Intuitive 3d Vector Math for Simulation

Matthias Müller, Ten Minute Physics

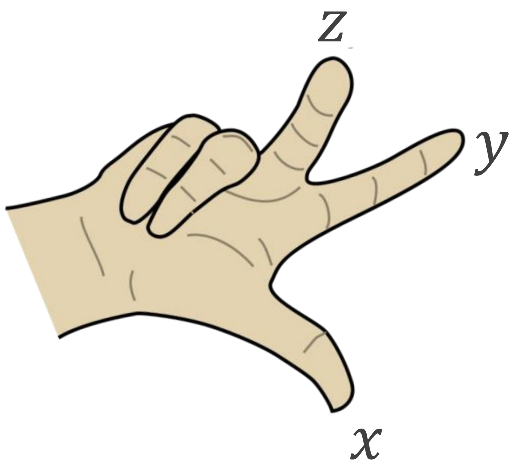
matthiasmueller.info/tenMinutePhysics

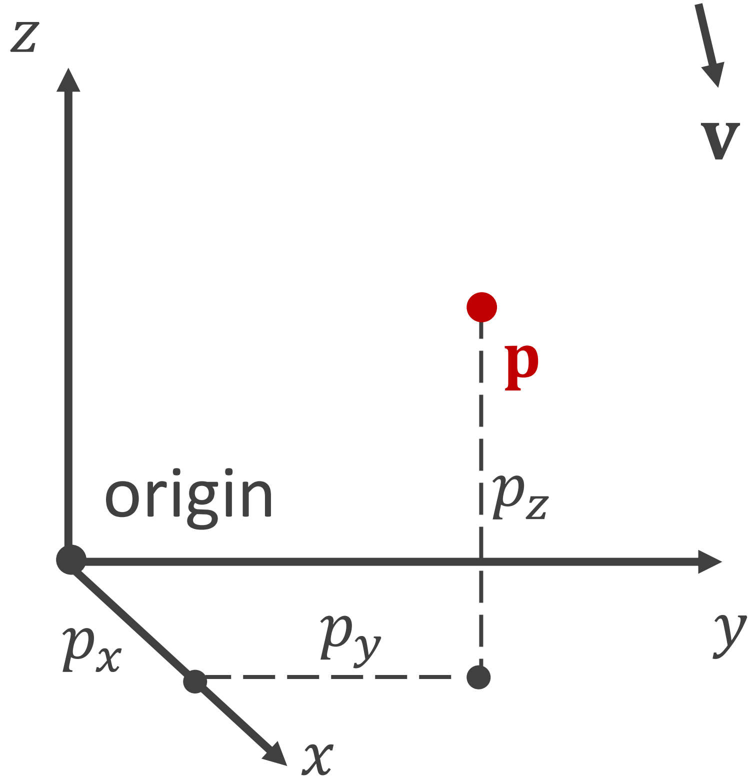
A 3d Vector

bold face

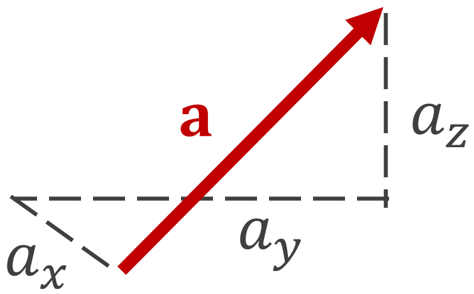
$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

components
coordinates
(regular numbers)



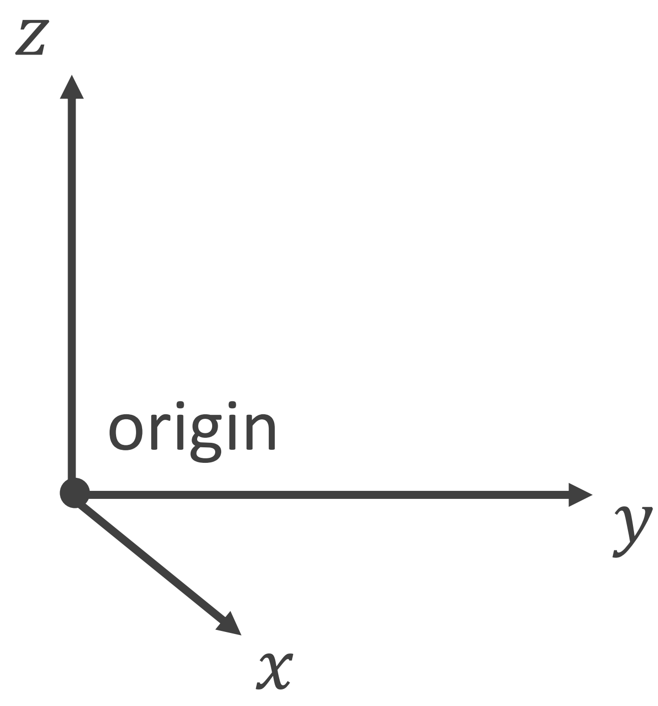


location
e.g. particle position

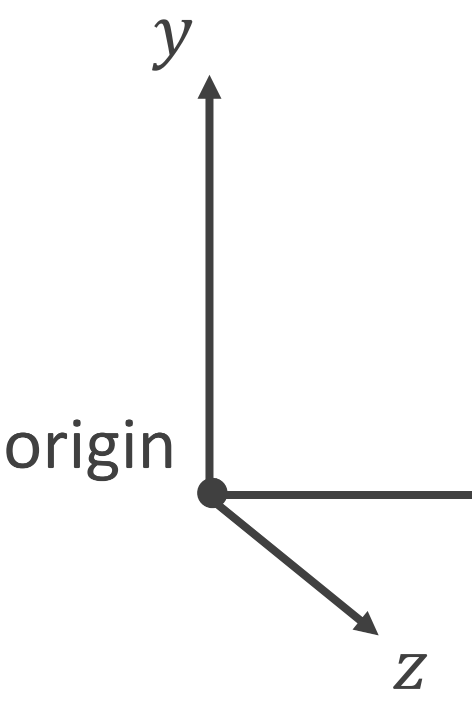


arrow
e.g. velocity

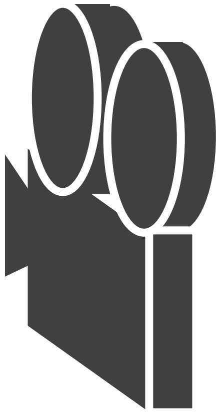
Graphics Coordinate System

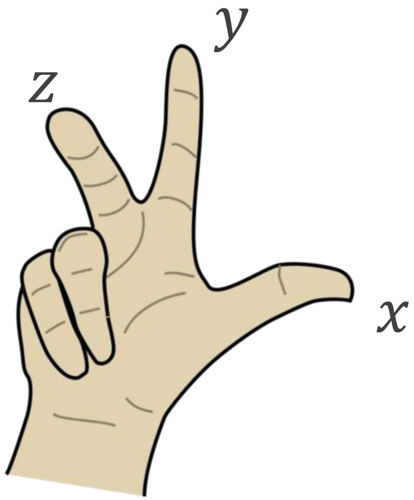


mathematical



graphics

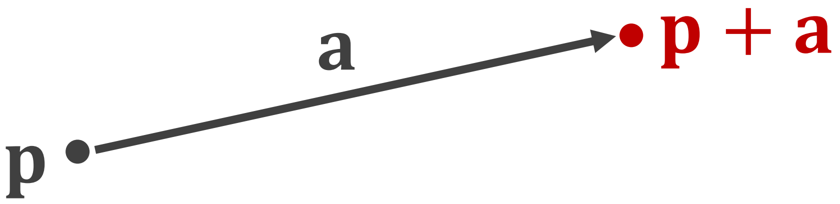




Vector Operations

Addition

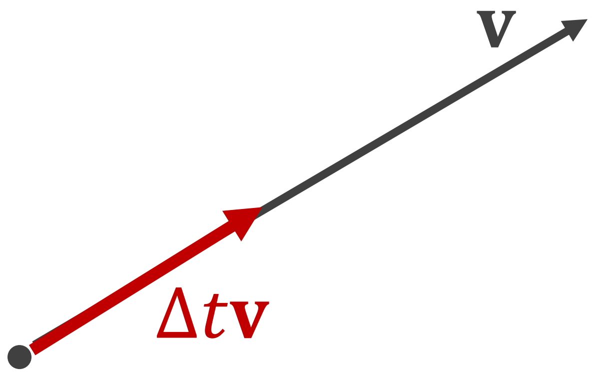
Move forward in time



$$\mathbf{p} + \mathbf{a} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} p_x + a_x \\ p_y + a_y \\ p_z + a_z \end{bmatrix}$$

Scaling

Going from velocity to position update

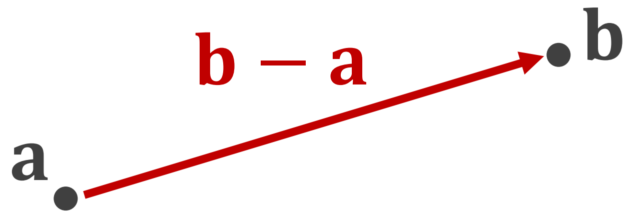


$$\Delta t\ \mathbf{v} = dt \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} \Delta t\ v_x \\ \Delta t\ v_y \\ \Delta t\ v_z \end{bmatrix}$$

direction preserved

Subtraction

Compute the vector from $\mathbf{a}$ to $\mathbf{b}$

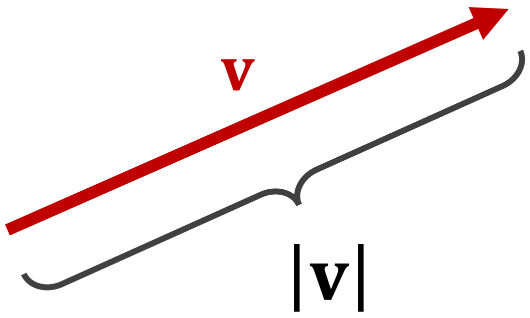


$$\mathbf{b} - \mathbf{a} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} - \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} b_x - a_x \\ b_y - a_y \\ b_z - a_z \end{bmatrix}$$

...not $\mathbf{a} - \mathbf{b}$!

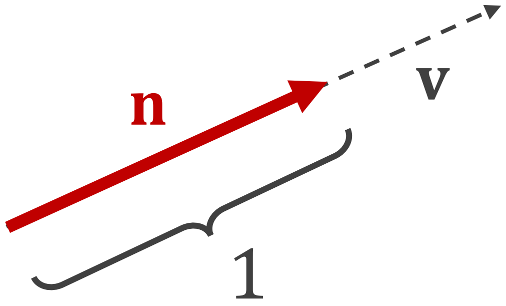
Vector Length and Normalization

vector length



$$|\mathbf{v}| = \sqrt{{v_x}^2 + {v_y}^2 + {v_z}^2}$$

normalized vector (unit vector)



$$\mathbf{n} = \frac{1}{|\mathbf{v}|}\mathbf{v}$$

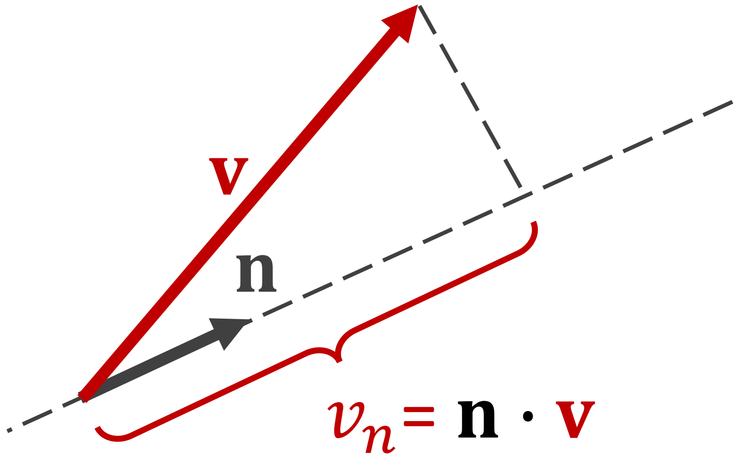
The Dot Product

$$\mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y + a_z b_z$$

Yields a scalar (simple number)

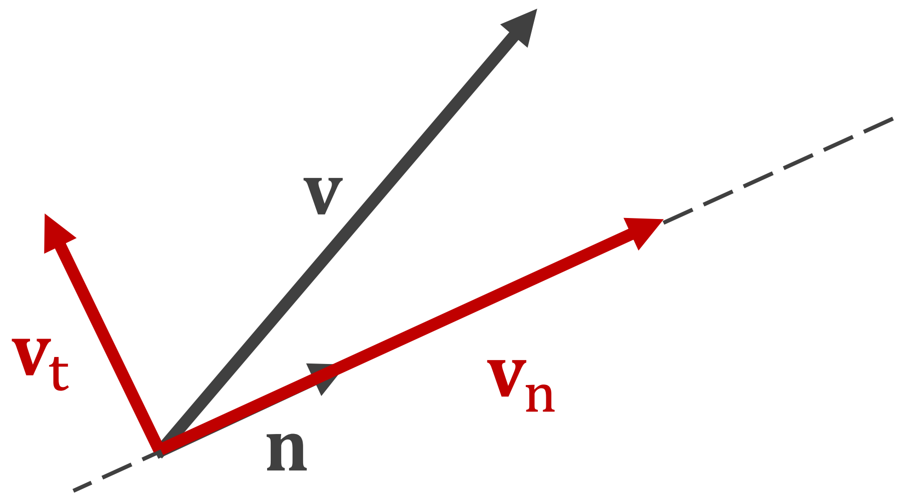
Super simple. Very useful!

Vector length along a given direction $\mathbf{n}$:



$$\mathbf{a} \cdot \mathbf{b} = 0 \quad \leftrightarrow \quad \mathbf{a} \perp \mathbf{b}$$

General Vector Components



$$\mathbf{v}_n = (\mathbf{n} \cdot \mathbf{v})\mathbf{n}$$

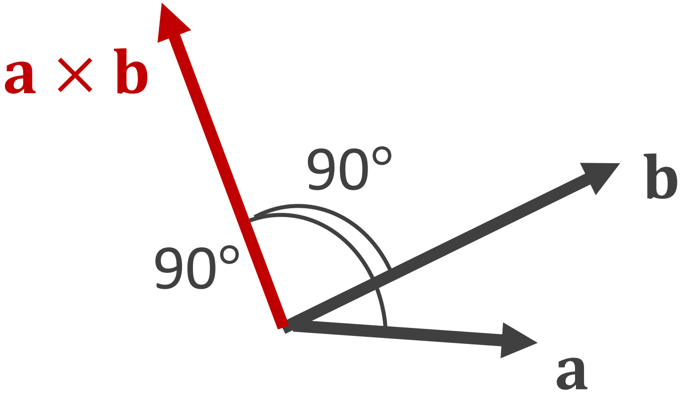
$$\mathbf{v}_t = \mathbf{v} - \mathbf{v}_n$$

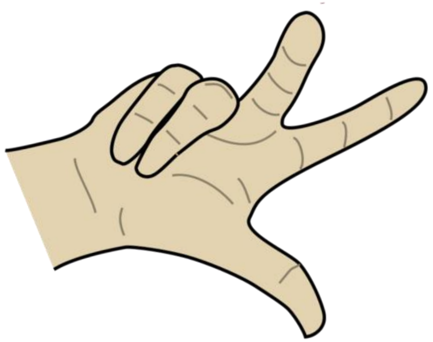
Split into restitution and friction effects

The Cross Product

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \times \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \begin{bmatrix} a_y b_z - b_y\, a_z \\ a_z b_x - b_z\, a_x \\ a_y b_z - b_y\, a_z \end{bmatrix}$$

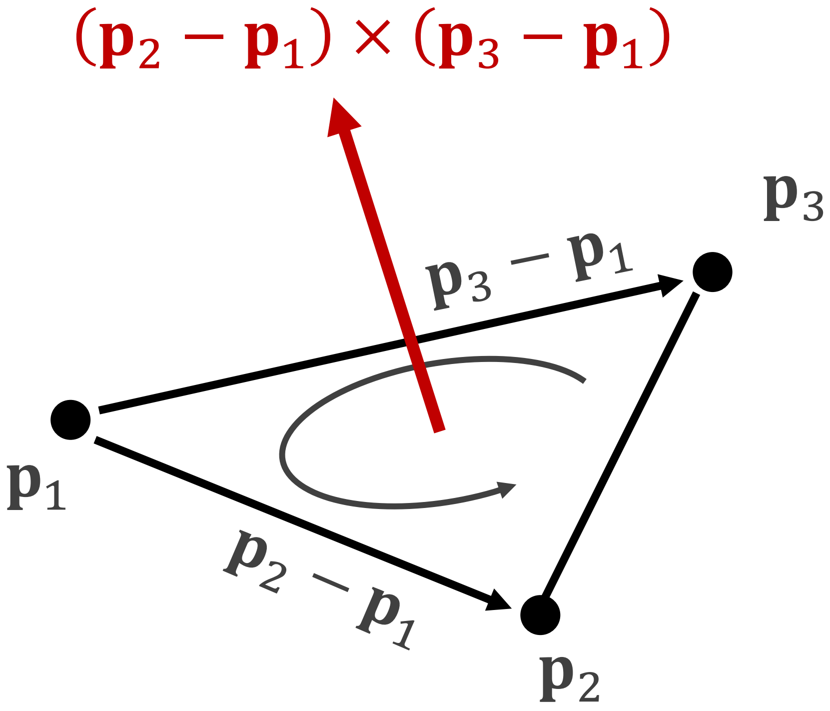
Yields a vector

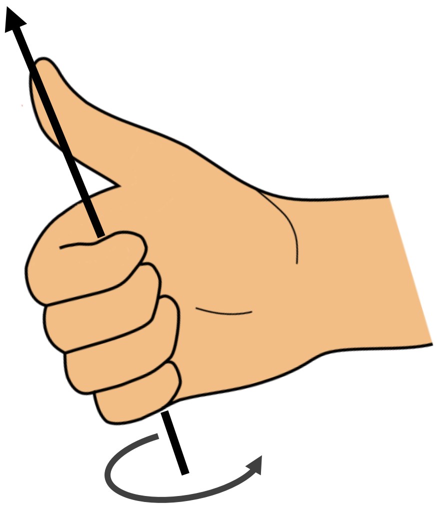




Create a vector that is perpendicular to two vectors

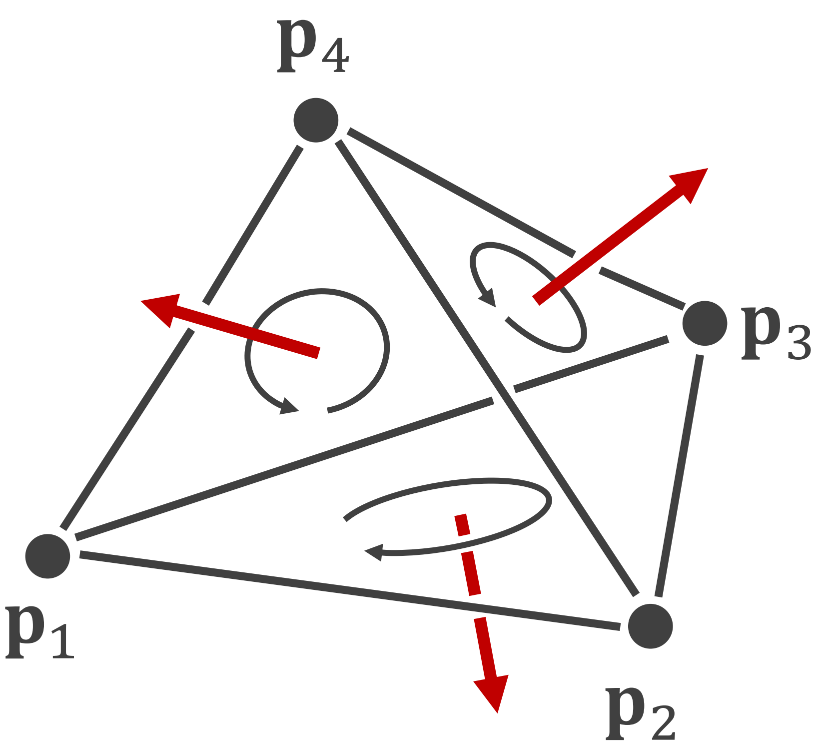
Normal of a triangle





$$\mathbf{n} = \frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)}{|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)|}$$

Orientation of Surface Meshes



Face definition:

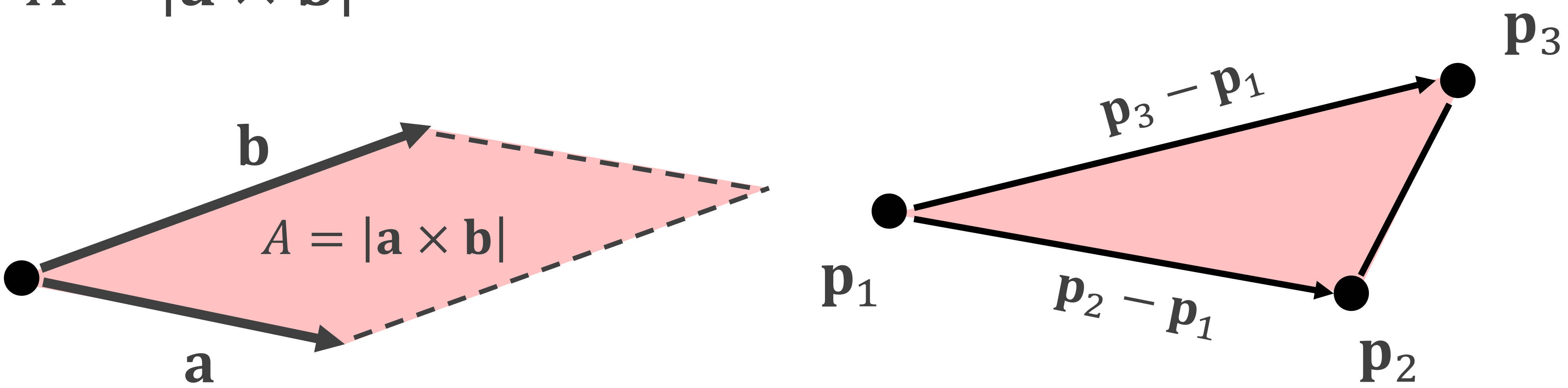
(3,2,1), (1,2,4), (2,3,4), (3,1,4)

Also OK:

(1,3,2), (2,4,1), (2,3,4), (4,3,1)

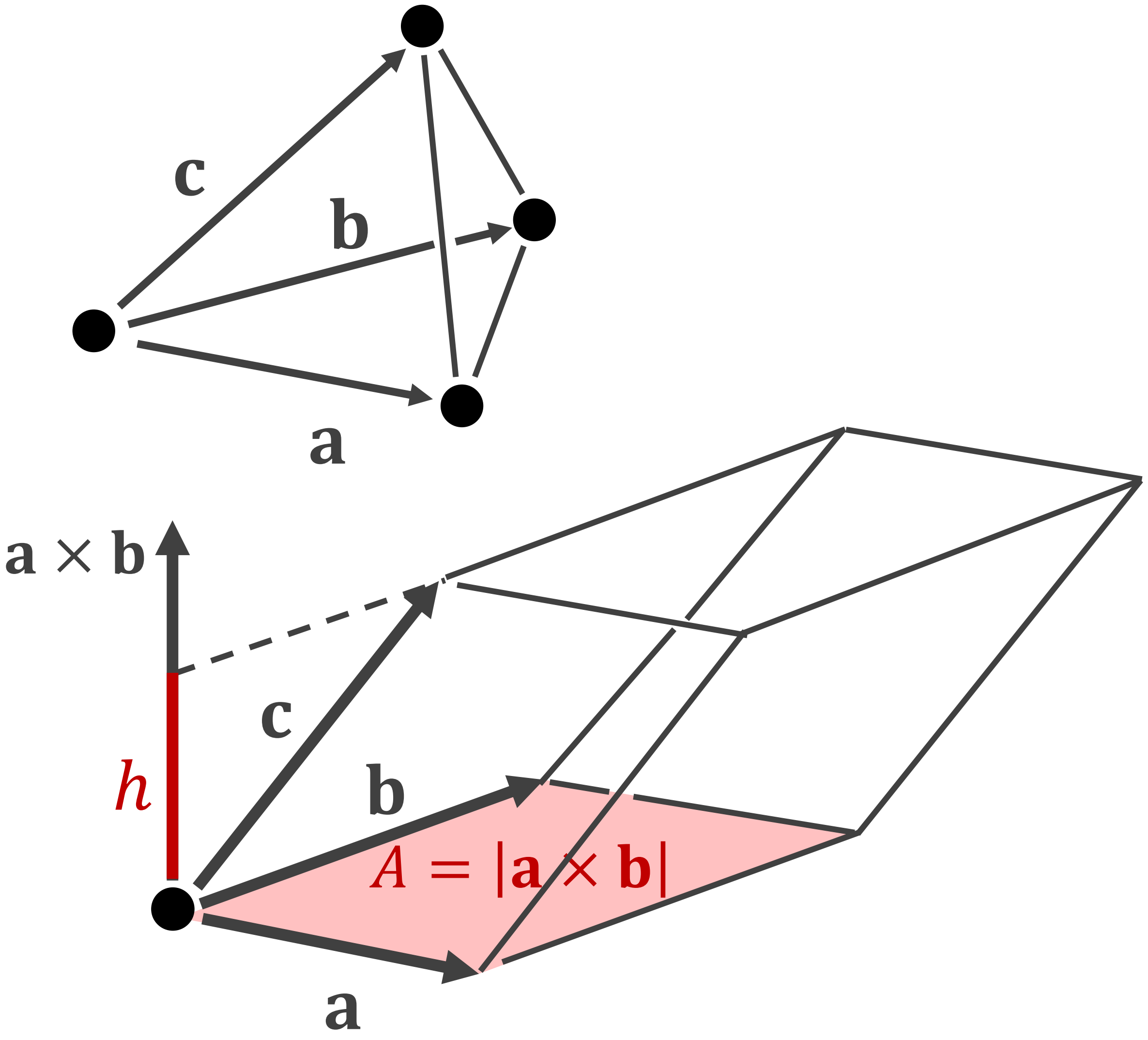
Length of Cross Product

$$A = |\mathbf{a} \times \mathbf{b}|$$



$$A_{\text{triangle}} = \frac{1}{2}|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)|$$

Tetrahedral Volume



$$h = \frac{\mathbf{a} \times \mathbf{b}}{|\mathbf{a} \times \mathbf{b}|} \cdot \mathbf{c}$$

$$h = \frac{\mathbf{a} \times \mathbf{b}}{A} \cdot \mathbf{c}$$

$$Ah = (\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}$$

$$V = (\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}$$

$$V_{\mathrm{tet}} = \frac{1}{6}(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}$$

Vector Transformations

3d Matrix

capital ↘

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$A\mathbf{x} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \end{bmatrix}$$

a vector

$$\mathrm{I}\,\mathbf{x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x} = \mathbf{x}$$

identity matrix

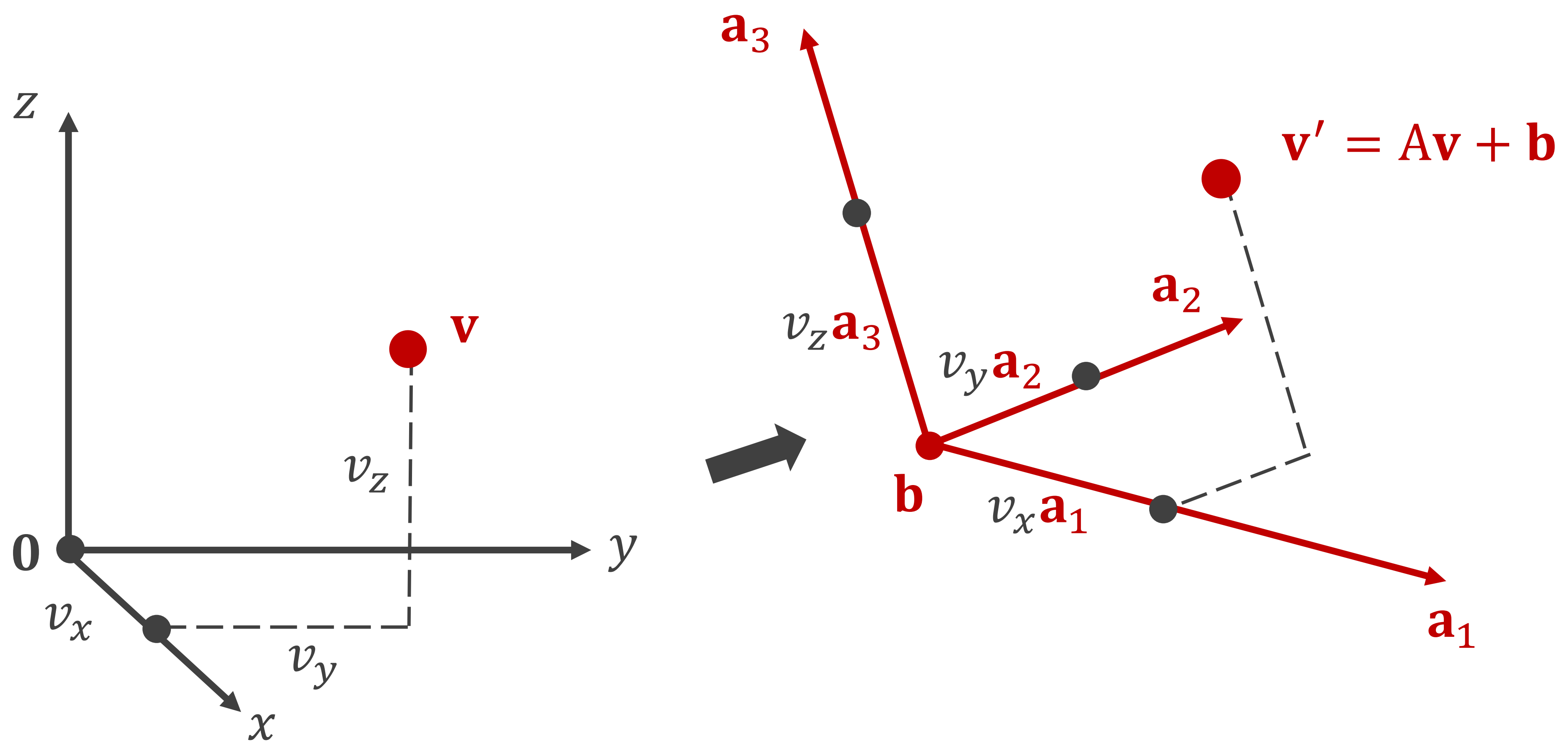
Column Representation

$$\mathrm{A}\mathbf{x} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \end{bmatrix}$$

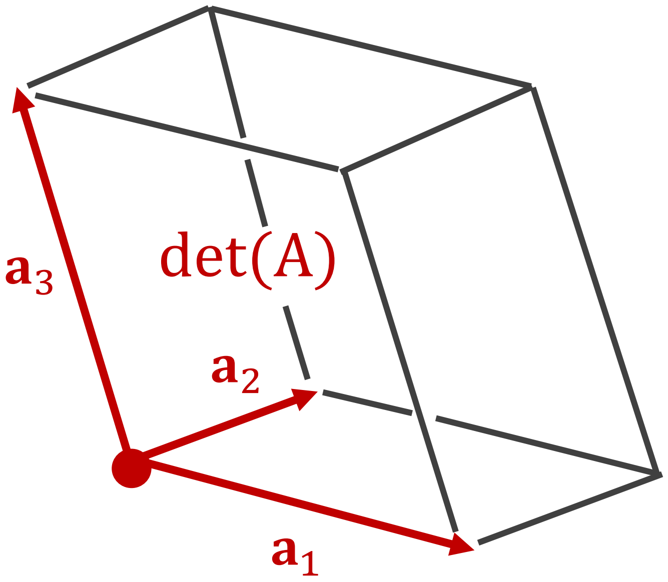
$$\mathrm{A}\mathbf{x} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3]\,\mathbf{x} = x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + x_3\mathbf{a}_3$$

Columns are Axes

$$\mathbf{v}' = \mathbf{b} + A\mathbf{v} = \mathbf{b} + v_x\mathbf{a}_1 + v_y\mathbf{a}_2 + v_z\mathbf{a}_3$$



Determinant of a 3x3 Matrix



$$\det(A) = (\mathbf{a}_1 \times \mathbf{a}_2) \cdot \mathbf{a}_3$$

$\det(A) = 1$: transformation is volume conserving

$\det(A) = 0$: not all points can be reached, inverse does not exist

3d Matrix Multiplication

Combining transformations:

$$\mathrm{B}(\mathrm{A}\mathbf{x}) = (\mathrm{BA})x = \mathrm{C}x$$

$$\mathrm{AB} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

$$c_{21} = a_{21}b_{11} + a_{22}b_{21} + a_{23}\, b_{31}$$

Matrix Inverse

Transform backwards

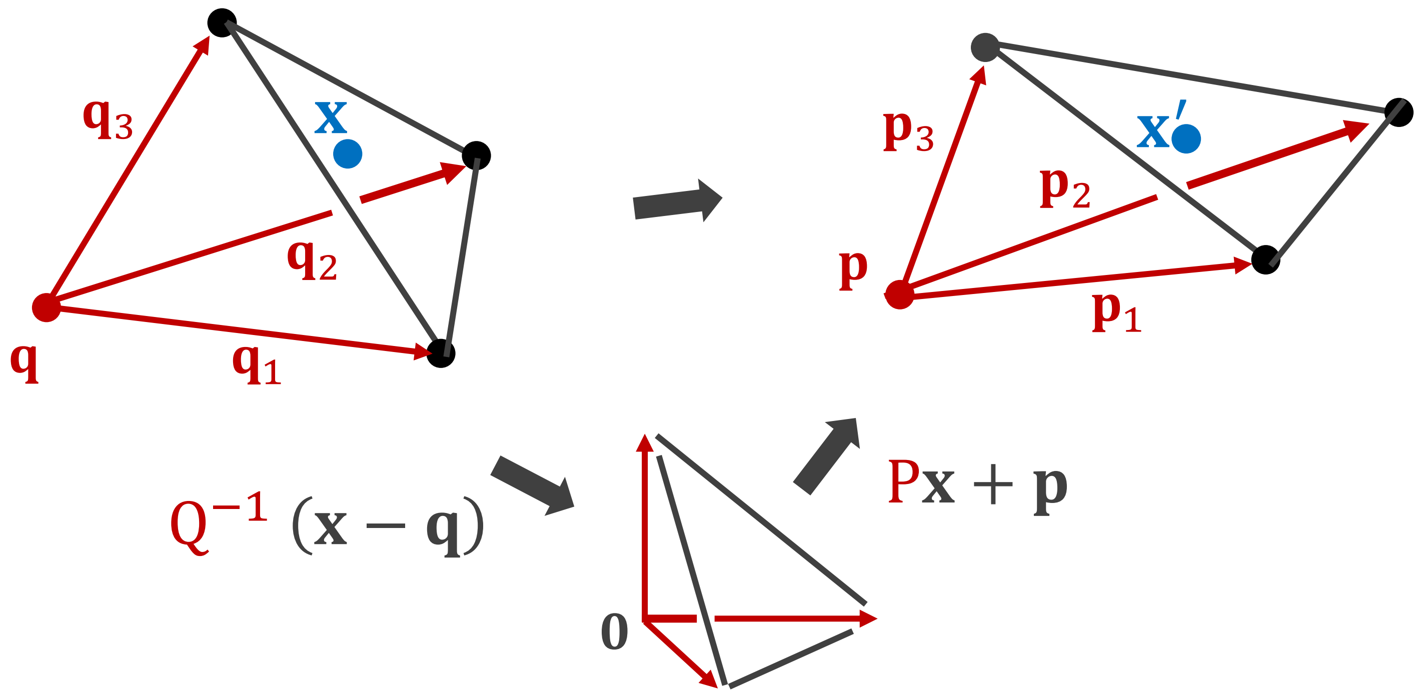
$$\mathrm{A}^{-1}(\mathrm{A}\mathbf{x}) = (\mathrm{A}^{-1}\mathrm{A})\mathbf{x} = \mathrm{I}\mathbf{x} = \mathbf{x}$$

Can be computed from the entries of A.

See textbooks.

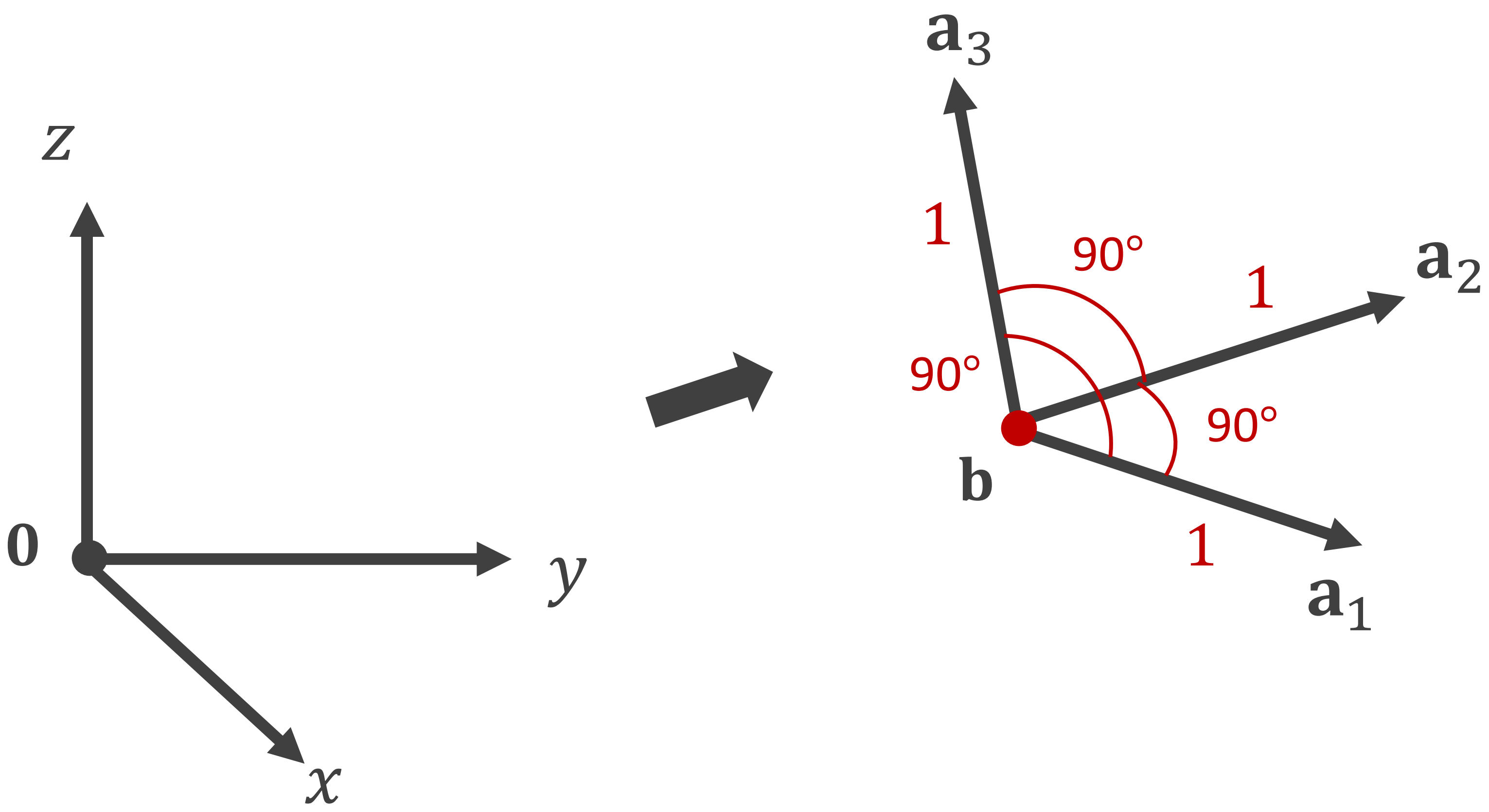
$$\mathbf{v}' = \mathrm{A}\mathbf{v} + \mathbf{b} \quad \Rightarrow \quad \mathbf{v} = \mathrm{A}^{-1}(\mathbf{v}' - \mathbf{b})$$

Tetrahedral Skinning



$$\mathbf{x}' = \mathrm{PQ}^{-1}\mathbf{x} + (\mathbf{p} - \mathrm{PQ}^{-1}\mathbf{q})$$

Rigid Transforms



The Transpose

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad A^{T} = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}$$

$$\mathbf{v} = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} \qquad \mathbf{v}^{T} = \begin{bmatrix} a_{11} & a_{21} & a_{31} \end{bmatrix}$$

Dot-less Dot Product

$$\mathbf{a}^{\mathrm{T}}\,\mathbf{b} = [a_x \quad a_y \quad a_z]\begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = a_x\,b_x + a_y\,b_y + a_z\,b_z = \mathbf{a}\cdot\mathbf{b}$$

dot product!

$$\mathbf{a}^{\mathrm{T}}\,\mathbf{a} = [a_x \quad a_y \quad a_z]\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \mathrm{a}_x^2 + \mathrm{a}_y^2 + \mathrm{a}_z^2 = |\mathbf{a}|^2$$

vector length squared

Rigid Transformations (Rotations)

$$R^T R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3] = \begin{bmatrix} |\mathbf{r}_1|^2 & \mathbf{r}_1 \cdot \mathbf{r}_2 & \mathbf{r}_1 \cdot \mathbf{r}_3 \\ \mathbf{r}_2 \cdot \mathbf{r}_1 & |\mathbf{r}_2|^2 & \mathbf{r}_2 \cdot \mathbf{r}_3 \\ \mathbf{r}_3 \cdot \mathbf{r}_1 & \mathbf{r}_3 \cdot \mathbf{r}_2 & |\mathbf{r}_3|^2 \end{bmatrix}$$

For a rigid transform: $\mathbf{r}_1 \cdot \mathbf{r}_2 = 0$ and $|\mathbf{r}_1|^2 = 1$

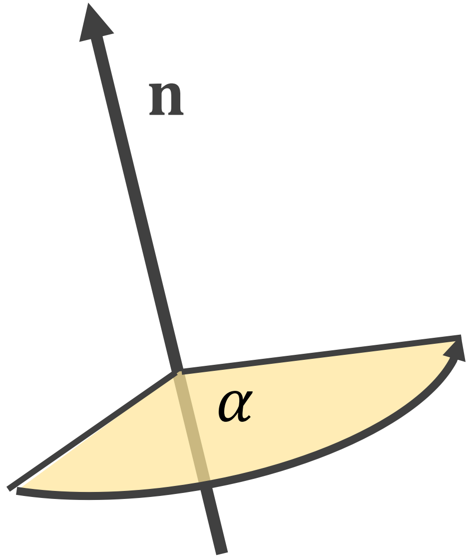
$$R^T R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I \quad \Rightarrow \quad R^{-1} = R^T$$

$$E_{deformation} = f\left(F^T F - I\right)$$

0 if not deformed

Axis and Angle

Every rotation in 3d can be expressed by a unit vector $\mathbf{n}$ and a scalar angle α

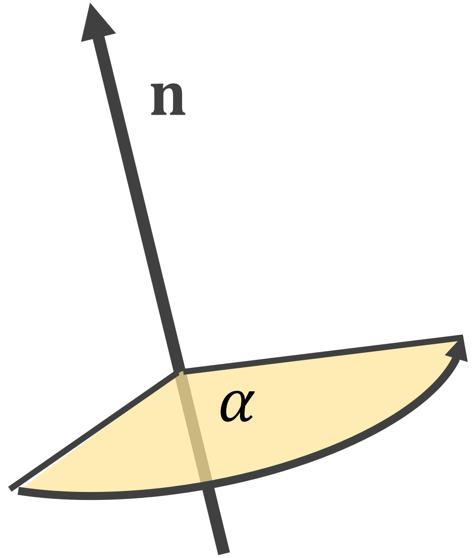


The corresponding rotation matrix is:

$$R(\alpha, \mathbf{n}) = \begin{bmatrix} \cos\alpha + n_x^2(1-\cos\alpha) & n_x n_y(1-\cos\alpha) - n_z\sin\alpha & n_y\sin\alpha + n_x n_z(1-\cos\alpha) \\ n_z\sin\alpha + n_x n_y(1-\cos\alpha) & \cos\alpha + n_y^2(1-\cos\alpha) & -n_x\sin\alpha + n_y n_z(1-\cos\alpha) \\ -n_y\sin\alpha + n_x n_z(1-\cos\alpha) & n_x\sin\alpha + n_y n_z(1-\cos\alpha) & \cos\alpha + n_z^2(1-\cos\alpha) \end{bmatrix}$$

9 values to store!

A Smaller Representation



$$\mathbf{r} = \alpha\,\mathbf{n} = [\alpha\, n_x \quad \alpha\, n_y \quad \alpha\, n_z]^{\mathrm{T}}$$

We need $\sin(\alpha)$ and $\cos(\alpha)$, expensive to compute!

Better:

$$\mathbf{q} = [\sin(\alpha)\, n_x \quad \sin(\alpha)\, n_y \quad \sin(\alpha)\, n_z \quad \cos(\alpha)]^{\mathrm{T}}$$

A quaternion!

Working with Quaternions

Vector rotation: $\mathbf{v}' = \mathrm{rot}(\mathbf{q}, \mathbf{v})$

Combining rotations: $\mathbf{q} = \mathbf{q_1}\ \mathbf{q_2}$

Inverse rotation $\mathbf{q}^{-1} = [-q_x \quad -q_y \quad -q_z \quad q_w]$

Only a few people know how to compute these

Everybody uses libraries, e.g. `THREE.Quaternion()`

Ready to write 3d simulations!