Rigid Body Dynamics

Simulation in Computer Graphics

Computer Graphics

University of Freiburg

WS 08/09

Motivation

www.hairyharry.de

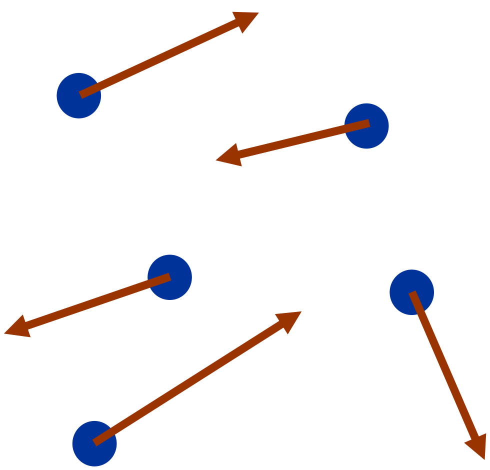
Outline

- representation of a rigid body
  - position (center of mass)
  - orientation
- rigid body kinematics
  - linear velocity
  - angular velocity
- rigid body dynamics
  - inertia tensor
  - linear and angular momentum
  - force and torque
  - simulation loop
- orientation
- contact handling

Particle System vs. Rigid Body

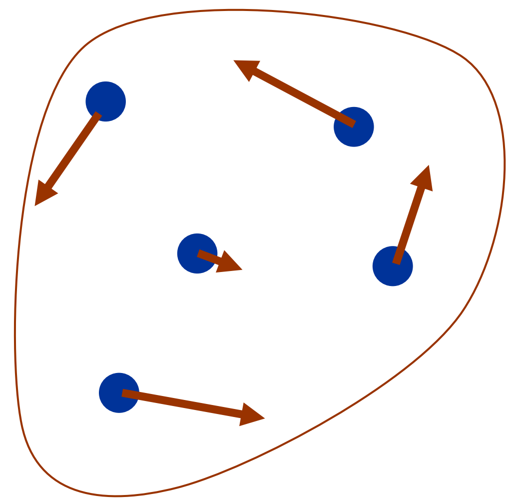
Particle System

- 3n degrees of freedom (dof)
- interaction modeled explicitly
- system of 3n unknown
- no notion of orientation
- scalar mass

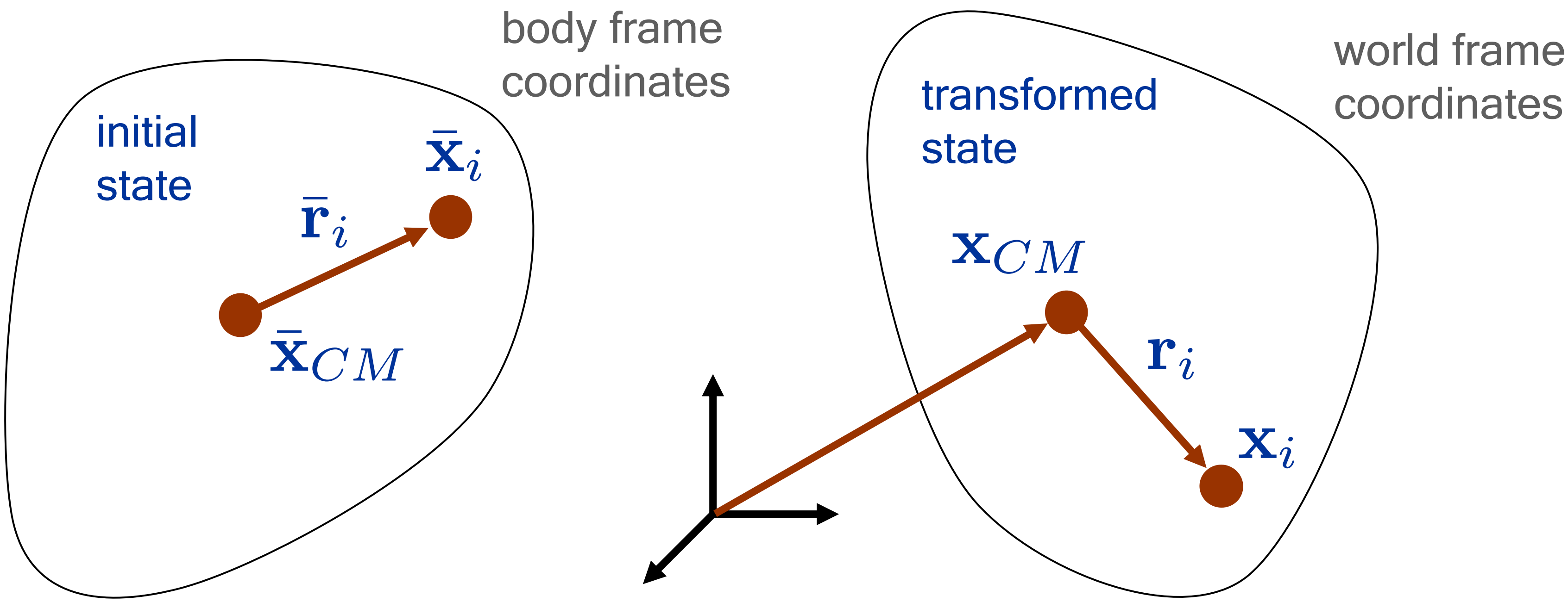


Rigid Body

- springs with infinite stiffness
- modeled implicitly
- 6 remaining dof (position and orientation of the entire body)
- meaningful orientation
- mass distribution



Representation of a Rigid Body



- body frame (local coordinate system)

$$\bar{\mathbf{x}}_{CM} = (0,0,0)^{\mathrm{T}}$$

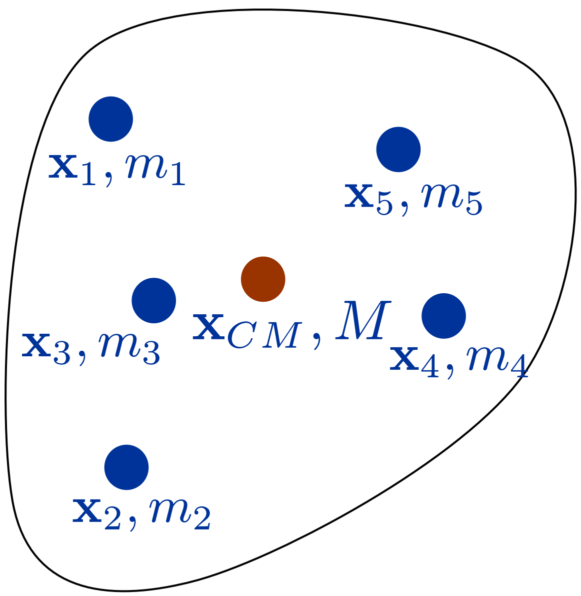
$$\bar{\mathbf{r}}_i = \bar{\mathbf{x}}_i - \bar{\mathbf{x}}_{CM}$$

- world frame (global coordinate system)

$$\mathbf{x}_i = \mathbf{x}_{CM} + \mathbf{r}_i = \bar{\mathbf{x}}_{CM} + \mathbf{t} + \mathrm{Rot}(\bar{\mathbf{r}}_i)$$

translation + rotation

Center of Mass (Massenmittelpunkt, Schwerpunkt)

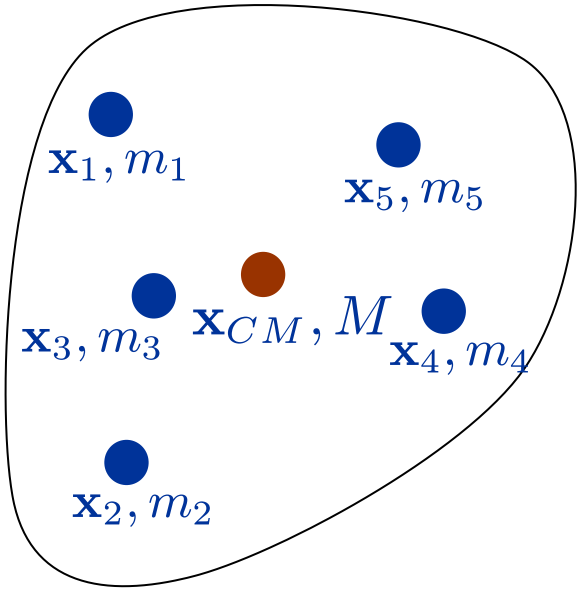


$$\mathbf{x}_{CM} = \frac{\sum m_i \mathbf{x}_i}{\sum m_i} = \frac{\sum m_i \mathbf{x}_i}{M}$$

- same point under translation and rotation

$$M\mathbf{x}_{CM} = \sum m_i \mathbf{x}_i$$

Center of Mass - Motivation



$$M\mathbf{x}_{CM} = \sum m_i \mathbf{x}_i$$

Newton's Second Law

$$\mathbf{f}_i = m_i \ddot{\mathbf{x}}_i$$

$$\mathbf{F} = \sum \mathbf{f}_i = \sum m_i \ddot{\mathbf{x}}_i = \frac{\partial^2}{\partial t^2} \sum m_i \mathbf{x}_i$$

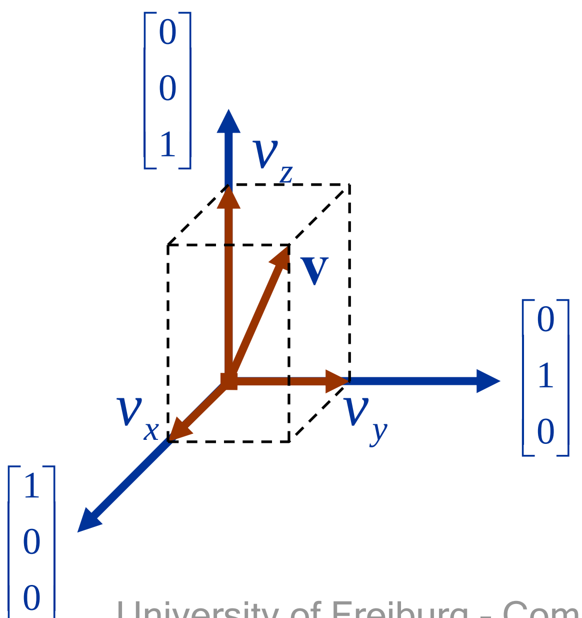
$$= \frac{\partial^2}{\partial t^2} M\mathbf{x}_{CM} = M\ddot{\mathbf{x}}_{CM}$$

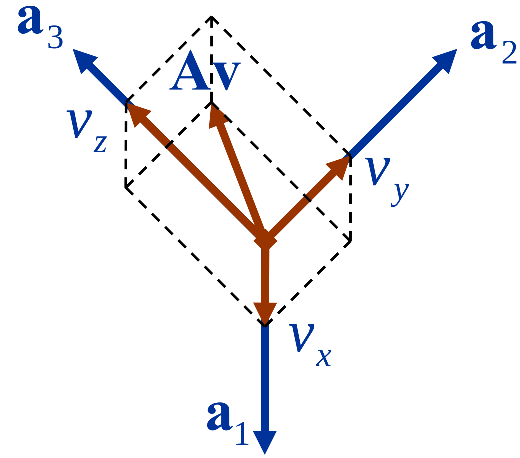
$$\mathbf{F} = M\ddot{\mathbf{x}}_{CM}$$

Orientation in 3D

- rotation matrix
  - 3x3 elements to represent 3 dof

$$\mathrm{Rot}(\mathbf{v}) = \mathbf{A}\mathbf{v} = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{pmatrix} \cdot \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = v_x\mathbf{a}_1 + v_y\mathbf{a}_2 + v_z\mathbf{a}_3$$





Orientation in 3D

- $\mathbf{A}$ has to be orthonormal

$$\mathbf{A}\mathbf{A}^{\mathrm{T}} = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a}_1^{\mathrm{T}} \\ \mathbf{a}_2^{\mathrm{T}} \\ \mathbf{a}_3^{\mathrm{T}} \end{pmatrix} = \mathbf{I}$$

$$\mathrm{Det}(\mathbf{A}) = 1$$

- magnitude of eigenvalues is 1, so $\mathbf{A}$ preserves the length of $\mathbf{r}$
- position of a body point $\mathbf{x}_i = \mathbf{x}_{CM} + \mathbf{r}_i = \bar{\mathbf{x}}_{CM} + \mathbf{t} + \mathbf{A}\bar{\mathbf{r}}_i$

translation + rotation

Outline

- representation of a rigid body
  - position (center of mass)
  - orientation
- rigid body kinematics
  - linear velocity
  - angular velocity
- rigid body dynamics
  - inertia tensor
  - linear and angular momentum
  - force and torque
  - simulation loop
- orientation
- contact handling

Body in Motion

- time-dependent position

$$\mathbf{x}_i(t) = \mathbf{x}_{CM}(t) + \mathbf{A}(t)\bar{\mathbf{r}}_i$$

- velocity

$$\dot{\mathbf{x}}_i = \dot{\mathbf{x}}_{CM}(t) + \dot{\mathbf{A}}\bar{\mathbf{r}}_i + \mathbf{A}\dot{\bar{\mathbf{r}}}_i = \dot{\mathbf{x}}_{CM}(t) + \dot{\mathbf{A}}\bar{\mathbf{r}}_i$$

linear velocity + angular velocity

Angular Velocity in 3D (Winkelgeschwindigkeit)

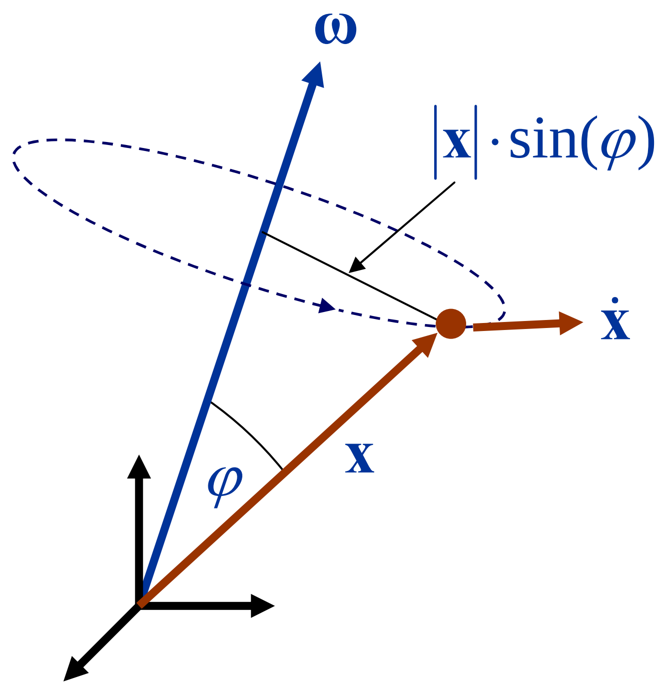
- angular velocity ω is a 3D vector
  - in direction of axis of rotation
  - $|\omega|$ is the magnitude of the angular velocity [rad/s]

$$|\dot{\mathbf{x}}| = |\omega| \cdot r = |\omega| \cdot |\mathbf{x}| \cdot \sin(\phi)$$

$$\dot{\mathbf{x}} = \omega \times \mathbf{x}$$

$$\omega = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad \rightarrow \quad \tilde{\omega} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$$

$$\dot{\mathbf{x}} = \tilde{\omega} \cdot \mathbf{x}$$



Rigid Body Kinematics

- what is the relation between $\tilde{\omega}$ and $\dot{\mathbf{A}}$?
- angular velocity rotates all axis (columns of $\mathbf{A}$)

$$\dot{\mathbf{A}} = \begin{pmatrix} \dot{\mathbf{a}}_1 & \dot{\mathbf{a}}_2 & \dot{\mathbf{a}}_3 \end{pmatrix} = \begin{pmatrix} \tilde{\omega} \cdot \mathbf{a}_1 & \tilde{\omega} \cdot \mathbf{a}_2 & \tilde{\omega} \cdot \mathbf{a}_3 \end{pmatrix} = \tilde{\omega} \cdot \mathbf{A}$$

- position of a point $\mathbf{x}_i(t) = \mathbf{x}_{CM}(t) + \mathbf{A}(t) \cdot \bar{\mathbf{r}}_i$
- velocity of a point $\dot{\mathbf{x}}_i(t) = \dot{\mathbf{x}}_{CM}(t) + \dot{\mathbf{A}}(t) \cdot \bar{\mathbf{r}}_i + \mathbf{A}(t) \cdot \dot{\bar{\mathbf{r}}}_i$

$$\dot{\mathbf{x}}_i(t) = \mathbf{v}(t) + \tilde{\omega}(t) \cdot \mathbf{A}(t) \cdot \bar{\mathbf{r}}_i$$

$$\dot{\mathbf{x}}_i(t) = \mathbf{v}(t) + \tilde{\omega}(t) \cdot (\mathbf{x}_i(t) - \mathbf{x}_{CM}(t))$$

State Vector

particle

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

rigid body

$$\begin{pmatrix} \mathbf{x}(t) \\ \mathbf{A}(t) \\ \mathbf{v}(t) \\ \omega(t) \end{pmatrix}$$

Outline

- representation of a rigid body
  - position (center of mass)
  - orientation
- rigid body kinematics
  - linear velocity
  - angular velocity
- rigid body dynamics
  - inertia tensor
  - linear and angular momentum
  - force and torque
  - simulation loop
- orientation
- contact handling

Rigid Body Dynamics

- forces change
  - linear velocity
  - angular velocity

- linear velocity change

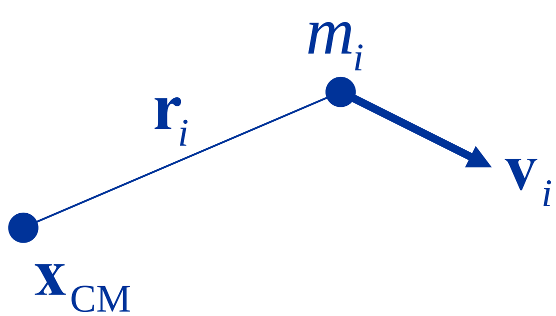
$$\mathbf{F} = \sum \mathbf{f}_i = \sum m_i \ddot{\mathbf{x}}_i = \frac{\partial^2}{\partial t^2} \sum m_i \mathbf{x}_i$$

$$= \frac{\partial^2}{\partial t^2} M \mathbf{x}_{CM} = M \ddot{\mathbf{x}}_{CM}$$

$$\ddot{\mathbf{x}}_{CM} = \frac{\mathbf{F}}{M} = \frac{\sum \mathbf{f}_i}{M}$$

- like a particle, but ...

Angular Momentum (Drehimpuls)



- the angular momentum of a particle w.r.t. the center of mass is

$$\mathbf{L}_i = \mathbf{r}_i \times m_i \mathbf{v}_i = \mathbf{r}_i \times m_i(\omega \times \mathbf{r}_i)$$

- the total angular momentum of the body is

$$\mathbf{L} = \sum \mathbf{L}_i = \sum \mathbf{r}_i \times m_i(\omega \times \mathbf{r}_i)$$

$$= \sum -m_i \tilde{\mathbf{r}}_i \tilde{\mathbf{r}}_i \omega = \left(\sum -m_i \tilde{\mathbf{r}}_i \tilde{\mathbf{r}}_i\right) \cdot \omega$$

$$= \mathbf{I}\omega$$

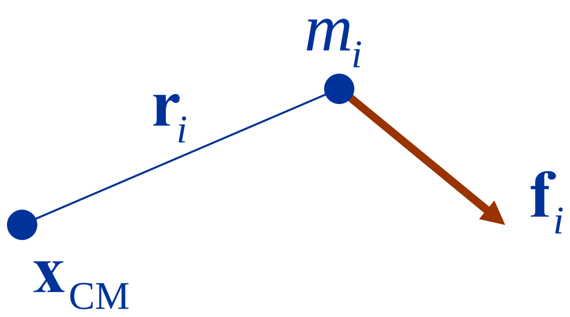
Inertia Tensor (Trägheitstensor)

- the total angular momentum is $\mathbf{L} = \mathbf{I}\omega$
  with $\mathbf{I}$ being a 3x3 matrix (the inertia tensor of the body)
- $\mathbf{I}$ depends on the rotated configuration $\mathbf{I} = (\sum -m_i \tilde{\mathbf{r}}_i \tilde{\mathbf{r}}_i)$
- the inertia tensor for the original body can be pre-computed,
  e. g. $\bar{\mathbf{I}} = (\sum -m_i \tilde{\bar{\mathbf{r}}}_i \tilde{\bar{\mathbf{r}}}_i)$

- similarity transform relates time-dependent $\mathbf{I}$ and
  pre-computed $\bar{\mathbf{I}}$ $\quad \mathbf{I} = \mathbf{A}\bar{\mathbf{I}}\mathbf{A}^{\mathrm{T}}$

$$\mathbf{L} = \mathbf{I}\omega = \mathbf{A}\bar{\mathbf{I}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\bar{\omega} = \mathbf{A}\bar{\mathbf{I}}\bar{\omega}$$

Torque (Drehmoment)



- the torque of a particle w.r.t. the center of mass is

$$\tau_i = \mathbf{r}_i \times \mathbf{f}_i$$

- the total torque of the body is

$$\tau = \sum \tau_i = \sum \mathbf{r}_i \times \mathbf{f}_i$$

Newton's Second Law (Angular)

- angular momentum $\mathbf{L} = \sum \mathbf{r}_i \times m_i \mathbf{v}_i = \mathbf{I}\omega$
- torque $\tau = \sum \mathbf{r}_i \times \mathbf{f}_i$
- the angular version of Newton's Second law reads
  $\dot{\mathbf{L}} = \tau$
- tells us, how the forces $\mathbf{f}_i$ change the angular velocity ω
  $\tau = \sum \mathbf{r}_i \times \mathbf{f}_i$
  $\mathbf{L} = \mathbf{L} + \Delta t \cdot \tau$
  $\omega = \mathbf{I}^{-1}\mathbf{L}$

Linear vs. Angular Quantities

linear momentum

$$\mathbf{p} = M\mathbf{v}$$

angular momentum

$$\mathbf{L} = \mathbf{I}\omega$$

linear velocity

$$\mathbf{v} = M^{-1}\mathbf{p}$$

angular velocity

$$\omega = \mathbf{I}^{-1}\mathbf{L}$$

time-derivative of
the linear momentum

$$\dot{\mathbf{p}} = \mathbf{F}$$

time-derivative of
the angular momentum

$$\dot{\mathbf{L}} = \tau$$

Dynamics

$$\frac{d}{dt}\mathbf{X}(t) = \frac{d}{dt}\begin{pmatrix} \mathbf{x}(t) \\ \mathbf{A}(t) \\ \mathbf{p}(t) \\ \mathbf{L}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \tilde{\omega} \cdot \mathbf{A}(t) \\ \mathbf{F}(t) \\ \tau(t) \end{pmatrix}$$

Simulation Loop (Euler)

Pre-compute:

$$M \leftarrow \sum m_i$$

$$\bar{\mathbf{x}}_{CM} \leftarrow \left(\sum \bar{\mathbf{x}}_i m_i\right) / M$$

$$\bar{\mathbf{r}}_i \leftarrow \bar{\mathbf{x}}_i - \bar{\mathbf{x}}_{CM}$$

$$\bar{\mathbf{I}}^{-1} \leftarrow \left(\sum -m_i \tilde{\bar{\mathbf{r}}}_i \tilde{\bar{\mathbf{r}}}_i\right)^{-1}$$

Initialize:

$$\mathbf{x}_{CM}, \mathbf{v}_{CM}, \mathbf{A}, \mathbf{L}$$

$$\mathbf{I}^{-1} \leftarrow \mathbf{A}\bar{\mathbf{I}}^{-1}\mathbf{A}^T$$

$$\boldsymbol{\omega} \leftarrow \mathbf{I}^{-1}\mathbf{L}$$

Loop:

sum up external forces

$$\boldsymbol{\tau} \leftarrow \sum \mathbf{r}_i \times \mathbf{f}_i$$

$$\mathbf{F} \leftarrow \sum \mathbf{f}_i$$

perform Euler integration step

$$\mathbf{x}_{CM} \leftarrow \mathbf{x}_{CM} + \Delta t \cdot \mathbf{v}_{CM}$$

$$\mathbf{v}_{CM} \leftarrow \mathbf{v}_{CM} + \Delta t \cdot \mathbf{F} / M$$

$$\mathbf{A} \leftarrow \mathbf{A} + \Delta t \cdot \tilde{\boldsymbol{\omega}}\mathbf{A}$$

$$\mathbf{L} \leftarrow \mathbf{L} + \Delta t \cdot \boldsymbol{\tau}$$

$$\mathbf{I}^{-1} \leftarrow \mathbf{A}\bar{\mathbf{I}}^{-1}\mathbf{A}^T$$

$$\boldsymbol{\omega} \leftarrow \mathbf{I}^{-1}\mathbf{L}$$

per particle quantities

$$\mathbf{r}_i \leftarrow \mathbf{A} \cdot \bar{\mathbf{r}}_i$$

$$\mathbf{x}_i \leftarrow \mathbf{x}_{CM} + \mathbf{r}_i$$

$$\mathbf{v}_i \leftarrow \mathbf{v}_{CM} + \boldsymbol{\omega} \times \mathbf{r}_i$$

Reorthonormalization of the Orientation

- orientation matrix is updated every time step

  $$\mathbf{A} \leftarrow \mathbf{A} + \Delta t \cdot \tilde{\boldsymbol{\omega}} \mathbf{A}$$

- errors accumulate
- $\mathbf{A}$ is not orthonormal anymore
- use Gram-Schmidt orthonormalization

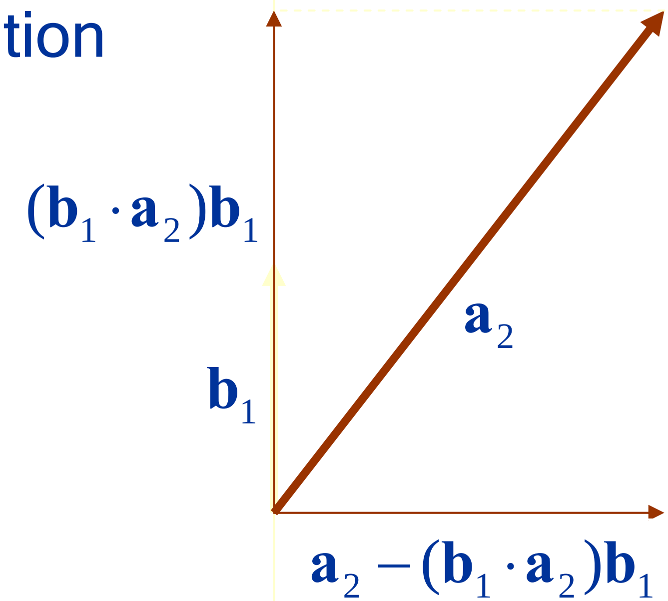
$$\mathbf{b}_1 = \mathbf{a}_1 / |\mathbf{a}_1|$$

$$\mathbf{b}_2 = \mathbf{a}_2 - (\mathbf{b}_1 \cdot \mathbf{a}_2)\mathbf{b}_1$$

$$\mathbf{b}_2 = \mathbf{b}_2 / |\mathbf{b}_2|$$

$$\mathbf{b}_3 = \mathbf{a}_3 - (\mathbf{b}_1 \cdot \mathbf{a}_3)\mathbf{b}_1 - (\mathbf{b}_2 \cdot \mathbf{a}_3)\mathbf{b}_2$$

$$\mathbf{b}_3 = \mathbf{b}_3 / |\mathbf{b}_3|$$

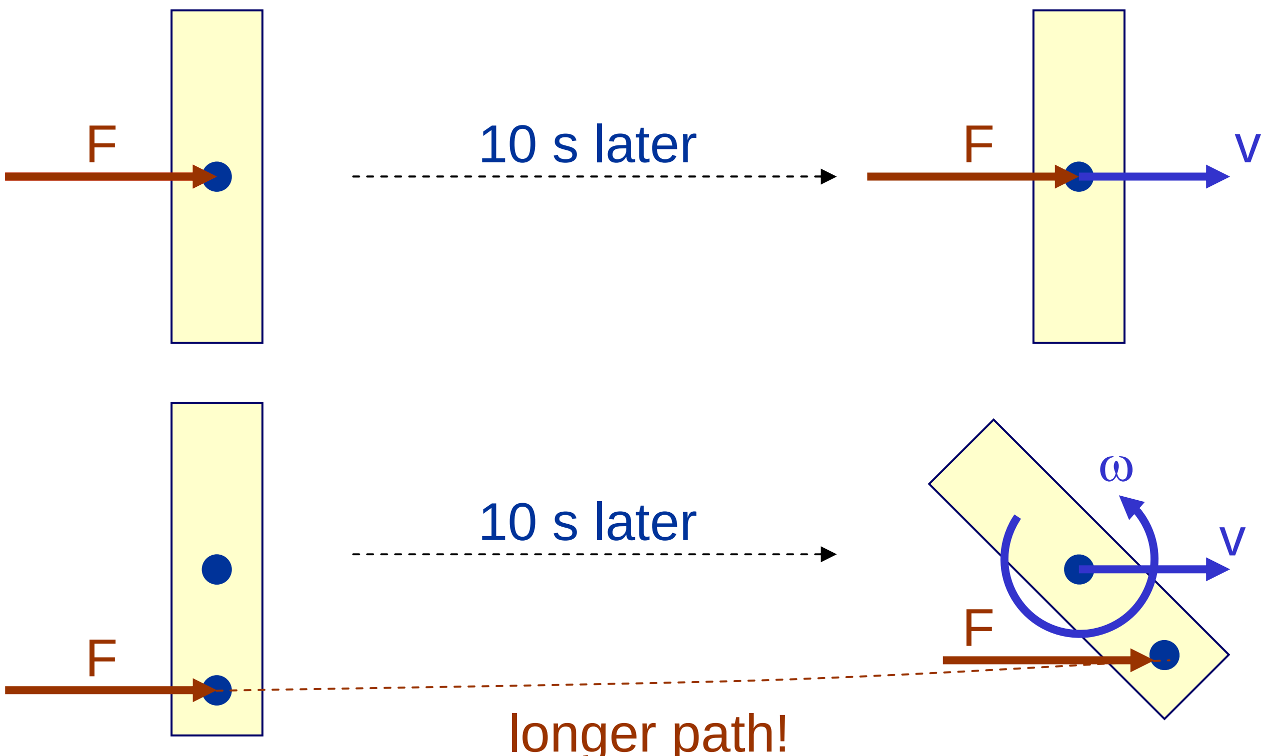


Force vs. Torque Puzzle

- is force being considered twice?
  - to accelerate center of mass
  - to cause the body to spin

$$\mathbf{F} = \sum \mathbf{f}_i$$

$$\tau = \sum \mathbf{r}_i \times \mathbf{f}_i$$



$$E = \frac{1}{2} M \mathbf{v}^T \mathbf{v}$$

$$E = \frac{1}{2} M \mathbf{v}^T \mathbf{v} + \frac{1}{2} \boldsymbol{\omega}^T \mathbf{I} \boldsymbol{\omega}$$

Outline

- representation of a rigid body
  - position (center of mass)
  - orientation
- rigid body kinematics
  - linear velocity
  - angular velocity
- rigid body dynamics
  - inertia tensor
  - linear and angular momentum
  - force and torque
  - simulation loop
- orientation
- contact handling

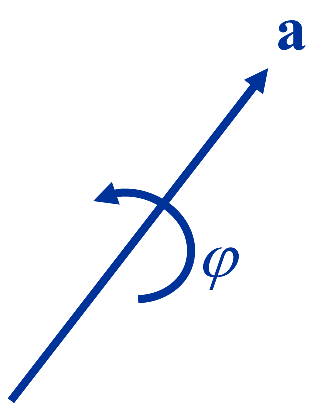
Alternative Orientation Representations

- quaternions

  $$\mathbf{q} = (w, x, y, z)$$

- 4 components represent 3 dof

  $$\mathbf{q} = (cos(\frac{\phi}{2}), sin(\frac{\phi}{2}) \cdot (a_x, a_y, a_z))$$



- often used in rigid body computations for rotations

  $$\mathrm{Rot}(\mathbf{v}) = \mathbf{q} \circ (0, v_x, v_y, v_z) \circ \mathbf{q}^{-1}$$

Quaternions

- quaternions are an extension of complex numbers

$$\mathbf{q} = (w, \mathbf{v}) = w + \mathbf{v} = (w, x, y, z)$$
$$= w + x \cdot \mathbf{i} + y \cdot \mathbf{j} + z \cdot \mathbf{k}$$

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$$

$$\mathbf{i} \cdot \mathbf{j} = \mathbf{k}, \quad \mathbf{j} \cdot \mathbf{k} = \mathbf{i}, \quad \mathbf{k} \cdot \mathbf{i} = \mathbf{j}$$

$$\mathbf{j} \cdot \mathbf{i} = -\mathbf{k}, \quad \mathbf{k} \cdot \mathbf{j} = -\mathbf{i}, \quad \mathbf{i} \cdot \mathbf{k} = -\mathbf{j}$$

Basic Operations

- addition $\mathbf{q_1} + \mathbf{q_2} = (w_1 + w_2) + (\mathbf{v_1} + \mathbf{v_2})$
- dot product $\mathbf{q_1} \cdot \mathbf{q_2} = w_1 w_2 + \mathbf{v_1} \cdot \mathbf{v_2}$
- conjugate $\bar{\mathbf{q}} = w \overline{+} \mathbf{v} = w - \mathbf{v}$
- magnitude (module) $$|\mathbf{q}|^2 = \mathbf{q} \cdot \bar{\mathbf{q}} = w^2 + \mathbf{v} \cdot \mathbf{v} = w^2 + v_x^2 + v_y^2 + v_z^2$$
- inverse $\mathbf{q}^{-1} = \frac{\bar{\mathbf{q}}}{|\mathbf{q}|^2}$

Basic Operations

- multiplication

$$
\begin{aligned}
\mathbf{q_1} \circ \mathbf{q_2} &= (w_1 + \mathbf{v_1})(w_2 + \mathbf{v_2}) \\
&= (w_1 w_2 - \mathbf{v_1} \mathbf{v_2}) + (w_1 \mathbf{v_2} + w_2 \mathbf{v_1} + \mathbf{v_1} \times \mathbf{v_2}) \\
&= (w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2) \\
&\quad + (w_1 x_2 + w_2 x_1 + y_2 z_1 - y_1 z_2)\mathbf{i} \\
&\quad + (w_1 y_2 + w_2 y_1 + z_2 x_1 - z_1 x_2)\mathbf{j} \\
&\quad + (w_1 z_2 + w_2 z_1 + x_2 y_1 - x_1 y_2)\mathbf{k}
\end{aligned}
$$

Unit Quaternions

- quaternions with $|\mathbf{q}| = 1$
  can be used to represent rotations
- corresponding rotation matrix

$$\mathbf{R} = 2 \begin{pmatrix} 0.5 - y^2 - z^2 & xy + wz & xz - wy \\ xy - wz & 0.5 - x^2 - z^2 & yz + wx \\ xz + wy & yz - wx & 0.5 - x^2 - y^2 \end{pmatrix}$$

- rotation of a point

$$\mathbf{R}\mathbf{p} = \mathbf{q} \circ (0, \mathbf{p}^{\mathrm{T}}) \circ \bar{\mathbf{q}}$$

Application in the Simulation Loop

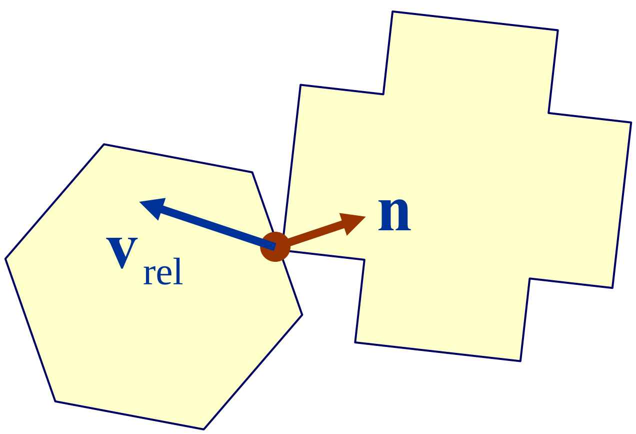
- initialization of orientation $\mathbf{q}$ and angular velocity $\bar{\omega}$ in the body frame
- update of the angular velocity using
  $\dot{\bar{\omega}} = \bar{\mathbf{I}}^{-1}(\bar{\tau} - \bar{\omega} \times (\bar{\mathbf{I}}\bar{\omega}))$
- update of the orientation using
  $\dot{\mathbf{q}} = 0.5\mathbf{q} \circ (0, \bar{\omega}^{\mathrm{T}})$
- updated orientation has to be normalized
- torque needs to be transformed from world to body frame, angular velocity from body to world frame

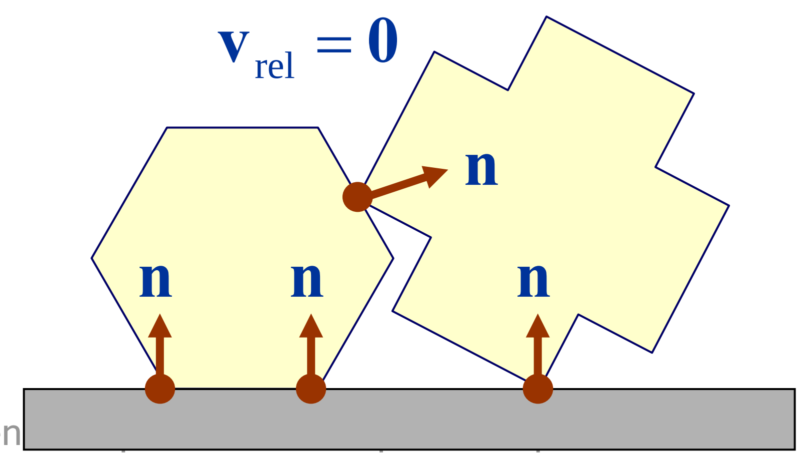
Outline

- representation of a rigid body
  - position (center of mass)
  - orientation
- rigid body kinematics
  - linear velocity
  - angular velocity
- rigid body dynamics
  - inertia tensor
  - linear and angular momentum
  - force and torque
  - simulation loop
- orientation
- contact handling

Non-Penetration

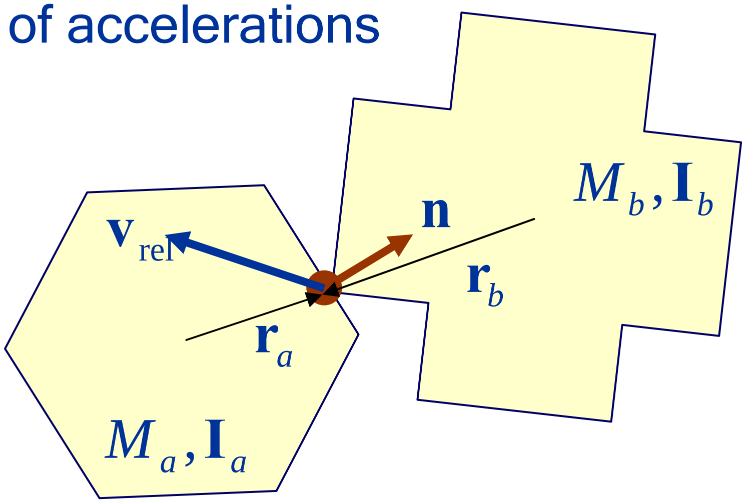
- detect collisions (see collision detection slides)
- avoid penetrations
  - change time step or
  - push body back
- compute collision response
  - colliding contact ("easy")
  - resting contact (very hard)





Colliding Contact

- force driven
  - penetration causes force (late, slow, easy to compute)
- impulse driven
  - manipulation of velocities (J) instead of accelerations (fast, more difficult to compute)
  - e - coefficient of restitution (e=0 plastic c., e=1 elastic c.)



$$\Delta \mathbf{v}_{CM} = \mathbf{J} / M$$

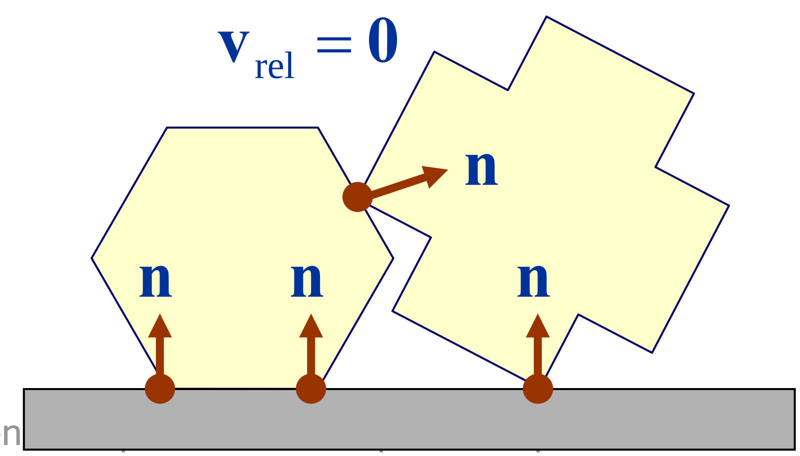
$$\Delta \mathbf{L} = (\mathbf{x}_{\text{impact}} - \mathbf{x}_{CM}) \times \mathbf{J}$$

$$\mathbf{J} = j\mathbf{n}$$

$$j = \frac{-(1+e)v_{\text{rel}}}{\frac{1}{M_a} + \frac{1}{M_b} + \left[(\mathbf{I}_a^{-1}(\mathbf{r}_a \times \mathbf{n})) \times \mathbf{r}_a + (\mathbf{I}_b^{-1}(\mathbf{r}_b \times \mathbf{n})) \times \mathbf{r}_b\right] \cdot \mathbf{n}}$$

Resting Contact

- find all collisions with small relative velocities
- solve for all contact forces simultaneously such that for each contact force
  - the force is strong enough to prevent interpenetration
  - the force is repulsive only (not glue like)
  - the force is zero if the bodies separate
- Linear complementary problem (LCP)



Summary

- representation of a rigid body
  - position (center of mass)
  - orientation
- rigid body kinematics
  - linear velocity
  - angular velocity
- rigid body dynamics
  - inertia tensor
  - linear and angular momentum
  - force and torque
  - simulation loop
- orientation
- contact handling

References

- Murilo G. Coutinho,
  "Dynamic Simulations of Multibody Systems",
  Springer, ISBN 0-387-95192-X
- Andrew Witkin, David Baraff, Michael Kass,
  "Physically-based modeling",
  ACM SIGGRAPH '01, Course Notes, 2001.
- David Eberly,
  "Game Physics",
  Elsevier, ISBN 1-55860-740-4.
- Kenny Erleben, J. Sporring, K. Henriksen, H. Dohlmann,
  "Physics-based Animation",
  Charles River Media, Hingham, Massachusetts,
  ISBN 1-58450-380-7, 2005.
- Jan Bender, **"Impulsbasierte Dynamiksimulation von Mehrkoerpersystemen in der virtuellen Realitaet"**,
  Dissertation, Universitaet Karlsruhe, 2007

Acknowledgement

- This slide set is courtesy of Matthias Mueller, NVIDIA Corp. It has been modified.