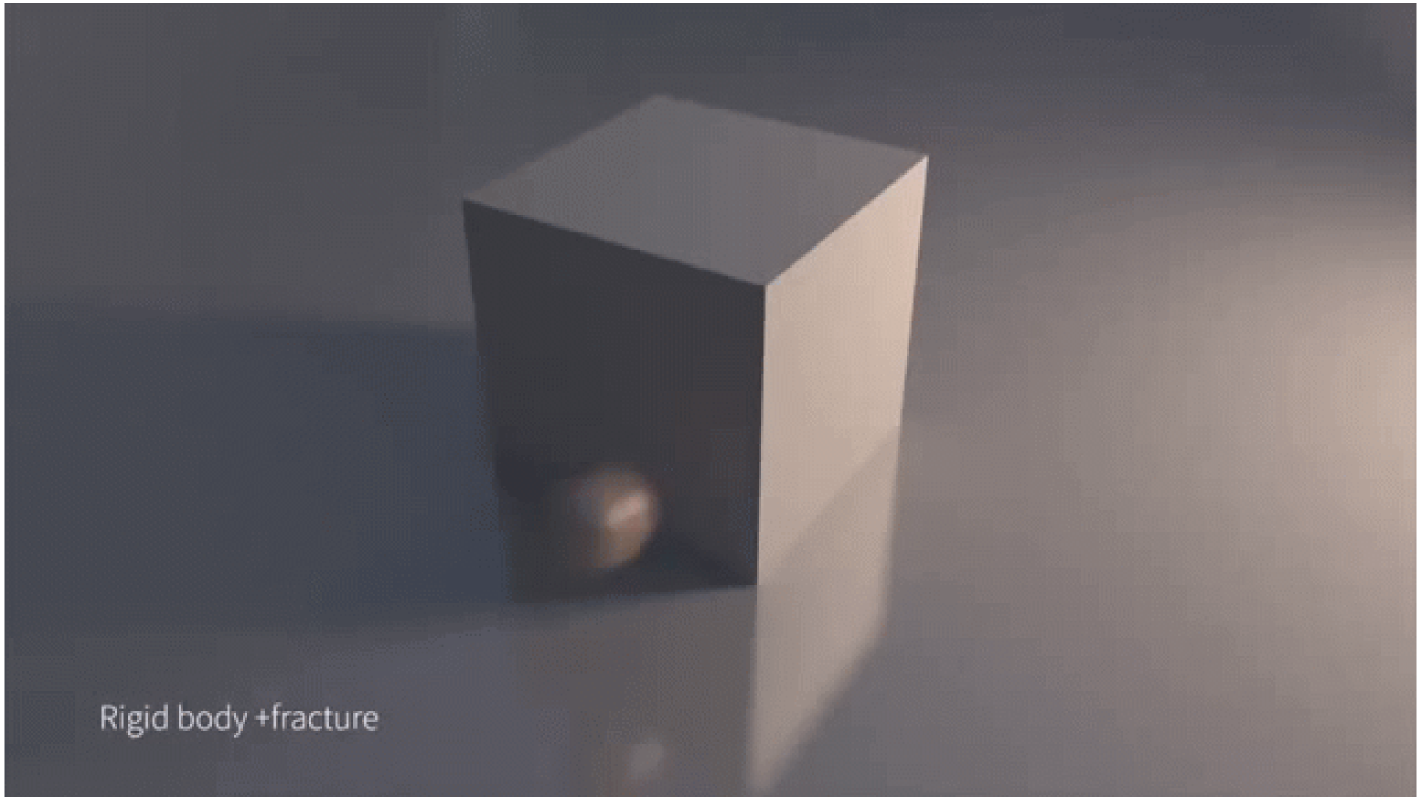
Simulation results

- **Fracture-based rigid body simulation**



What you will get finally?

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



Next lecture: Computer anmiation 3