PSE Game Physics

Session (3)
Springs, Ropes, Linear Momentum and Rotations

Atanas Atanasov, Philipp Neumann, Martin Schreiber

20.05.2011

Outline

Springs and ropes

- Overview
- Springs
- Ropes

Linear momentum

- Overview
- Object properties
- Conservation laws
- Updated velocities
- Getting more insight, no formulas
- Why to use the inverse of mass?
- Momentum in 3D

Rotations

Outline

Springs and ropes

- Overview
- Springs
- Ropes

Linear momentum

- Overview
- Object properties
- Conservation laws
- Updated velocities
- Getting more insight, no formulas
- Why to use the inverse of mass?
- Momentum in 3D

Rotations

Springs and Ropes

- Springs and ropes are so called **constraints** since they force different objects to behave in a specific way.
- Other constraints are e. g. a **hinge** or a **wheel** attached to a car
- Springs and ropes are frequently used in games:
  - Ropes and springs to drag & drop something
  - Building bridges out of ropes
  - Setting up a “3rd-person“ camera with damped spring
  - Connecting 2 objects
  - ...
- For this worksheet, we assume a few simplifications:
  - The spring is attached to the mass center of the object
  - Spring/Rope collisions are not considered

Springs (1/2)

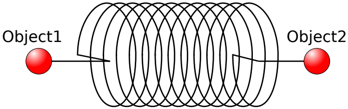


Figure: Spring

- Create a **force acting on both involved objects**
- Force depends on the **distance** of the spring anchors
- Hook's law gives us the force acting on each object:

$$\mathbf{F} = -k \cdot \mathbf{d}$$

- F: Applied force, k: Spring coefficient, d: Distance between both objects

Springs (2/2)

- How to implement this force into our physics engine?
- This force can be implemented via a **soft constraint**:
  - Since we know the time-step size, we are able to **compute the acceleration** exceeded by the spring's force for each object within this time-step (use an explicit euler step).
  - Then the change of acceleration can be simply **added to the acceleration accumulator** during each time-step.
  - The **direction** is given by the line between the **anchor points** of both objects
- Question: Are there problems with large time-steps and large spring coefficients?

Ropes (1/2)

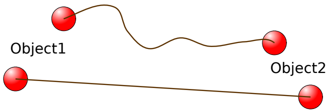


Figure: Top image: rope without exceeding its maximum length, Bottom image: “Colliding rope“

- We only consider ropes **without rope collisions**!!!
- After finishing all worksheets, you’ll be able to simulate this as well :-)

Ropes (2/2)

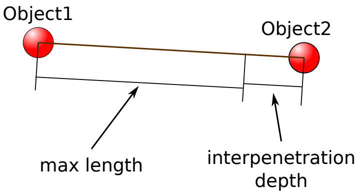


Figure: Colliding details

- Imagine 2 objects with increasing distance.
- Once the anchor points of the ropes at the objects exceed a specific rope length an **impulse acts onto both attached objects**
- This is similar to applying an impulse during collision handling
  $\rightarrow$ **hard constraint**
  $\rightarrow$ Create a simple collision dataset

Outline

Springs and ropes

- Overview
- Springs
- Ropes

Linear momentum

- Overview
- Object properties
- Conservation laws
- Updated velocities
- Getting more insight, no formulas
- Why to use the inverse of mass?
- Momentum in 3D

Rotations

Linear momentum

Once there is a collision, several things have to be done:

- Resolving the interpenetration (finished during last session)
- The velocity of at least one object has to be modified according to its current states

So far, we only consider elastic collisions!

Basics

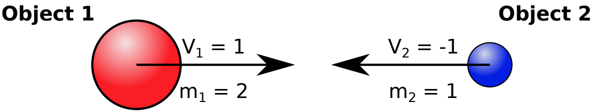


Figure: 2 colliding spheres

- Relevant object properties for linear momentum in 1D:
  - mass of objects: m_1, m_2
  - velocity of object before collision: v_1, v_2
  - velocity of object after collision: v_1', v_2'
- **Closing velocity**:

$$v_c = v_1 - v_2$$

- What's the separating velocity for each object?

We start with a few basic conservation laws:

- **Energy conservation** for elastic collisions:

$$\frac{1}{2}m_1 v_1^2 + \frac{1}{2}m_2 v_2^2 = \frac{1}{2}m_1 {v_1'}^2 + \frac{1}{2}m_2 {v_2'}^2$$

- **Conservation of momentum**:

$$m_1 v_1 + m_2 v_2 = m_1 v_1' + m_2 v_2'$$

- We start by **rewriting the energy conservation law** to

$$m_1(v_1^2 - {v_1'}^2) = -m_2(v_2^2 - {v_2'}^2)$$

and avoid an expression of the velocity with a square exponent:

$$\underbrace{m_1(v_1 - v_1')}_{A}(v_1 + v_1') = -m_2(v_2 - v_2')(v_2 + v_2')$$

- By rewriting the **equation for the conservation of momentum** to

$$m_1(v_1 - v_1') = -m_2(v_2 - v_2')$$

- we can replace (A) by the right side of the upper equation:

$$-m_2(v_2 - v_2')(v_1 + v_1') = -m_2(v_2 - v_2')(v_2 + v_2')$$

- Finally, we get the first result: $v_1 + v_1' = v_2 + v_2'$ or

$$v_1 - v_2 = v_2' - v_1'$$

with $v_1' - v_2' = v_s$ as the **separating velocity** pointing into the opposite direction of the closing velocity (both objects separate)

Updated Velocities

- Using the relation $v_1 - v_2 = v_2' - v_1'$ of the previous slide, we can rearrange it to

$$v_2' = v_1 - v_2 + v_1'$$

- By putting this equation in the momentum equation, we get

$$m_1(v_1 - v_1') = -m_2(v_2 - (v_1 - v_2 + v_1')) = -m_2 v_2 + m_2 v_1 - m_2 v_2 + m_2 v_1'$$

$$\Leftrightarrow m_1 v_1' + m_2 v_1' = 2 m_2 v_2 - m_2 v_1 + m_1 v_1 \overbrace{+ m_2 v_1 - m_2 v_1}^{\text{Extension by 0}}$$

$$\Leftrightarrow (m_1 + m_2) v_1' = 2(m_2 v_2 - m_2 v_1) + (m_1 + m_2) v_1$$

- By dividing the equation by $(m_1 + m_2)$, the velocity of object 1 after collision is given by

$$\Leftrightarrow v_1' = 2 \frac{m_2(v_2 - v_1)}{m_1 + m_2} + v_1$$

- Normally, collisions are not completely elastic. The **coefficient of restitution** C_r is used to represent the respective deviation depending on the closing and separating velocity:

$$C_r = \frac{v_2' - v_1'}{v_1 - v_2} = \frac{v_s}{v_c}$$

- By replacing $(v_1 - v_2)$ with the closing velocity v_c and including the coefficient of restitution, the velocity of object 1 after collision is given by

$$\Leftrightarrow v_1' = 2\frac{m_2}{m_1 + m_2} v_c C_r + v_1$$

- Similarly, the velocity of object 2 after the collision is given by

$$\Leftrightarrow v_2' = -2\frac{m_1}{m_1 + m_2} v_c C_r + v_2$$

Getting more insight, no formulas

- Understanding the formulas is one of the most important things. Therefore we have a look at the most important parts:

$$\underbrace{v_2'}_{\text{new velocity}} = -\underbrace{2}_{2}\underbrace{\frac{m_1}{m_1+m_2}}_{\text{mass fraction}}\underbrace{v_c}_{\text{closing velocity}} C_r + \underbrace{v_2}_{\text{old velocity}}$$

- **New velocity**: Simply the new velocity
- **Impulse**: The change of velocity is computed via an impulse which becomes apparent when separating into **mass** and **velocity**
- **Mass fraction**: The fraction accounts for the change of velocity depending on the fraction m_{object}/m_{total}
- **2**: Imagine an object 2 with mass of 1 hitting another object with infinite mass.
  - Then object 1 is not moving at all but object 2 has to reverse its velocity by adding n-times the closing velocity.
  - To modify the velocity of object 2 to aim into the inversed direction, the closing velocity is **taken twice**.

Using inverse mass...

- Computing v_2' directly, we get problems when the masses of the objects are "close to infinity", i.e. very huge. Assuming, that $m_1 = \infty$, we get

$$\Leftrightarrow v_2' = -2 \underbrace{\frac{\infty}{\infty + m_2}}_{!!!} v_c + v_2$$

- Without further checks, we run into severe problems due to the limited accuracy of the computer numbers.
- Therefore we are working with the inverse of mass:

$$\frac{\frac{1}{m_2^{-1}}}{\left(\frac{1}{m_1^{-1}} + \frac{1}{m_2^{-1}}\right)} = \frac{\frac{1}{m_2^{-1}}(m_1^{-1} m_2^{-1})}{\left(\frac{1}{m_1^{-1}} + \frac{1}{m_2^{-1}}\right)(m_1^{-1} m_2^{-1})} = \frac{m_1^{-1}}{m_1^{-1} + m_2^{-1}}$$

Momentum in 3D

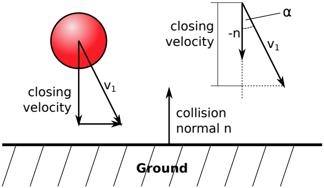


Figure: Collision involving the collision normal

- For 3D, we have to take the collision normal into account.
- Only the velocity component parallel to the collision normal may be involved in computing the new velocity (no friction so far!)
- Thus $\overline{v}$ is computed by using only the fraction of the velocity in direction of the collision normal.
- Using the dot product we can compute this fraction by:

$$\overline{v} = -\vec{n}_c \cdot \vec{v}$$

Outline

Springs and ropes

- Overview
- Springs
- Ropes

Linear momentum

- Overview
- Object properties
- Conservation laws
- Updated velocities
- Getting more insight, no formulas
- Why to use the inverse of mass?
- Momentum in 3D

Rotations

Rotations

Different approaches to describe rotations

- Euler angles
- Quaternions
- ...

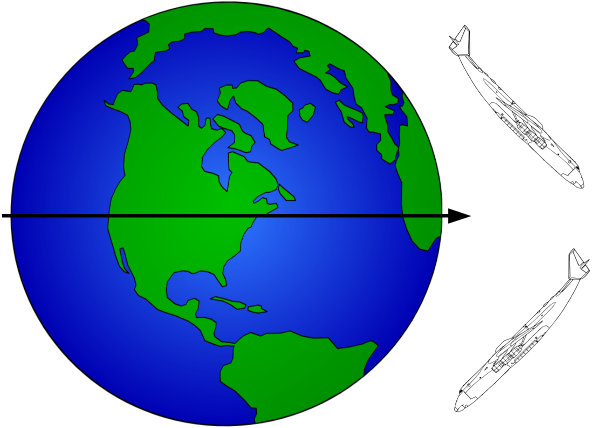
Rotation matrix in 2D:

$$R^{2D}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \quad (1)$$

Example: Rotate vector $v = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ by 90 degrees $(= \frac{\pi}{2})$

$$\rightarrow \qquad v^{new} = R^{2D}(\frac{\pi}{2}) \cdot v = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (2)$$

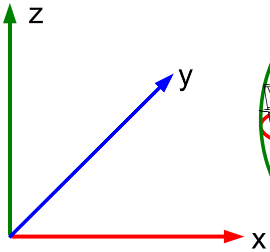
It's not that easy...

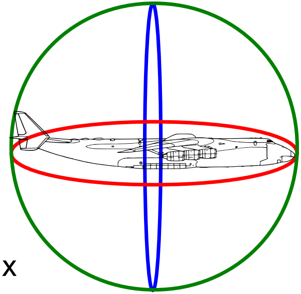


→ Handling rotations sounds straight-forward, but definitely isn't!

Euler Angles

- From 2D rotation: 1 rotation axis → therefore, we use one type of rotation matrix R^{2D}
- 3D: 3 possible rotation axes (x,y,z) → use three rotations to describe any rotation in space
- Euler angles: Describe rotated position by angles between this position and reference position

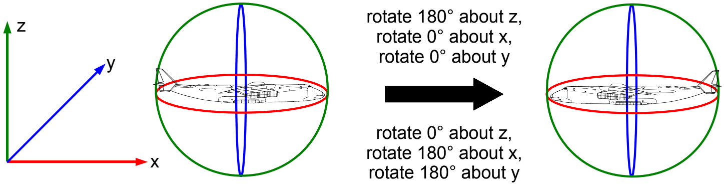




Euler Angles: Problems

- Different conventions/ approaches
- Example: zxy-convention
  1. Rotation about z-axis
  2. Rotation about x-axis
  3. Rotation about y-axis

  $\rightarrow$ 3 degrees of freedom



$\rightarrow$ Rotational description not unique

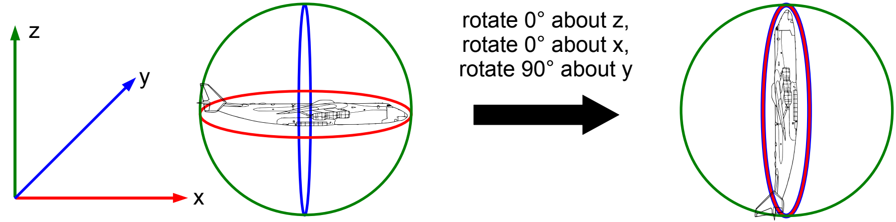
Euler Angles: Problems - Gimbal Lock (1)

Excerpt from the NASA logs and commentary (taken from *Visualizing Quaternions, A.J. Hanson,2006*):

After Stafford's camera failed, he and Cernan had little to do except look at the scenery until time to dump the descent stage. Stafford had the vehicle in the right attitude 10 minutes early. Cernan asked, "You ready?" Then he suddenly exclaimed, "Son of a bitch!" Snoopy seemed to be throwing a fit, lurching wildly about. He later said it was like flying an Immelmann turn in an aircraft, a combination of pitch and yaw. Stafford yelled that they were in gimbal lock–that the engine had swiveled over to a stop and stuck- and they almost were. He called out for Cernan to thrust forward. Stafford then hit the switch to get rid of the descent stage and realized they were 30 degrees off from their previous attitude. The lunar module continued its crazy gyrations across the lunar sky, and a warning light indicated that the inertial measuring unit really was about to reach its limits and go into gimbal lock. Stafford then took over in manual control, made a big pitch maneuver, and started working the attitude control switches. Snoopy finally calmed down.

Euler Angles: Problems - Gimbal Lock (2)

- Gimbal Lock: Loss of one degree of freedom in rotations



Example: x- and z-axis become parallel

Quaternions: Basics

- Describe a rotation θ around an *arbitrary* axis (x, y, z) in space
- Advantage: 3+1 degrees of freedom
- Solves the Gimbal Lock-Problem

Review: Complex numbers $\mathbb{C}$

- Field (German: “Körper”), isomorph to $\mathbb{R}^2$,
  i.e. we can write a complex number $x + iy$ as vector $\begin{pmatrix} x \\ y \end{pmatrix}$
- Addition:
  $$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ y_1 + y_2 \end{pmatrix}$$
- Multiplication:
  $$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \cdot \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 x_2 - y_1 y_2 \\ x_1 y_2 + x_2 y_1 \end{pmatrix}$$
- Rotation in this context:
  $$\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta x - \sin\theta y \\ \sin\theta x + \cos\theta y \end{pmatrix} \Leftrightarrow R^{2D} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Quaternions (1)

- Skew field $\mathbb{H}$ (German: "Schiefkörper")

  (field, but no commutativity in multiplication, i.e. $q_1 \cdot q_2 \neq q_2 \cdot q_1$)
- Addition: Just like regular vector addition (compare complex number addition)
- Multiplication:

$$\begin{pmatrix} w_1 \\ i_1 \\ j_1 \\ k_1 \end{pmatrix} \cdot \begin{pmatrix} w_2 \\ i_2 \\ j_2 \\ k_2 \end{pmatrix} = \begin{pmatrix} w_1 w_2 - (i_1 i_2 + j_1 j_2 + k_1 k_2) \\ w_1 i_2 + w_2 i_1 + j_1 k_2 - k_1 j_2 \\ w_1 j_2 + w_2 j_1 + k_1 i_2 - i_1 k_2 \\ w_1 k_2 + w_2 k_1 + i_1 j_2 - j_1 i_2 \end{pmatrix}$$

Quaternions (2)

- Choose a vector $\vec{n} \in \mathbb{R}^3$ with $\|\vec{n}\| = 1$ (rotation axis) and consider two quaternions q_1, q_2 of the form:

$$q_1 = \begin{pmatrix} w_1 \\ a_1 \cdot \vec{n} \end{pmatrix}, q_2 = \begin{pmatrix} w_2 \\ a_2 \cdot \vec{n} \end{pmatrix}$$

- Multiplication yields:

$$q_1 \cdot q_2 = \begin{pmatrix} w_1 w_2 - a_1 a_2 \\ (w_1 a_2 + w_2 a_1)\vec{n} \end{pmatrix}$$

- Compare this to multiplication of complex numbers!

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \cdot \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 x_2 - y_1 y_2 \\ x_1 y_2 + x_2 y_1 \end{pmatrix}$$

Quaternions (3)

- Quaternions: Generalisation of complex numbers: $w + ix + jy + kz$ with complex units i, j, k
- Rules for imaginary units:

$$\begin{aligned} i^2 = j^2 = k^2 &= -1 \\ ijk &= -1 \end{aligned}$$

Quaternions: Convertions

- Given rotation axis (x, y, z) and angle θ

$$\rightarrow q := \begin{pmatrix} w \\ i \\ j \\ k \end{pmatrix} = \begin{pmatrix} \cos\frac{\theta}{2} \\ x\sin\frac{\theta}{2} \\ y\sin\frac{\theta}{2} \\ z\sin\frac{\theta}{2} \end{pmatrix} \in \mathbb{R} \times \mathbb{R}^3$$

- Convert quaternion back to general rotation matrix R via

$$R = \begin{pmatrix} 1-2(j^2+k^2) & 2(ij+kw) & 2(ik-jw) \\ 2(ij-kw) & 1-2(i^2+k^2) & 2(jk+iw) \\ 2(ik+jw) & 2(jk-iw) & 1-2(i^2+j^2) \end{pmatrix}$$