



香港大學

THE UNIVERSITY OF HONG KONG

Physics-based animation

Tutorial: Rigid body dynamics

Floyd M. Chitalu & Prof Taku Komura



Overview

- Rigid body dynamics
- Extra work
 - Using quaternions
 - Handling collisions
- Resources

Rigid body dynamics

- State variables

$\mathbf{x}(t)$

$\mathbf{R}(t)$

$\mathbf{P}(t)$

$\mathbf{L}(t)$

Rigid body dynamics

- State variables

$\mathbf{x}(t)$ Position
 $\mathbf{R}(t)$ Orientation
 $\mathbf{P}(t)$ Linear momentum
 $\mathbf{L}(t)$ Angular momentum

Rigid body dynamics

- Change of state variables w.r.t time

$$\frac{d}{dt} \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{R}(t) \\ \mathbf{P}(t) \\ \mathbf{L}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^{\star}(t)\mathbf{R}(t) \\ \mathbf{f}(t) \\ \boldsymbol{\tau}(t) \end{pmatrix}$$

Change in position

Change in orientation

Change linear momentum

Change in angular momentum

Rigid body dynamics

- Change of state variables w.r.t time

$$\frac{d}{dt} \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{R}(t) \\ \mathbf{P}(t) \\ \mathbf{L}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^{\star}(t)\mathbf{R}(t) \\ \mathbf{f}(t) \\ \boldsymbol{\tau}(t) \end{pmatrix}$$

Rigid body dynamics

- Auxiliary variables

$$\mathbf{v}(t) = \frac{\mathbf{P}(t)}{M}, \quad \mathbf{I}(t) = \mathbf{R}(t)\mathbf{I}_0\mathbf{R}(t)^T, \quad \text{and} \quad \boldsymbol{\omega}(t) = \mathbf{I}(t)^{-1}\mathbf{L}(t)$$

- *Variable computed from the state variables.*

Implementation (An example)

- Pre-computation

$$M = \sum_i m_i$$

$$\mathbf{x}_{cm} = \frac{1}{M} \sum_i \mathbf{x}_i m_i$$

$$\mathbf{r}_{0i} = \mathbf{x}_{0i} - \mathbf{x}_{cm}$$

$$\mathbf{I}_0^{-1} = \left(\sum_i^N m_i \mathbf{r}_{0i}^T \mathbf{r}_{0i} \delta - \mathbf{r}_{0i} \mathbf{r}_{0i}^T \right)^{-1}$$

Implementation (An example)

- Initialization

$$\mathbf{x}_{cm}, \mathbf{v}_{cm}, \mathbf{R}, \mathbf{L}$$

$$\mathbf{I}^{-1} = \mathbf{R}\mathbf{I}_0^{-1}\mathbf{R}^T$$

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

Implementation (An example)

- Initialization

$$\mathbf{x}_{cm}, \mathbf{v}_{cm}, \mathbf{R}, \mathbf{L}$$

Initial conditions of your simulation

$$\mathbf{I}^{-1} = \mathbf{R}\mathbf{I}_0^{-1}\mathbf{R}^T$$

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

Implementation (An example)

- Simulation step (1)
 - Per particle forces

$$\boldsymbol{\tau} = \sum_i \mathbf{r}_i \times \mathbf{f}_i$$

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

Implementation (An example)

- Simulation step (2)
 - Rigid body state update

$$\mathbf{x}_{cm} = \mathbf{x}_{cm} + \Delta t \cdot \mathbf{v}_{cm}$$

$$\mathbf{v}_{cm} = \mathbf{v}_{cm} + \Delta t \cdot \mathbf{F} / M$$

$$\mathbf{R} = \mathbf{R} + \Delta t \cdot \omega \mathbf{R}$$

$$\mathbf{L} = \mathbf{L} + \Delta t \cdot \tau$$

$$\mathbf{I}^{-1} = \mathbf{R} \mathbf{I}_0^{-1} \mathbf{R}^T$$

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

Implementation (An example)

- Simulation step (3)
 - Particle update

$$\mathbf{r}_i = \mathbf{R} \cdot \mathbf{r}_{0i}$$

$$\mathbf{x}_i = \mathbf{x}_{cm} + \mathbf{r}_i$$

$$\mathbf{v}_i = \mathbf{v}_{cm} + \omega \mathbf{r}_i$$

Issues with Rotation (matrix)

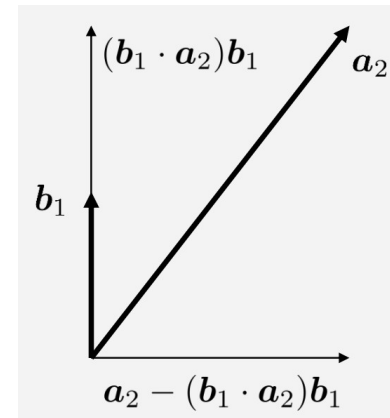
$$\mathbf{R} = \mathbf{R} + \Delta t \cdot \omega^* \mathbf{R}$$

Issues with Rotation (matrix)

$$\mathbf{R} = \mathbf{R} + \Delta t \cdot \boldsymbol{\omega}^* \mathbf{R}$$

- Errors accumulate (numerical drift)
 - The columns of the rotation matrix become no longer orthonormal
- So we must orthonormalize the rotation matrix (occasionally)

$$\begin{aligned}\mathbf{b}_1 &= \mathbf{a}_1 / |\mathbf{a}_1| \\ \mathbf{b}_2 &= \mathbf{a}_2 - (\mathbf{b}_1 \cdot \mathbf{a}_2) \mathbf{b}_1 \\ \mathbf{b}_2 &= \mathbf{b}_2 / |\mathbf{b}_2| \\ \mathbf{b}_3 &= \mathbf{a}_3 - (\mathbf{b}_1 \cdot \mathbf{a}_3) \mathbf{b}_1 - (\mathbf{b}_2 \cdot \mathbf{a}_3) \mathbf{b}_2 \\ \mathbf{b}_3 &= \mathbf{b}_3 / |\mathbf{b}_3|\end{aligned}$$





Getting started ...

- Open the zip file provided to get started.
- You need to
 - know C++ (a bit)
 - Know how to configure CMake
 - ...
 - Glance through the README file for instructions



Extras

- Quaternions for rotations...?
- Handling collisions...?
- More complex objects...?

Additional resources

- Adam W. Bargteil, Tamar Shinar, and Paul G. Kry. 2020. An introduction to physics-based animation. In SIGGRAPH Asia 2020 Courses (SA '20). Association for Computing Machinery, New York, NY, USA, Article 5, 1–57. DOI: <https://doi.org/10.1145/3415263.3419147>
- An Introduction to Physically Based Modeling: Rigid Body Simulation I—Unconstrained Rigid Body Dynamics. David Baraff: <https://www.cs.cmu.edu/~baraff/sigcourse/notesd1.pdf>
- GPU Gems 3: Chapter 29. Real-Time Rigid Body Simulation on GPUs. Takahiro Harada University of Tokyo: <https://developer.nvidia.com/gpugems/gpugems3/part-v-physics-simulation/chapter-29-real-time-rigid-body-simulation-gpus>