Chapter 7: Linear maps and transformation matrices

K. Jünemann
Department Informations- und Elektrotechnik
HAW Hamburg

Content

Linear maps and matrices

In the following: think of vectors as columns!

A map $f : \mathbb{R}^N \to \mathbb{R}^N$ is called *linear* iff
- $f(\alpha\vec{x}) = \alpha f(\vec{x})$
- $f(\vec{x} + \vec{y}) = f(\vec{x}) + f(\vec{y})$

for any two vector $\vec{x}, \vec{y} \in \mathbb{R}^N$ and any scalar α.

Remarks:

- Each linear map satisfies $f(\vec{0}) = \vec{0}$.
  Proof: $\vec{0} = 0 \cdot \vec{x} \implies f(\vec{0}) = f(0 \cdot \vec{x}) = 0 \cdot f(\vec{x}) = \vec{0}$ for an arbitrary vector $\vec{x}$.
- A linear map is determined by its action on basis vectors, e.g. in $\mathbb{R}^2$:

$$f(\vec{x}) = f(x_1\vec{e}_1 + x_2\vec{e}_2) = x_1 f(\vec{e}_1) + x_2 f(\vec{e}_2)$$

Linear maps and matrices

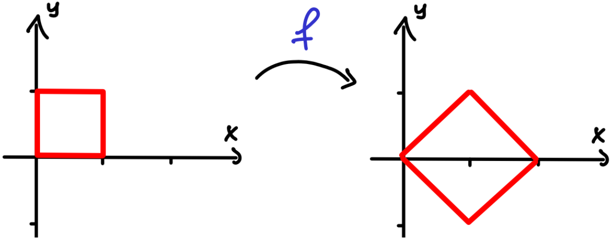
Example: Consider a linear map f in $\mathbb{R}^2$, defined by

$$f(\vec{e}_1) = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \qquad f(\vec{e}_2) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

What ist $f(\vec{x})$ with $\vec{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \vec{e}_1 + \vec{e}_2$?

$$f(\vec{x}) = f(\vec{e}_1 + \vec{e}_2) = f(\vec{e}_1) + f(\vec{e}_2) = \begin{pmatrix} 1 \\ -1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

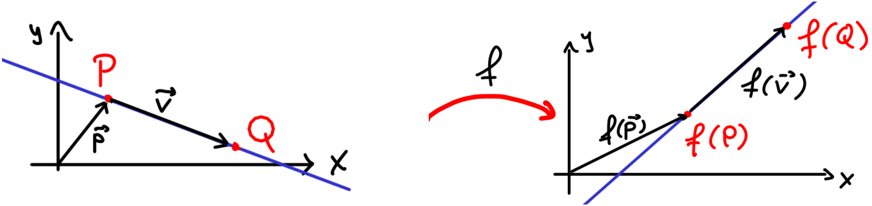
Visualize by action on unit square: rotation and scaling



Linear maps and matrices

Consider straight line through points P and Q:

$$\vec{p} + \alpha\vec{v} \qquad \text{with } \vec{v} := \overline{PQ} \text{ and any } \alpha \in \mathbb{R}$$



Apply linear map f to straight line:

$$f(\vec{p} + \alpha\vec{v}) = f(\vec{p}) + \alpha f(\vec{v})$$

Result: straight line through points $f(P)$ and $f(Q)$!

$\boxed{\text{Linear maps map straight lines to straight lines.}}$

Linear maps and matrices

Matrices and linear maps are the same because

1. Every $N \times N$ matrix $\mathbf{A}$ defines a linear map $f : \mathbb{R}^N \to \mathbb{R}^N$ by the rule
$$f(\vec{x}) := \mathbf{A} \cdot \vec{x}.$$
2. For any given linear map $f : \mathbb{R}^N \to \mathbb{R}^N$ there is a matrix $\mathbf{A}_f$ such that $f(\vec{x}) = \mathbf{A}_f \cdot \vec{x}$.

The first statement is obvious but why is the second true?

Linear maps and matrices

Let $\vec{b}_1, \ldots, \vec{b}_N$ be a basis $\implies$ any $\vec{x}$ has the form

$$\vec{x} = x_1\vec{b}_1 + \ldots + x_N\vec{b}_N$$

Apply some given linear map f:

$$\begin{aligned}
f(\vec{x}) &= f(x_1\vec{b}_1 + \ldots + x_N\vec{b}_N) \\
&= x_1 f(\vec{b}_1) + \ldots + x_N f(\vec{b}_N) \\
&= \underbrace{\begin{pmatrix} | & \ldots & | \\ f(\vec{b}_1) & \ldots & f(\vec{b}_N) \\ | & \ldots & | \end{pmatrix}}_{\mathbf{A}_f} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} \\
&= \mathbf{A}_f \cdot \vec{x}
\end{aligned}$$

Linear maps and matrices

Any linear map $f : \mathbb{R}^N \to \mathbb{R}^N$ has the form

$$f(\vec{x}) = \mathbf{A}_f \cdot \vec{x}$$

where the columns of the $N \times N$ matrix $\mathbf{A}_f$ consists of f applied to the basis vectors.

Remarks:

- $\mathbf{A}_f$ is called *representation matrix* of the map f.
- $\mathbf{A}_f$ depends on choice of basis.
- The identity map $\mathrm{id}(\vec{x}) = \vec{x}$ is represented by unit matrix $\mathbf{E}$.
- The inverse matrix $\mathbf{A}_f^{-1}$ is the representation matrix of the inverse map f^{-1}.

Linear maps and matrices

Example: Again the linear map f in $\mathbb{R}^2$, defined by

$$f(\vec{e}_1) = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \qquad f(\vec{e}_2) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Representation matrix:

$$\mathbf{A}_f = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$

Check:

$$\mathbf{A}_f \cdot \vec{e}_1 = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$\mathbf{A}_f \cdot \vec{e}_2 = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

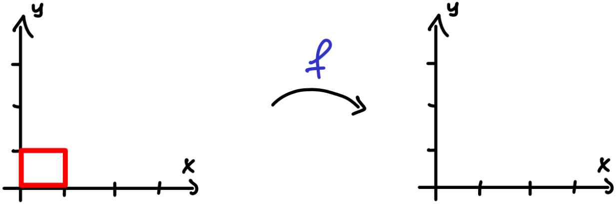
$$\mathbf{A}_f \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

Exercise 1

For some linear map in $\mathbb{R}^2$ we know

$$f(\vec{e}_1) = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \qquad f(\vec{e}_2) = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

- Find the matrix $\mathbf{A}_f$ of the linear map.
- What is $f(\vec{x})$ with $\vec{x} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$?
- Draw the image of the unit square under f?

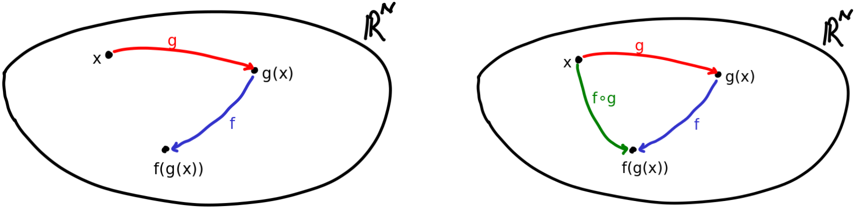


Linear maps and matrices

Consider two linear maps f and g with representation matrices $\mathbf{A}_f$ and $\mathbf{A}_g$. What is the meaning of the matrix product $\mathbf{A}_f \cdot \mathbf{A}_g$?

$$\mathbf{A}_f \cdot \underbrace{\mathbf{A}_g \cdot \vec{x}}_{g(\vec{x})} = f\left(g(\vec{x})\right) =: f \circ g(\vec{x})$$

$\boxed{\text{The product } \mathbf{A}_f \cdot \mathbf{A}_g \text{ of two representation matrices is the representation matrix of the composition map } f \circ g.}$



Linear maps and matrices

Remarks on the composition of linear maps:

- Read composition of maps $f \circ g$ from right to left!
  $\mathbf{A}_f \cdot \mathbf{A}_g$ means:
  - *First* apply the map g with representation matrix $\mathbf{A}_g$.
  - *Then* apply the map f with representation matrix $\mathbf{A}_f$.
- The order of maps matters:
  - $f \circ g \neq g \circ f$
  - $\mathbf{A}_f \cdot \mathbf{A}_g \neq \mathbf{A}_g \cdot \mathbf{A}_f$
- Composition of many maps represented by single matrix:

$$f_1 \circ f_2 \circ \ldots \circ f_N(\vec{x}) = \underbrace{\mathbf{A}_{f_1} \cdot \mathbf{A}_{f_2} \cdot \ldots \cdot \mathbf{A}_{f_N}}_{\mathbf{A}} \cdot \vec{x} = \mathbf{A} \cdot \vec{x}$$

  allows for efficient implementation of $f_1 \circ f_2 \circ \ldots \circ f_N$ applied to a large number of vectors $\vec{x}$!

Linear maps in 2D

General form of a linear map in $\mathbb{R}^2$:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \to \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

contains 4 paramters.

Every linear map in $\mathbb{R}^2$ is the composition of a

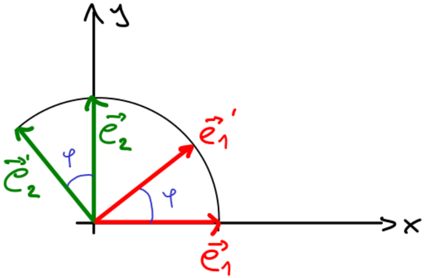
- rotation
- scale transform
- shear transform

These are called *elementary* transforms.

Linear maps in 2D

Rotations in $\mathbb{R}^2$:

- Rotation angle: φ defined modulo 2π (1 parameter)
- Rotation matrix $\mathbf{R}_\varphi$: columns are rotated basis vectors $\vec{e}'_1$ and $\vec{e}'_2$



$$\vec{e}'_1 = \begin{pmatrix} \cos(\varphi) \\ \sin(\varphi) \end{pmatrix} , \qquad \vec{e}'_2 = \begin{pmatrix} -\sin(\varphi) \\ \cos(\varphi) \end{pmatrix}$$

Linear maps in 2D

A rotation matrix in 2D has the form

$$\mathbf{R}_\varphi = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix}$$

where φ is the angle of rotation.

Check that $\mathbf{R}_\varphi$ satisfies the conditions

1. $\det(\mathbf{R}_\varphi) = 1$
2. $\mathbf{R}_\varphi^T = \mathbf{R}_\varphi^{-1}$

- Any rotation matrix in any dimension satisfies these two conditions.
- A matrix satisfying 2. is called *orthogonal* matrix.
  Orthogonal matrices preserve length and angles of transformed vectors.

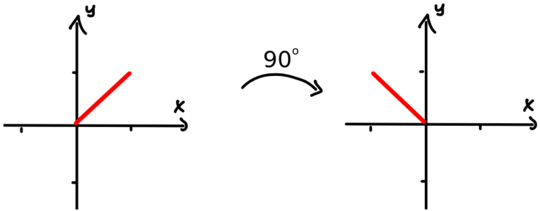
Linear maps in 2D

Example: rotation by 90°.

$$\mathbf{R}_{90^\circ} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

applied to $(1, 1)$:

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} \to \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$



Exercise 2

Consider a rotation in $\mathbb{R}^2$ by 45°.

1. Write down the rotation matrix.
2. What is the result of rotating the vector $(2, 1)$ by 45°?
3. Draw the image of the square with vertices $(1, 0), (2, 0), (2, 1), (1, 1)$ under a rotation of 45°.

Hint: $\cos(\pi/4) = \sin(\pi/4) = 1/\sqrt{2}$

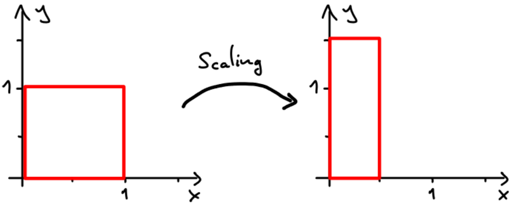
Linear maps in 2D

Scale transforms in $\mathbb{R}^2$ (2 parameters): $\begin{pmatrix} s_1 & 0 \\ 0 & s_2 \end{pmatrix}$

- scale factor > 1: expansion
- scale factor < 1: contraction

Action on a vector:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \to \begin{pmatrix} s_1 & 0 \\ 0 & s_2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_1 \end{pmatrix} = \begin{pmatrix} s_1 x_1 \\ s_2 x_2 \end{pmatrix}$$



Scaling with matrix $\frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}$

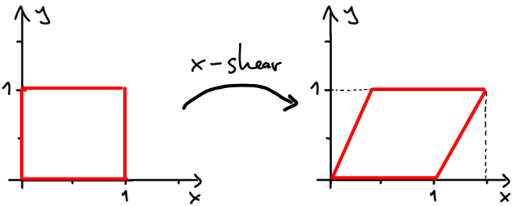
Linear maps in 2D

Shear transforms in $\mathbb{R}^2$ (2 parameters):

$$\begin{pmatrix} 1 & \sigma_1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ \sigma_2 & 1 \end{pmatrix}$$

Action on a vector:

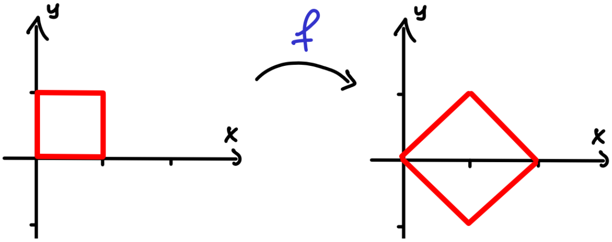
$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \to \begin{pmatrix} 1 & \sigma_1 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1 + \sigma_1 x_2 \\ x_2 \end{pmatrix}$$



Shear transform with matrix $\begin{pmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{pmatrix}$

Linear maps in 2D

The example from pages 4 and 9 as composition of two maps:



$$\mathbf{A}_f = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{pmatrix}}_{\text{Scaling by } \sqrt{2}} \cdot \underbrace{\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}}_{\text{Rot. by } -45^\circ}$$

f is the composition of a rotation and a scaling

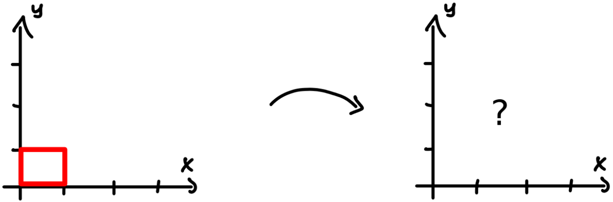
Exercise 3

Write the map of exercise 1 on slide 10 as a composition of two elementary transforms.

Exercise 4

Let **R** be a 2D-rotation by 45° and **S** be a scaling by $\frac{1}{2}$ into the y-direction.

1. What happens to the unit square when it is first rotated by **R** and then scaled by **S**?
2. What happens to the unit square when it is first scaled by **S** and then rotated by **R**?
3. What happens to the unit square when it is first rotated by **R**, then scaled by **S** and then rotated back by $\mathbf{R}^{-1}$?



Rotations in 3D

- Shear and scale matrices are similar in 2D and 3D.
- Rotations in 3D depend on 3 parameters
  $\Longrightarrow$ more complicated in 3D!
- The order of rotations matter!

  $$\text{In general: } \mathbf{R}_1 \cdot \mathbf{R}_2 \neq \mathbf{R}_2 \cdot \mathbf{R}_1$$

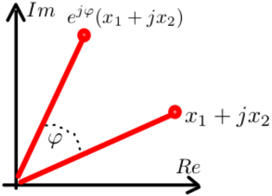
  for two different rotation matrices.
- Several approaches to 3D rotations exist:
  - Quaternions (generalized complex numbers) used internally in `three.js`
  - Euler (or Tait-Bryan) angles
  - Axis-angle matrices

Rotations in 3D

A word about quaternions:

- 2D rotations can be implemented using complex numbers:

$$
\begin{aligned}
&(x_1 + ix_2) \cdot e^{i\varphi} \\
&= (x_1 + ix_2) \cdot (\cos(\varphi) + i\sin(\varphi)) \\
&= (x_1\cos(\varphi) - x_2\sin(\varphi)) \\
&+ i(x_1\sin(\varphi) + x_2\cos(\varphi))
\end{aligned}
$$



- Extend this to 3D by adding more imaginary units:
  - Just adding one more imaginary unit j doesn't work.
  - Adding two units works! Definition of a quaternion q:

$$q = x_1 + ix_2 + jx_3 + kx_4 \qquad \text{with} \qquad i^2 = j^2 = k^2 = -1$$

and

$$ij = -ji = k\,,\;\; jk = -kj = i\,,\;\; ki = -ik = j\,,$$

Rotations in 3D: Euler angles

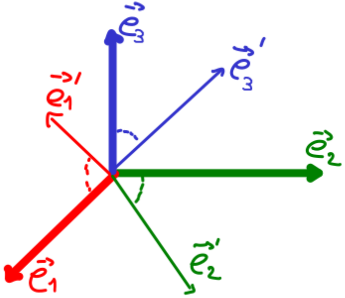
- More precisely: Tait-Bryan angles
- Euler observed that any rotation can be thought of as *succession of three elementary rotations*:

$$\mathbf{R}_{z''}(\gamma) \cdot \mathbf{R}_{y'}(\beta) \cdot \mathbf{R}_{x}(\alpha)$$

- α, β, γ: angles of elementary rotations.
- Axes of elementary rotations define an order, here XYZ.

Rotations in 3D

Digression: Fixed and rotated frames



Given

- a basis $\vec{e}_1$, $\vec{e}_2$, $\vec{e}_3$ defining a fixed frame,
- a rotation matrix $\mathbf{M}$,

one defines a rotated (or primed) frame with basis $\vec{e}_{1'}$, $\vec{e}_{1'}$, $\vec{e}_{3'}$:

Relation between bases:

$$\vec{e}_{1'} = \mathbf{M} \cdot \vec{e}_1 \ , \qquad \vec{e}_{2'} = \mathbf{M} \cdot \vec{e}_2 \ , \qquad \vec{e}_{3'} = \mathbf{M} \cdot \vec{e}_3$$

Rotations in 3D: Euler angles

Axes of elementary rotations for order *XYZ*

- First rotation $\mathbf{R}_x(\alpha)$ around x-axis of fixed coordinate system.
- Second rotation $\mathbf{R}_{y'}(\beta)$ around y'-axis of $\mathbf{R}_x(\alpha)$- rotated coordinate system.
- Third rotation $\mathbf{R}_{z''}(\gamma)$ around z''-axis of $\mathbf{R}_{y'}(\beta) \cdot \mathbf{R}_x(\alpha)$-rotated coordinate system.

Remarks:

- 6 possible orders: *XYZ*, *XZY*, *YXZ*, *YZX*, *ZXY*, *ZYX*.
- *XYZ* is the default in `three.js`.
- Euler orders are read from left to right.

Rotations in 3D: Euler angles in `three.js`

`Object3D.rotation` is `Euler` object:

```
THREE.Euler = function ( x, y, z, order ) {
  this._x = x || 0; // angle around x-axis
  this._y = y || 0; // angle around y-axis
  this._z = z || 0; // angle around z-axis
  this._order = order
  || THREE.Euler.DefaultOrder; // = XYZ
};
```

Code to test example on next slide:

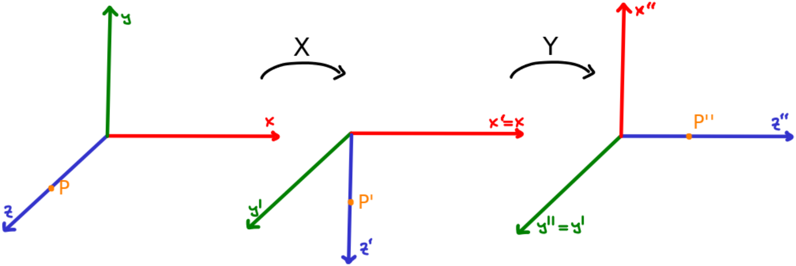
```
const a = Math.PI/2;
const eu = new THREE.Euler(a,a,0,"XYZ");
const m = new THREE.Matrix4();
m.makeRotationFromEuler(eu);
const v = new THREE.Vector3(0,0,1).applyMatrix4(m);
```

Rotations in 3D: Euler angles

Example: consider an Euler rotation with

- x-angle $\alpha = 90^\circ$
- y-angle $\beta = 90^\circ$
- z-angle $\gamma = 0^\circ$

The result of applying this Euler rotation to $(0, 0, 1)$ with order *XYZ* is $(1, 0, 0)$ (in the original coordinate system).

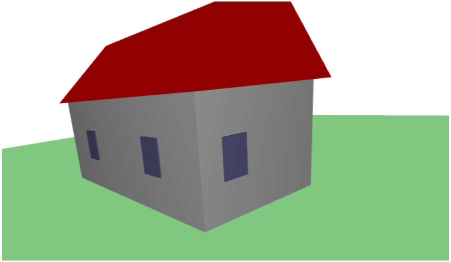


Exercise 5

1. Do the example of the with order YXZ.
2. Check your result with `three.js`.

Exercise 6

Add windows to the house:



- Implement windows as thin black box geometries.
- Position them correctly on the walls.

Hint: Add axes showing the *local* frame of any object

```
// len: length of axes
obj.add(new THREE.AxesHelper(len));
```

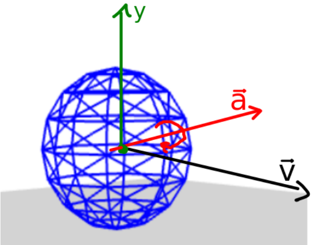
Rotations in 3D: Euler angles

Features of Euler angles:

- Given angles and order, rotation is hard to visualize in the fixed frame.
- Easier to visualize in body frame.
- Loss of degree of freedom if two axes of rotation happen to coincide, e.g. $\beta = \pi/2$ (gimbal lock).

Rotations in 3D: Axis-angle representation

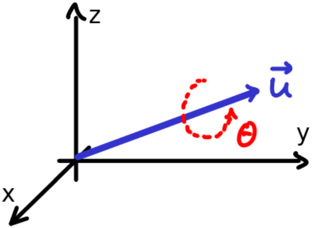
How to implement a rolling ball?



- Assume ball moves in x-z-plane
- Speed of ball: $\vec{v} = (v_x, 0, v_z)$ arbitrary
- Radius of ball: R

- Axis of rotation: $\vec{a} = \dfrac{\vec{e}_y \times \vec{v}}{|\vec{e}_y \times \vec{v}|}$
  $\vec{e}_y$: normal to rolling plane
- Angular velocity: $\omega = \dfrac{|\vec{v}|}{R}$
- Task: Rotate ball around axis $\vec{n}$ by angle $\theta = \omega \cdot t$:

Rotations in 3D: Axis-angle representation



A rotation is specified by

- its axis of rotation $\vec{u}$ (unit vector, 2 parameters)
- the rotation angle θ (1 parameter)
- Notation: $\mathbf{R}_{\vec{u},\theta}$

Properties of $\mathbf{R}_{\vec{u},\theta}$:

- The axis remains unchanged by the rotation:

$$\mathbf{R}_{\vec{u},\theta} \cdot \vec{u} = \vec{u}$$

- The angle θ is given by

$$\mathrm{tr}\left(\mathbf{R}_{\vec{u},\theta}\right) = 1 + 2\cos(\theta)$$

tr($\mathbf{M}$): trace of $\mathbf{M}$ = sum of its diagonal elements.

Rotations in 3D: Axis-angle representation

Big question: Given $\vec{u}$ and θ, what does the rotation matrix $\mathbf{R}_{\vec{u},\theta}$ look like?

Easy cases (with $c := \cos(\theta),\ s := \sin(\theta)$):

$$\begin{aligned}
\mathbf{R}_{\vec{e}_x,\theta} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{pmatrix} \\
\mathbf{R}_{\vec{e}_y,\theta} &= \begin{pmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s & 0 & c \end{pmatrix} \\
\mathbf{R}_{\vec{e}_z,\theta} &= \begin{pmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{pmatrix}
\end{aligned}$$

Rotations in 3D: Axis-angle representation

For a given rotation axis $\vec{u} = (u_1, u_2, u_3)$ and rotation angle θ the axis-angle rotation matrix is

$$\mathbf{R}_{\vec{u},\theta} = \begin{pmatrix} (1-c)u_1^2 + c & (1-c)u_2 u_1 - s u_3 & (1-c)u_3 u_1 + s u_2 \\ (1-c)u_1 u_2 + s u_3 & (1-c)u_2^2 + c & (1-c)u_3 u_2 - s u_1 \\ (1-c)u_1 u_3 - s u_2 & (1-c)u_2 u_3 + s u_1 & (1-c)u_3^2 + c \end{pmatrix}$$

with $c := \cos(\theta)$ and $s := \sin(\theta)$.

- Note: $\vec{u}$ must be a unit vector!
- Properties:
  - $\mathbf{R}_{\vec{u},\theta}$ is orthogonal matrix with unit determinant, i.e.

$$\det(\mathbf{R}_{\vec{u},\theta}) = 1 \qquad \text{and} \qquad \mathbf{R}_{\vec{u},\theta}^T = \mathbf{R}_{\vec{u},\theta}^{-1} = \mathbf{R}_{\vec{u},-\theta}$$

  - Consecutive rotations with *same* axis:

$$\mathbf{R}_{\vec{u},\theta} \cdot \mathbf{R}_{\vec{u},\varphi} = \mathbf{R}_{\vec{u},\theta+\varphi}$$

Exercise 7

1. Check that $\vec{u} = \vec{e}_z$ reproduces the correct result.
2. Work out the rotation matrix for a rotation by 60° around an axis pointing into direction $(1, 1, 1)$
   Hint: $\cos(\pi/3) = 1/2$, $\sin(\pi/3) = \sqrt{3}/2$
3. What do the vectors $(1, -1, 0)$ and $(2, 2, 2)$ rotate into?
4. Read the `THREE.Matrix4` documentation (look for `makeRotationAxis`) to create this rotation matrix in the browser. Use this to check the results of parts 2. and 3.
   Hint: Use the provided `printMat` function (see `lib` directory).

Exercise 8

Use `three.js` to verify the relation

$$\mathbf{R}_{\vec{u},\theta} \cdot \mathbf{R}_{\vec{u},\varphi} = \mathbf{R}_{\vec{u},\theta+\varphi}$$

1. Create a random unit vector $\vec{u}$.
2. Create two random angles θ and φ.
3. Calculate both sides of the equation and compare by inspection (use `printMat`).
4. Verify that in this case the rotation matrices commute:

$$\mathbf{R}_{\vec{u},\theta} \cdot \mathbf{R}_{\vec{u},\varphi} = \mathbf{R}_{\vec{u},\varphi} \cdot \mathbf{R}_{\vec{u},\theta}$$