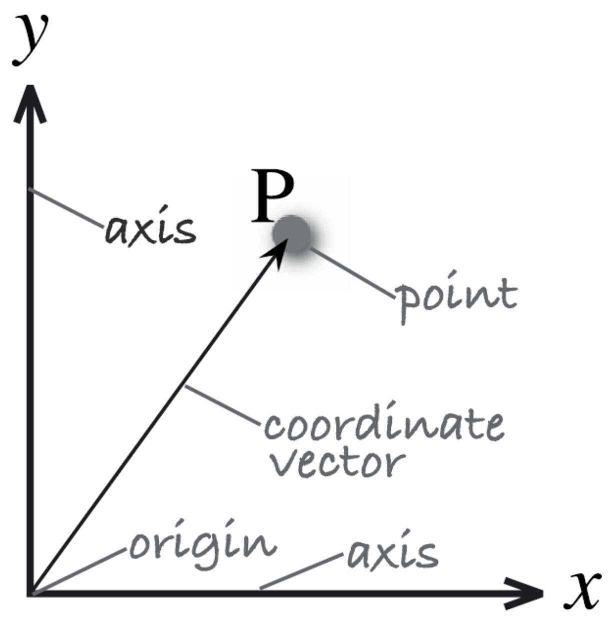
CS 3630

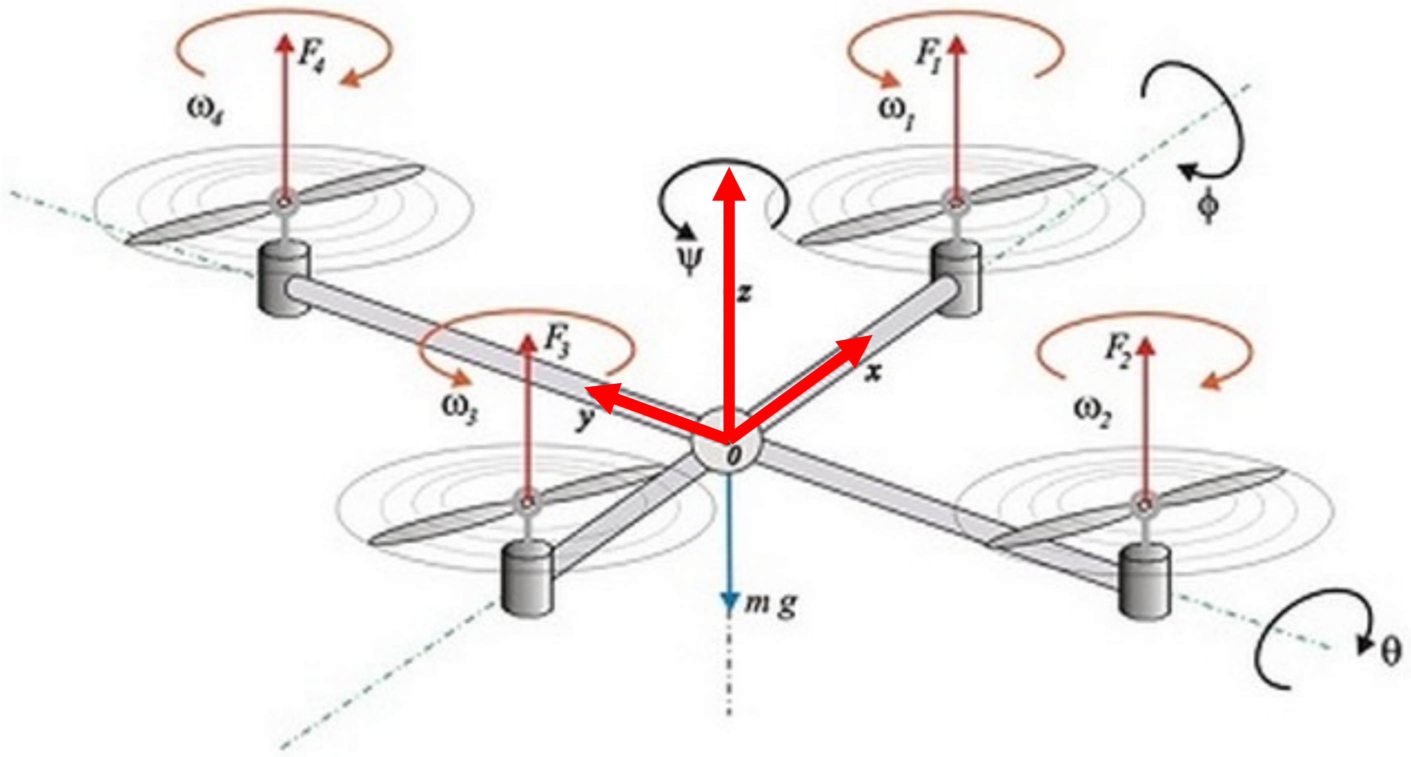
Pose in 3D

Reference Frames

- Robotics is all about management of reference frames
  - **Perception** is about estimation of reference frames
  - **Planning** is how to move reference frames
  - **Control** is the implementation of trajectories for reference frames
- The relation between references frames is essential to a successful system



Application to Drones



To characterize the position and orientation of a drone in flight,

- attach a coordinate frame to the drone (rigid attachment)
- specify the position and orientation of the frame.

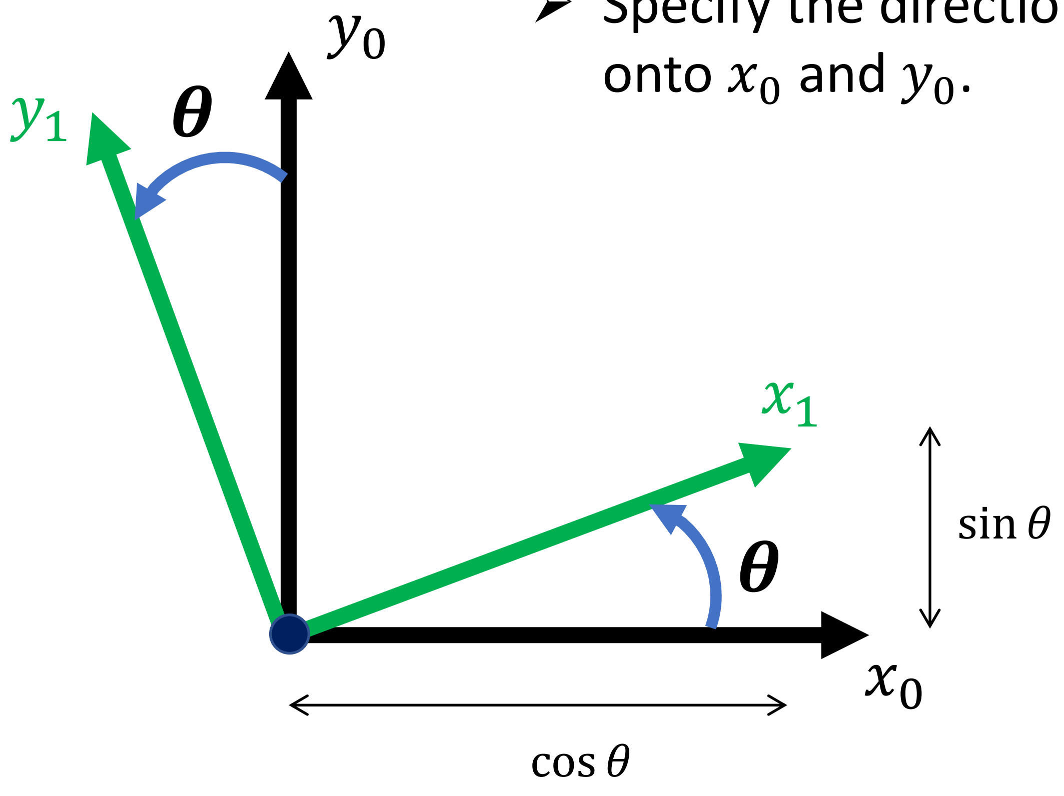
First… a quick review

Nearly everything we learned about position and orientation in the plane can be easily generalized to position and orientation in 3D.

We'll start with a quick review of the 2D case, then generalize to 3D, and show the corresponding mathematical formulations.

Specifying Orientation in the Plane

Given two coordinate frames with a common origin, how should we describe the orientation of Frame 1 w.r.t. Frame 0?



- Specify the directions of x_1 and y_1 with respect to Frame 0 by projecting onto x_0 and y_0.

$$x_1^0 = \begin{bmatrix} x_1 \cdot x_0 \\ x_1 \cdot y_0 \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

Notation: x_1^0 denotes the x-axis of Frame 1, specified w.r.t Frame 0.

$$y_1^0 = \begin{bmatrix} y_1 \cdot x_0 \\ y_1 \cdot y_0 \end{bmatrix} = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

We obtain y_1^0 in the same way.

Rotation Matrices (rotation in the plane)

We combine these two vectors to obtain a ***rotation matrix***: $R_1^0 = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

All rotation matrices have certain properties:

1. The two columns are each unit vectors.
2. The two columns are orthogonal, e.g., $c_1 \cdot c_2 = 0$.
3. $\det R = +1$

$\boxed{\textit{For such matrices } \boldsymbol{R^{-1} = R^T}}$

- The first two properties imply that the matrix R is ***orthogonal***.
- The third property implies that the matrix is ***special***! (After all, there are plenty of orthogonal matrices whose determinant is -1, not at all special.)

The collection of 2×2 rotation matrices is called the ***Special Orthogonal Group of order 2***, or, more commonly $\boldsymbol{SO(2)}$.

This concept generalizes to $\boldsymbol{SO(n)}$ for $n \times n$ rotation matrices.

Rotation Matrices (3D)

All of the properties of SO(2) apply as well to SO(3)!

All rotation matrices have certain properties:

1. The two columns are each unit vectors.
2. The two columns are orthogonal, e.g., $c_1 \cdot c_2 = 0$.
3. $\det R = +1$

$\boxed{\textbf{\textit{For such matrices }} \boldsymbol{R^{-1} = R^T}}$

- The first two properties imply that the matrix R is ***orthogonal***.
- The third property implies that the matrix is ***special***! (After all, there are plenty of orthogonal matrices whose determinant is -1, not at all special.)

The collection of 3×3 rotation matrices is called the ***<u>Special Orthogonal Group of order 3</u>***, or, more commonly $\underline{\boldsymbol{SO(3)}}$.

Rotation Matrices for 3D rotations

To build a rotation matrix, say R_1^0: project the axes of Frame 1 onto Frame 0. Each column of R_1^0 corresponds to the projection of one axis of Frame 1 onto Frame 0.

$$R_1^0 = [x_1 \cdot F_0 \mid y_1 \cdot F_0 \mid z_1 \cdot F_0] = \left[\begin{array}{c|c|c} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{array}\right]$$

Rotation Matrices for 3D rotations

To build a rotation matrix, say R_1^0: project the axes of Frame 1 onto Frame 0. Each column of R_1^0 corresponds to the projection of one axis of Frame 1 onto Frame 0.

$$R_1^0 = \begin{bmatrix} \boxed{x_1 \cdot F_0} & y_1 \cdot F_0 & z_1 \cdot F_0 \end{bmatrix} = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix}$$

Project the x-axis of Frame 1 onto the axes of Frame 0

Rotation Matrices for 3D rotations

To build a rotation matrix, say R_1^0: project the axes of Frame 1 onto Frame 0. Each column of R_1^0 corresponds to the projection of one axis of Frame 1 onto Frame 0.

$$R_1^0 = [x_1 \cdot F_0 \quad \boxed{y_1 \cdot F_0} \quad z_1 \cdot F_0] = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix}$$

Project the y-axis of Frame 1 onto the axes of Frame 0

Rotation Matrices for 3D rotations

To build a rotation matrix, say R_1^0: project the axes of Frame 1 onto Frame 0. Each column of R_1^0 corresponds to the projection of one axis of Frame 1 onto Frame 0.

$$R_1^0 = \begin{bmatrix} x_1 \cdot F_0 & y_1 \cdot F_0 & \boxed{z_1 \cdot F_0} \end{bmatrix} = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix}$$

Project the z-axis of Frame 1 onto the axes of Frame 0

Rotation Matrices for 3D rotations

To build a rotation matrix, say R_1^0: project the axes of Frame 1 onto Frame 0. Each column of R_1^0 corresponds to the projection of one axis of Frame 1 onto Frame 0.

$$R_1^0 = \begin{bmatrix} \boxed{x_1 \cdot F_0} & \boxed{y_1 \cdot F_0} & \boxed{z_1 \cdot F_0} \end{bmatrix} = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix}$$

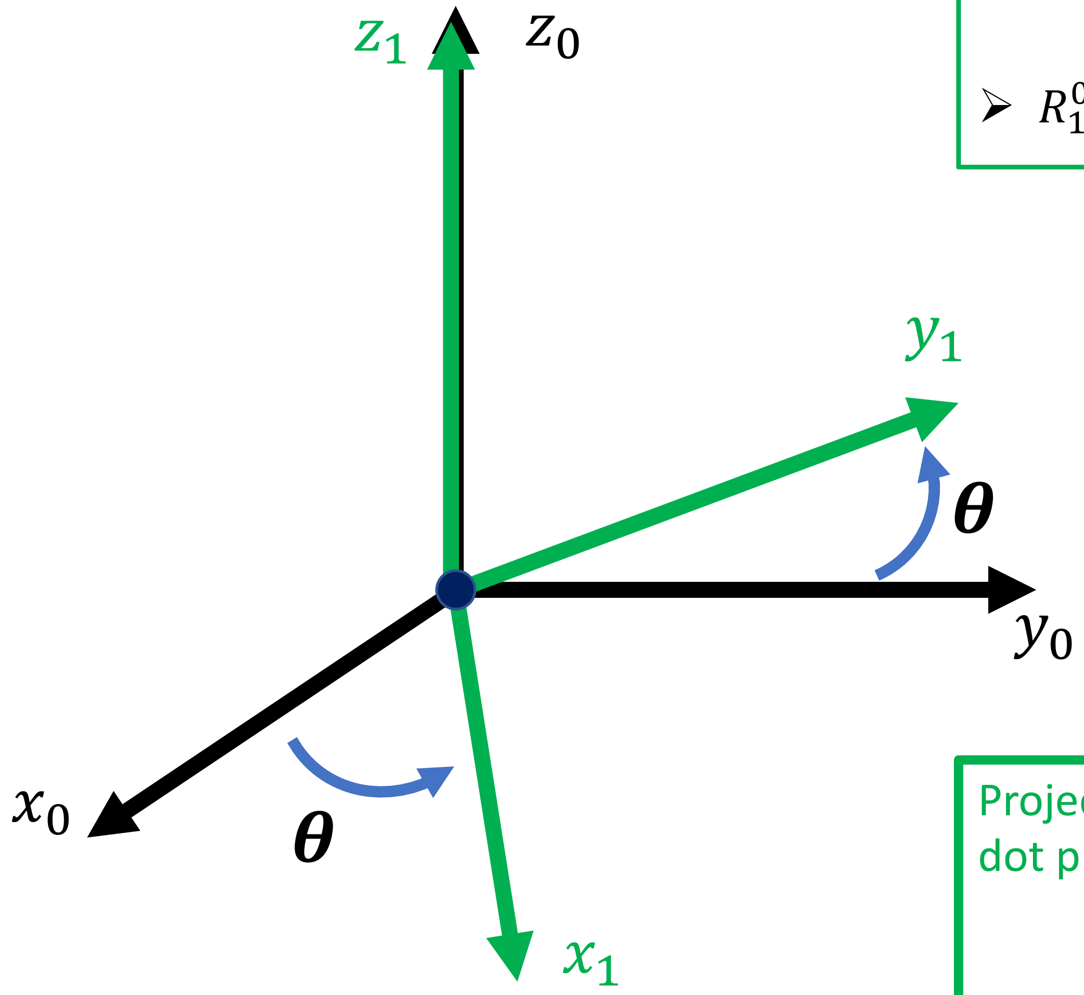
Project the x-axis of Frame 1 onto the axes of Frame 0

Project the y-axis of Frame 1 onto the axes of Frame 0

Project the z-axis of Frame 1 onto the axes of Frame 0

This process is exactly the same as the process for building rotation matrices in SO(2), even though it can be more difficult to visualize in 3D for rotation matrices in SO(3).

The simplest example: rotation about the z axis



Recall: for rotation in the plane, we built a rotation matrix as a function of θ, the angle between x_1 and x_0 (and also between y_1 and y_0):

➢ $R_1^0 = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ FOR ROTATION IN THE PLANE

This is *easily* extended to the case of rotation in 3D about the z-axis, since all of the interesting action is in the x-y plane (the two z-axes are the same)!

In fact, you'll see that the 2D rotation matrix shows up in the 3D rotation matrix:

➢ $R_1^0 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$ FOR ROTATION IN 3D

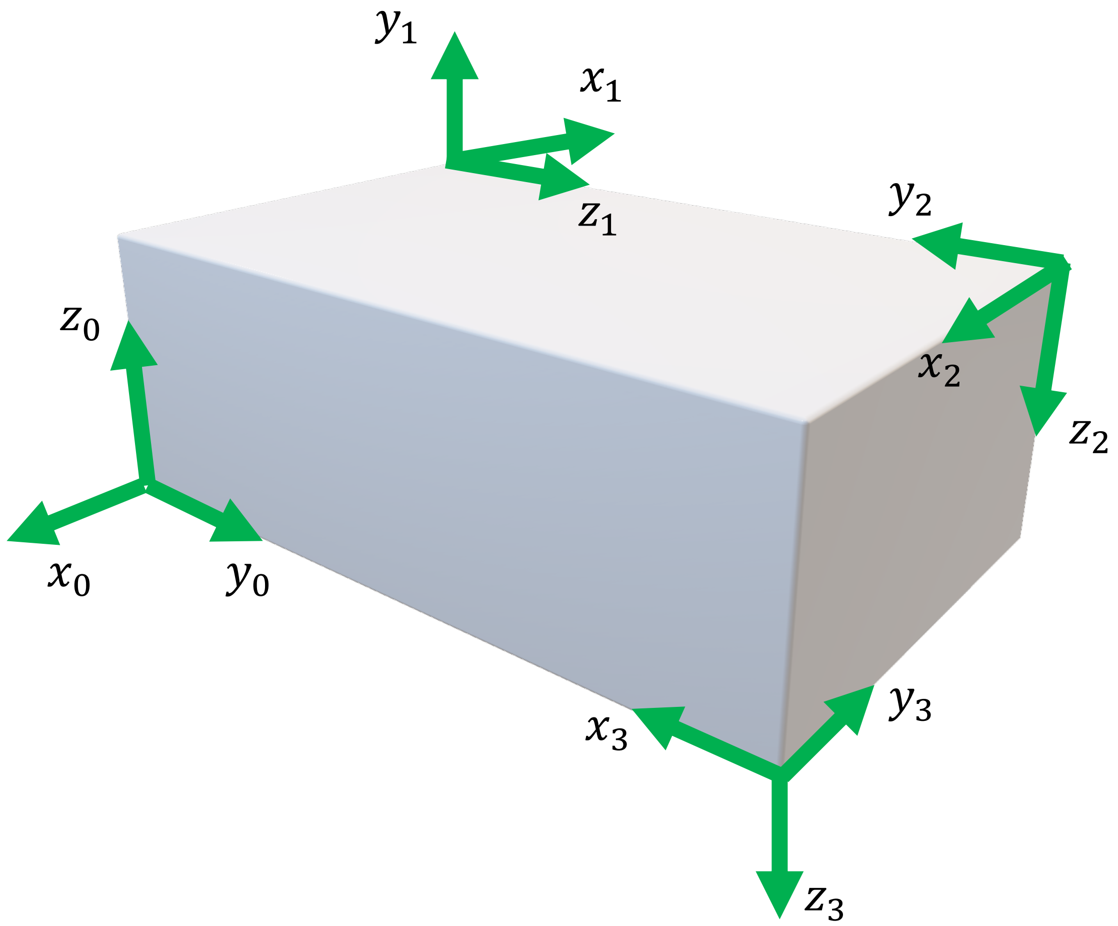
Projecting z_1 onto Frame 0 involves three dot products:

$$z_1 \cdot x_0 = 0$$
$$z_1 \cdot y_0 = 0$$
$$z_1 \cdot z_0 = 1$$

A bunch of examples:

A rectangular solid: all angles are multiples of $\pi/2$.

$$R_j^i = \begin{bmatrix} x_j \cdot x_i & y_j \cdot x_i & z_j \cdot x_i \\ x_j \cdot y_i & y_j \cdot y_i & z_j \cdot y_i \\ x_j \cdot z_i & y_j \cdot z_i & z_j \cdot z_i \end{bmatrix}$$



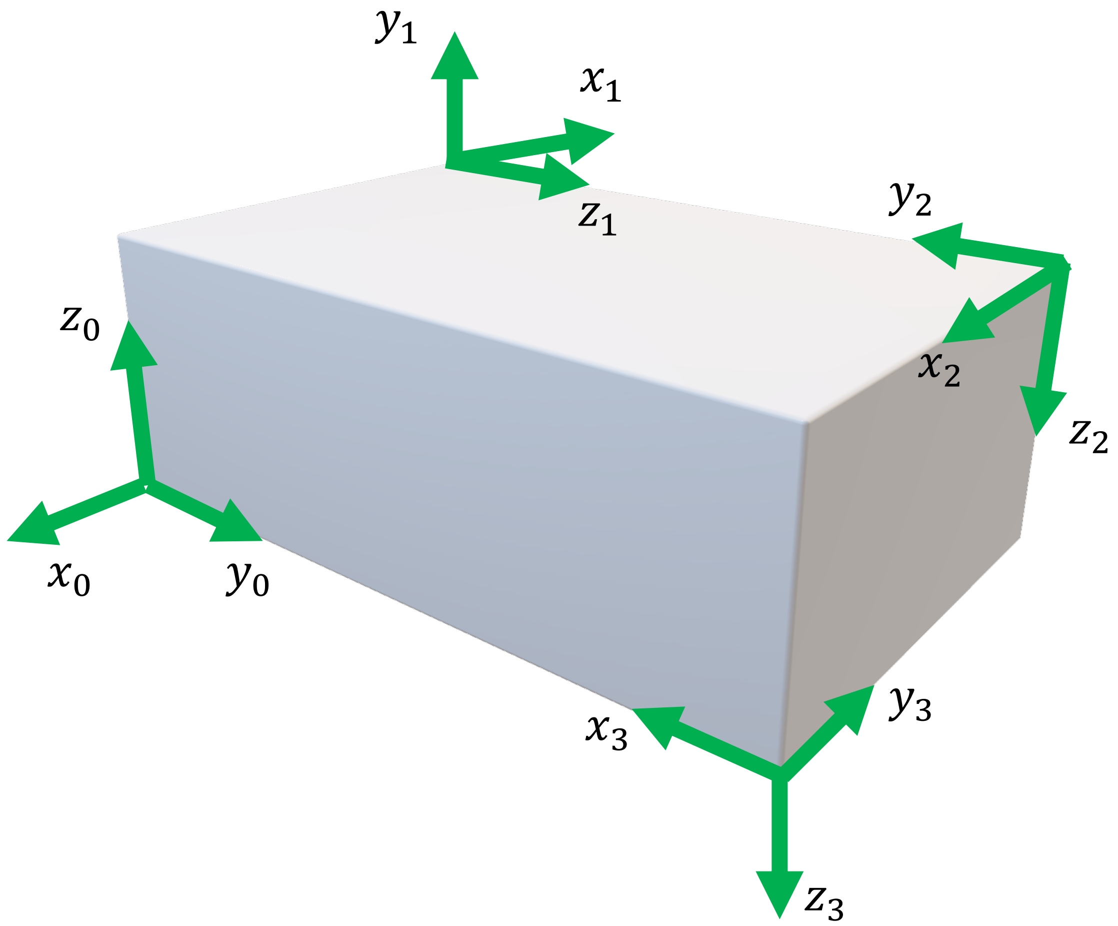
$$R_1^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$R_0^1 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

A bunch of examples:

A rectangular solid: all angles are multiples of $\pi/2$.

$$R_j^i = \begin{bmatrix} x_j \cdot x_i & y_j \cdot x_i & z_j \cdot x_i \\ x_j \cdot y_i & y_j \cdot y_i & z_j \cdot y_i \\ x_j \cdot z_i & y_j \cdot z_i & z_j \cdot z_i \end{bmatrix}$$



$$R_1^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad R_0^1 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

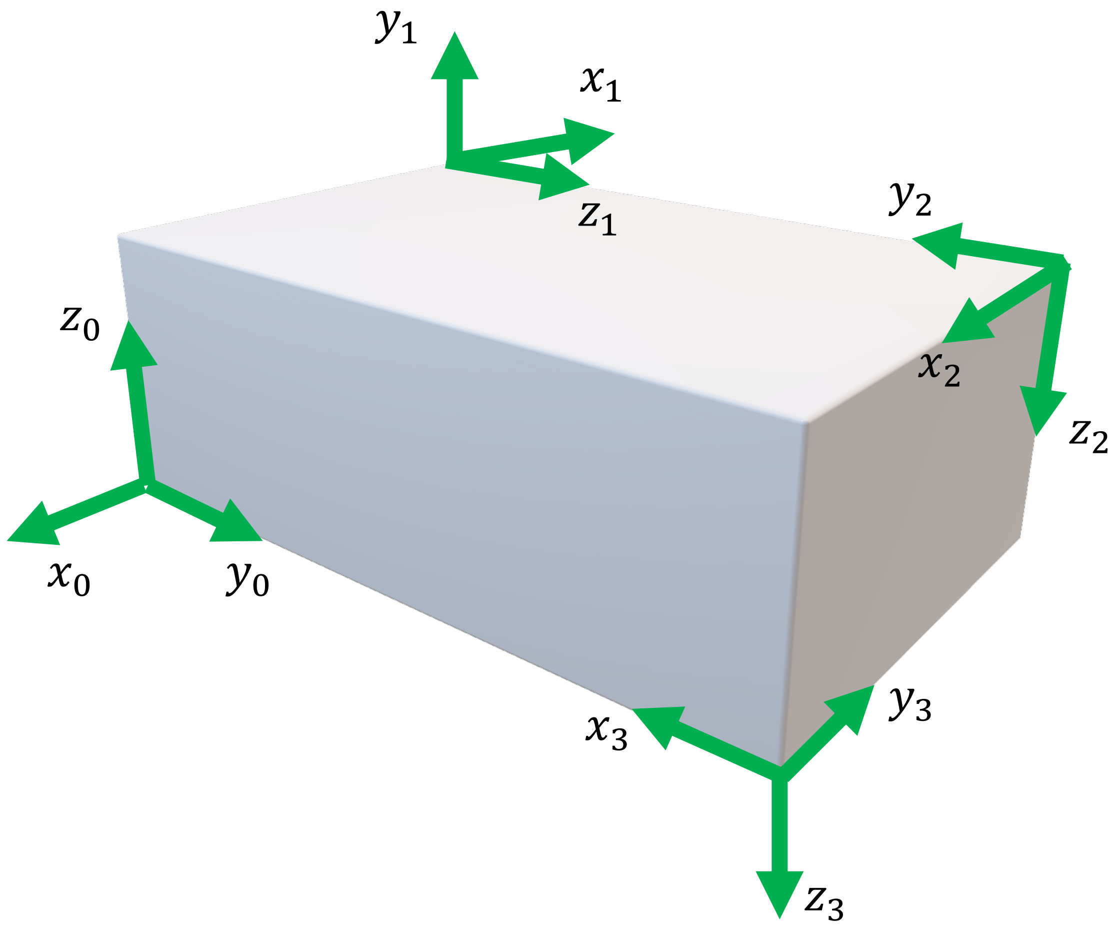
$$R_1^0 R_0^1 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boxed{(R_1^0)^{-1} = R_0^1 = (R_1^0)^T}$$

A bunch of examples:

A rectangular solid: all angles are multiples of $\pi/2$.

$$R_j^i = \begin{bmatrix} x_j \cdot x_i & y_j \cdot x_i & z_j \cdot x_i \\ x_j \cdot y_i & y_j \cdot y_i & z_j \cdot y_i \\ x_j \cdot z_i & y_j \cdot z_i & z_j \cdot z_i \end{bmatrix}$$



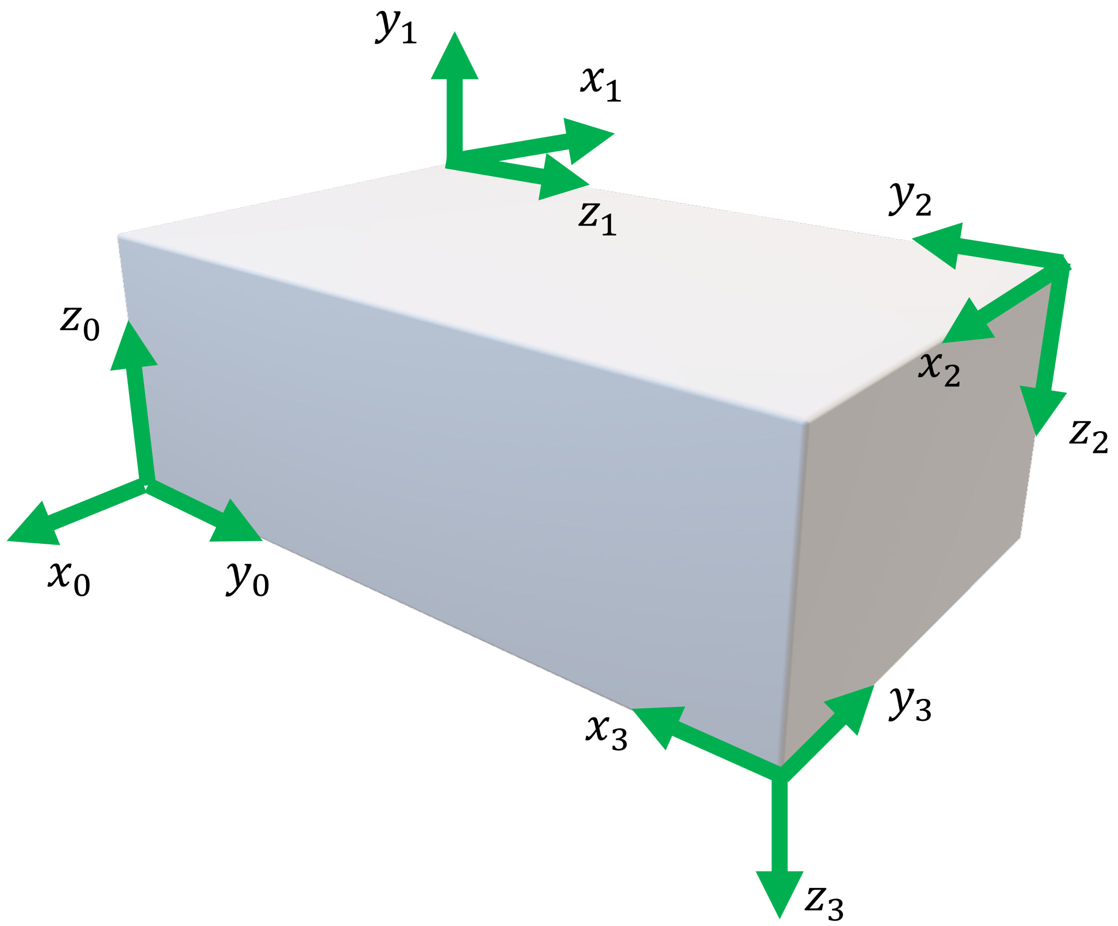
$$R_2^1 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$R_2^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

A bunch of examples:

A rectangular solid: all angles are multiples of $\pi/2$.

$$R_j^i = \begin{bmatrix} x_j \cdot x_i & y_j \cdot x_i & z_j \cdot x_i \\ x_j \cdot y_i & y_j \cdot y_i & z_j \cdot y_i \\ x_j \cdot z_i & y_j \cdot z_i & z_j \cdot z_i \end{bmatrix}$$



$$R_3^0 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

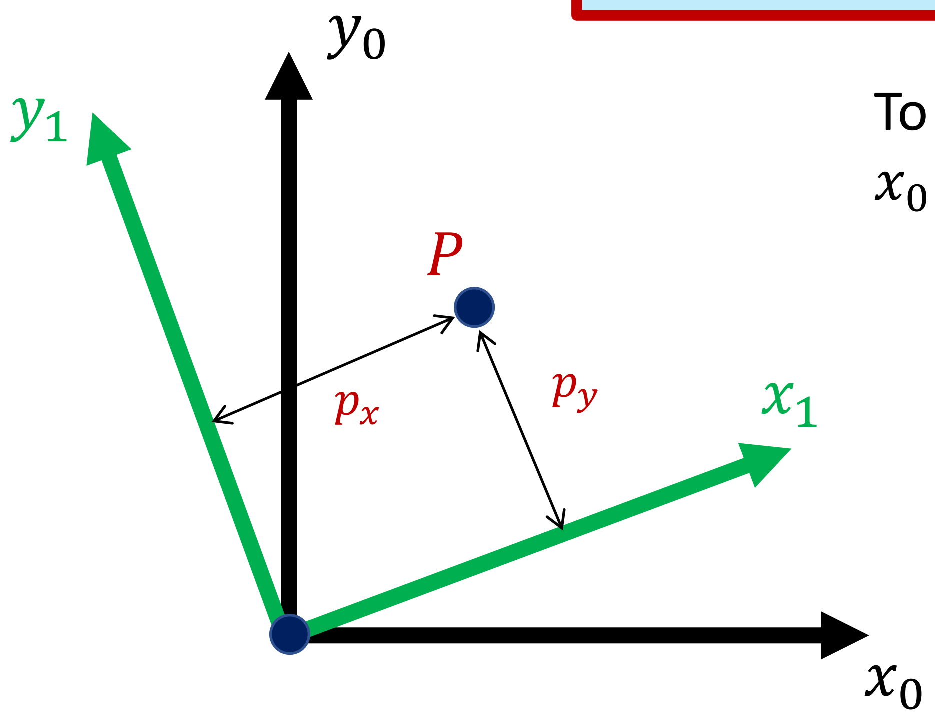
Let’s extend this to 3D rotational coordinate transformations.

Coordinate Transformations (rotation only)

Suppose a point P is rigidly attached to coordinate Frame 1, with coordinates given by $P^1 = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$.

We can express the location of the point P in terms of its coordinates

$$P = p_x x_1 + p_y y_1$$



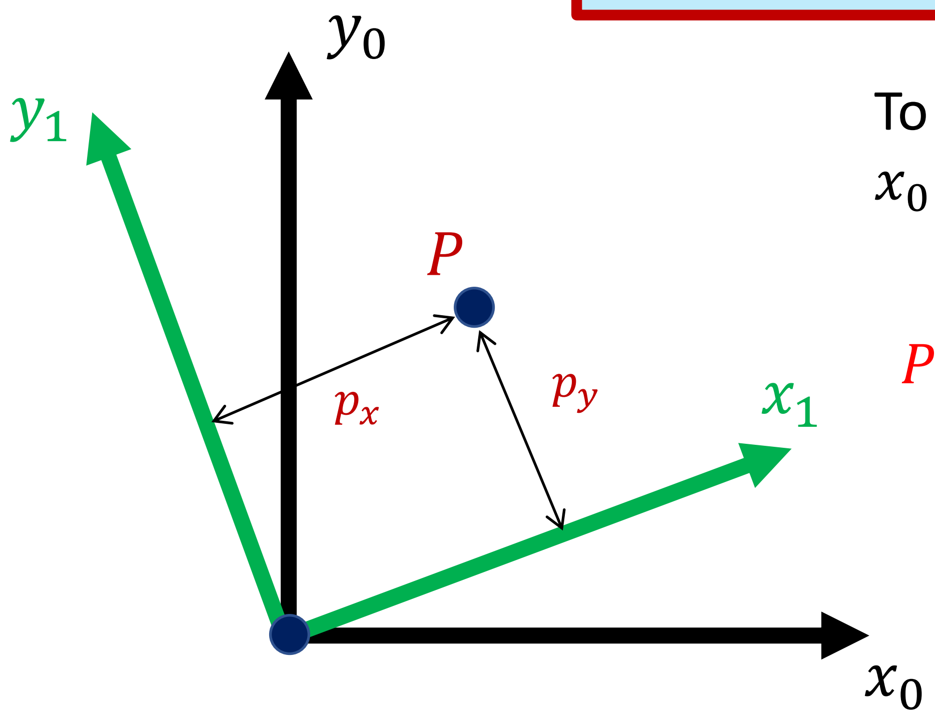
To obtain the coordinates of P w.r.t. Frame 0, we project P onto the x_0 and y_0 axes:

Coordinate Transformations (rotation only)

Suppose a point P is rigidly attached to coordinate Frame 1, with coordinates given by $P^1 = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$.

We can express the location of the point P in terms of its coordinates

$$P = p_x x_1 + p_y y_1$$



To obtain the coordinates of P w.r.t. Frame 0, we project P onto the x_0 and y_0 axes:

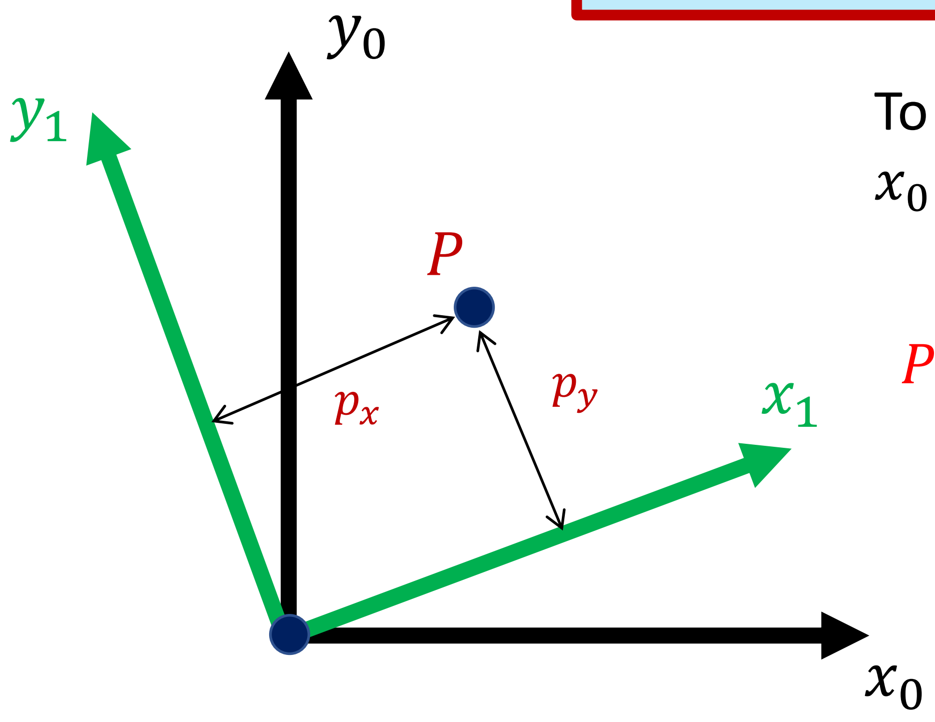
$$P^0 = \begin{bmatrix} P \cdot x_0 \\ P \cdot y_0 \end{bmatrix} =$$

Coordinate Transformations (rotation only)

Suppose a point P is rigidly attached to coordinate Frame 1, with coordinates given by ${}^1P = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$.

We can express the location of the point P in terms of its coordinates

$$P = p_x x_1 + p_y y_1$$



To obtain the coordinates of P w.r.t. Frame 0, we project P onto the x_0 and y_0 axes:

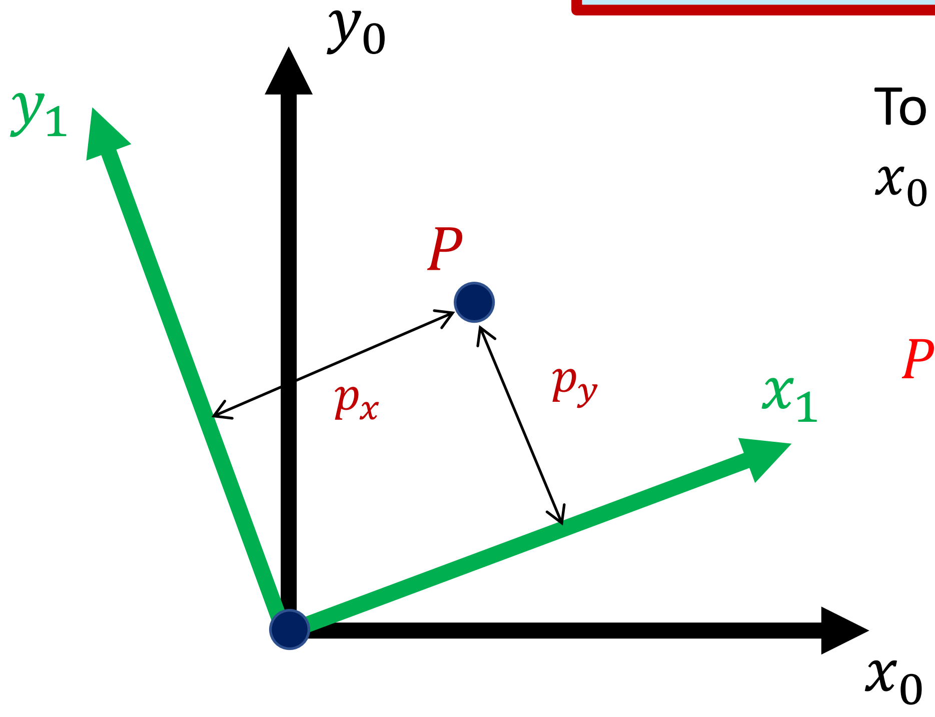
$$P^0 = \begin{bmatrix} P \cdot x_0 \\ P \cdot y_0 \end{bmatrix} = \begin{bmatrix} (p_x x_1 + p_y y_1) \cdot x_0 \\ (p_x x_1 + p_y y_1) \cdot y_0 \end{bmatrix} =$$

Coordinate Transformations (rotation only)

Suppose a point P is rigidly attached to coordinate Frame 1, with coordinates given by ${}^1P = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$.

We can express the location of the point P in terms of its coordinates

$$P = p_x x_1 + p_y y_1$$



To obtain the coordinates of P w.r.t. Frame 0, we project P onto the x_0 and y_0 axes:

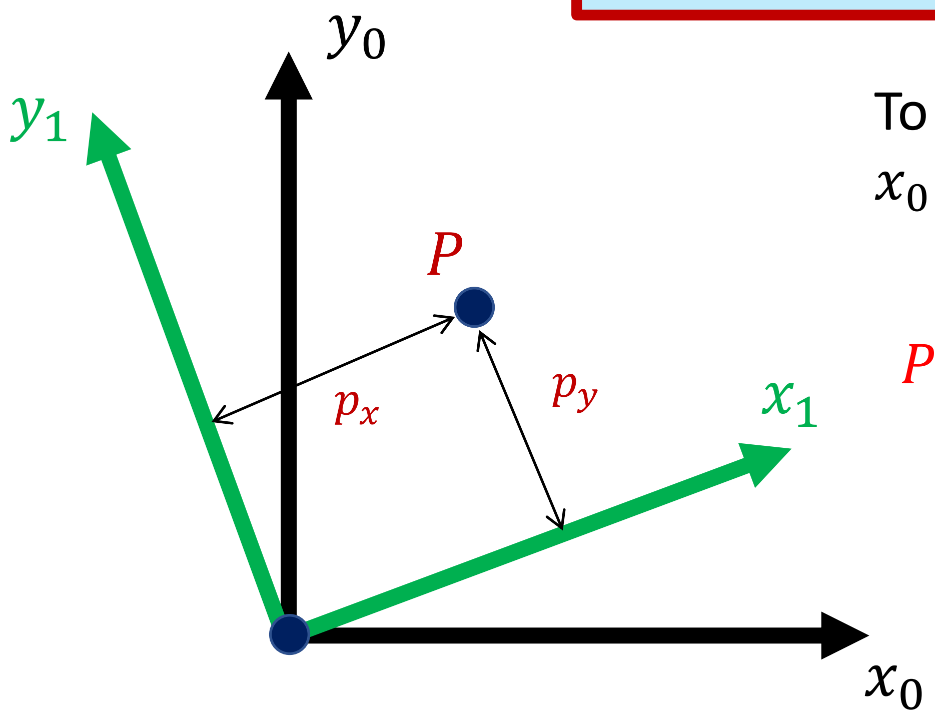
$$P^0 = \begin{bmatrix} P \cdot x_0 \\ P \cdot y_0 \end{bmatrix} = \begin{bmatrix} (p_x x_1 + p_y y_1) \cdot x_0 \\ (p_x x_1 + p_y y_1) \cdot y_0 \end{bmatrix} = \begin{bmatrix} p_x (x_1 \cdot x_0) + p_y (y_1 \cdot x_0) \\ p_x (x_1 \cdot y_0) + p_y (y_1 \cdot y_0) \end{bmatrix}$$

Coordinate Transformations (rotation only)

Suppose a point P is rigidly attached to coordinate Frame 1, with coordinates given by ${}^1P = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$.

We can express the location of the point P in terms of its coordinates

$$P = p_x x_1 + p_y y_1$$



To obtain the coordinates of P w.r.t. Frame 0, we project P onto the x_0 and y_0 axes:

$$P^0 = \begin{bmatrix} P \cdot x_0 \\ P \cdot y_0 \end{bmatrix} = \begin{bmatrix} (p_x x_1 + p_y y_1) \cdot x_0 \\ (p_x x_1 + p_y y_1) \cdot y_0 \end{bmatrix} = \begin{bmatrix} p_x (x_1 \cdot x_0) + p_y (y_1 \cdot x_0) \\ p_x (x_1 \cdot y_0) + p_y (y_1 \cdot y_0) \end{bmatrix}$$

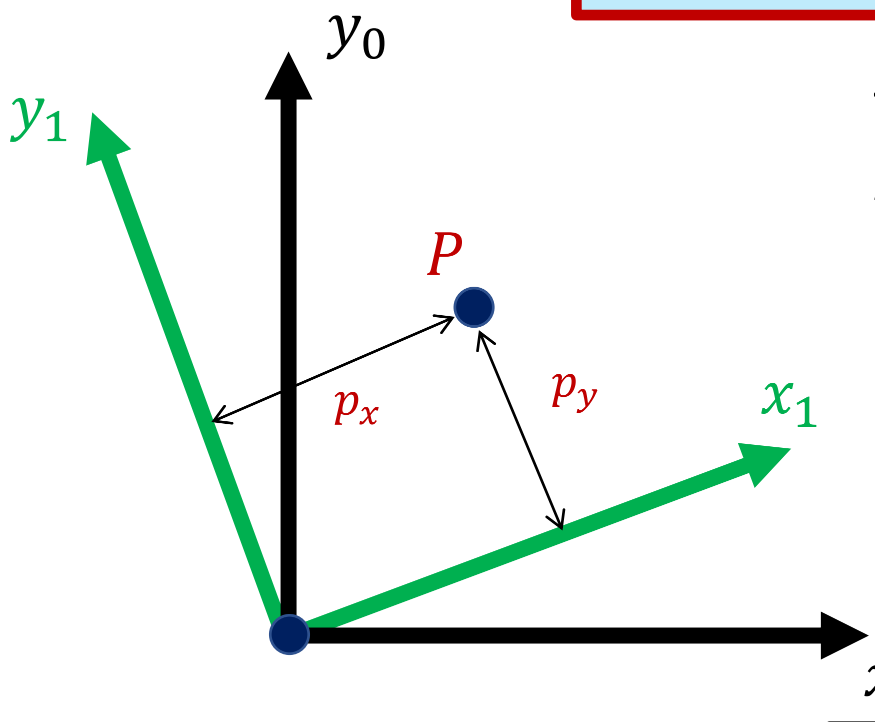
$$= \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

Coordinate Transformations (rotation only)

Suppose a point P is rigidly attached to coordinate Frame 1, with coordinates given by $P^1 = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$.

We can express the location of the point P in terms of its coordinates

$$P = p_x x_1 + p_y y_1$$



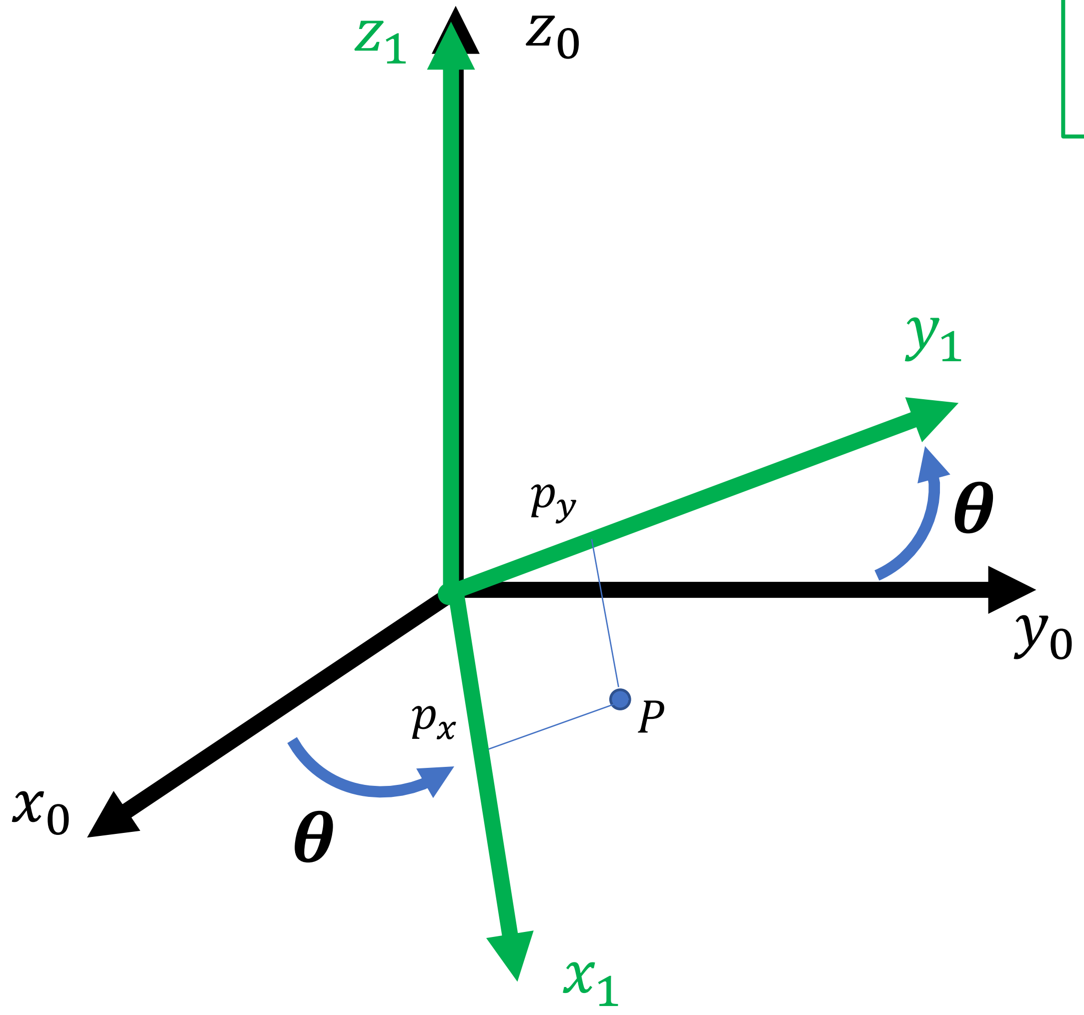
To obtain the coordinates of P w.r.t. Frame 0, we project P onto the x_0 and y_0 axes:

$$p^0 = \begin{bmatrix} P \cdot x_0 \\ P \cdot y_0 \end{bmatrix} = \begin{bmatrix} (p_x x_1 + p_y y_1) \cdot x_0 \\ (p_x x_1 + p_y y_1) \cdot y_0 \end{bmatrix} = \begin{bmatrix} p_x (x_1 \cdot x_0) + p_y (y_1 \cdot x_0) \\ p_x (x_1 \cdot y_0) + p_y (y_1 \cdot y_0) \end{bmatrix}$$

$$= \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \boldsymbol{R_1^0}\, \boldsymbol{P^1}$$

$$\boxed{\boldsymbol{P^0 = R_1^0\, P^1}}$$

The simplest example: rotation about the z axis



As we saw above:

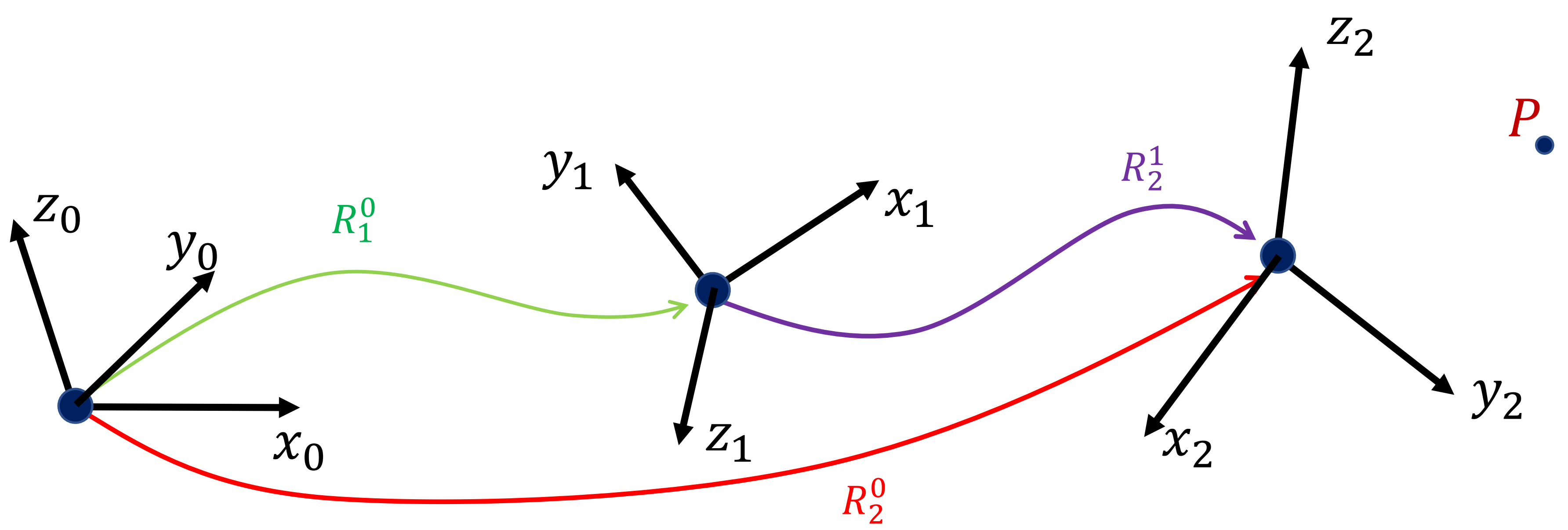
$$R_1^0 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The equation for rotational coordinate transformations generalizes immediately to the 3D case!

$$\boldsymbol{P^0} = \boldsymbol{R_1^0}\,\boldsymbol{P^1} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix}$$

Composition of Rotations

For now, only consider the rotation, *not the translation*!
This is an "exploded" view of three coordinate frames that share the same origin.



From our previous results, we know:

$$\left.\begin{aligned} P^0 &= R_1^0 P^1 \\ P^1 &= R_2^1 P^2 \end{aligned}\right\} \longrightarrow P^0 = R_1^0 R_2^1 P^2$$

But we also know: $P^0 = R_2^0 P^2$

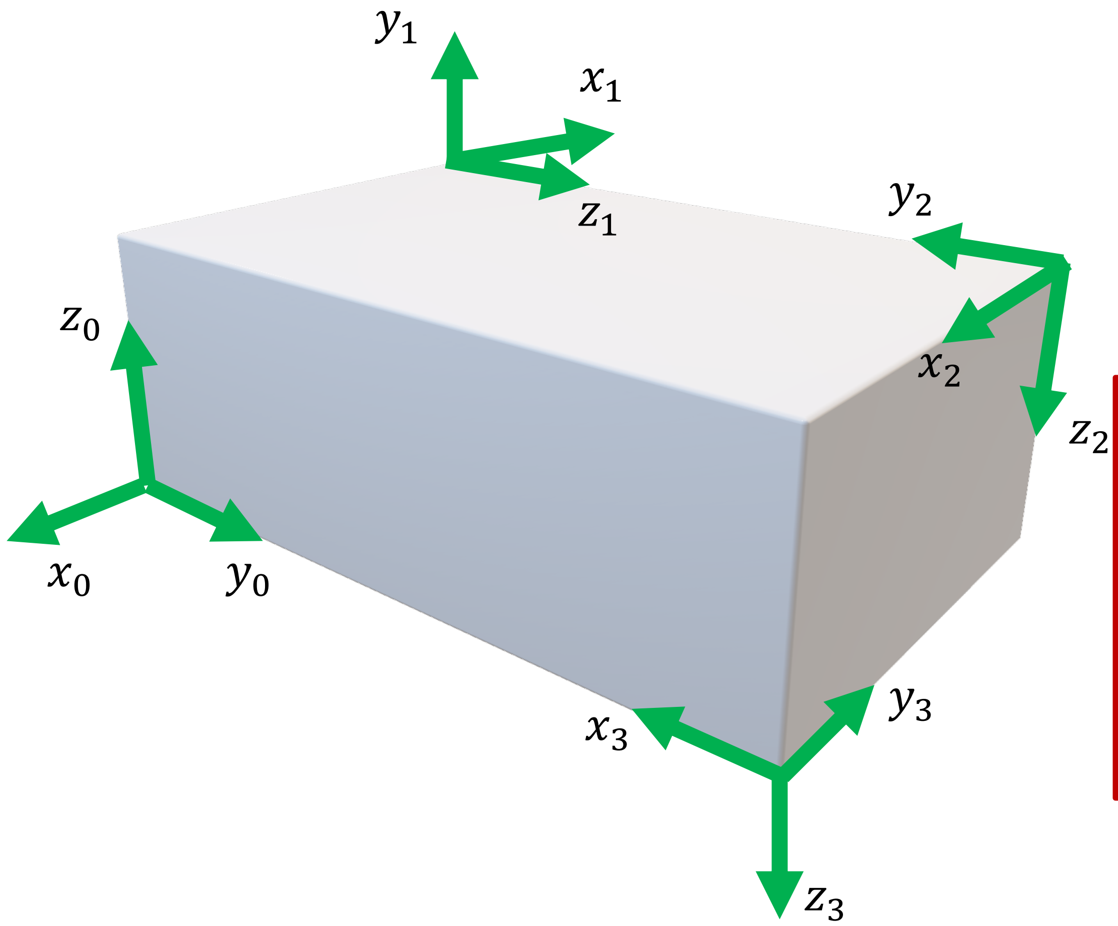
$$\left.\begin{aligned} P^0 &= R_1^0 R_2^1 P^2 \\ P^0 &= R_2^0 P^2 \end{aligned}\right\} \longrightarrow R_2^0 = R_1^0 R_2^1$$

This is the composition law for rotation transformations.

A bunch of examples:

A rectangular solid: all angles are multiples of $\pi/2$.

$$R_j^i = \begin{bmatrix} x_j \cdot x_i & y_j \cdot x_i & z_j \cdot x_i \\ x_j \cdot y_i & y_j \cdot y_i & z_j \cdot y_i \\ x_j \cdot z_i & y_j \cdot z_i & z_j \cdot z_i \end{bmatrix}$$



$$R_1^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad R_2^1 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$R_2^0 = R_1^0 R_2^1$$

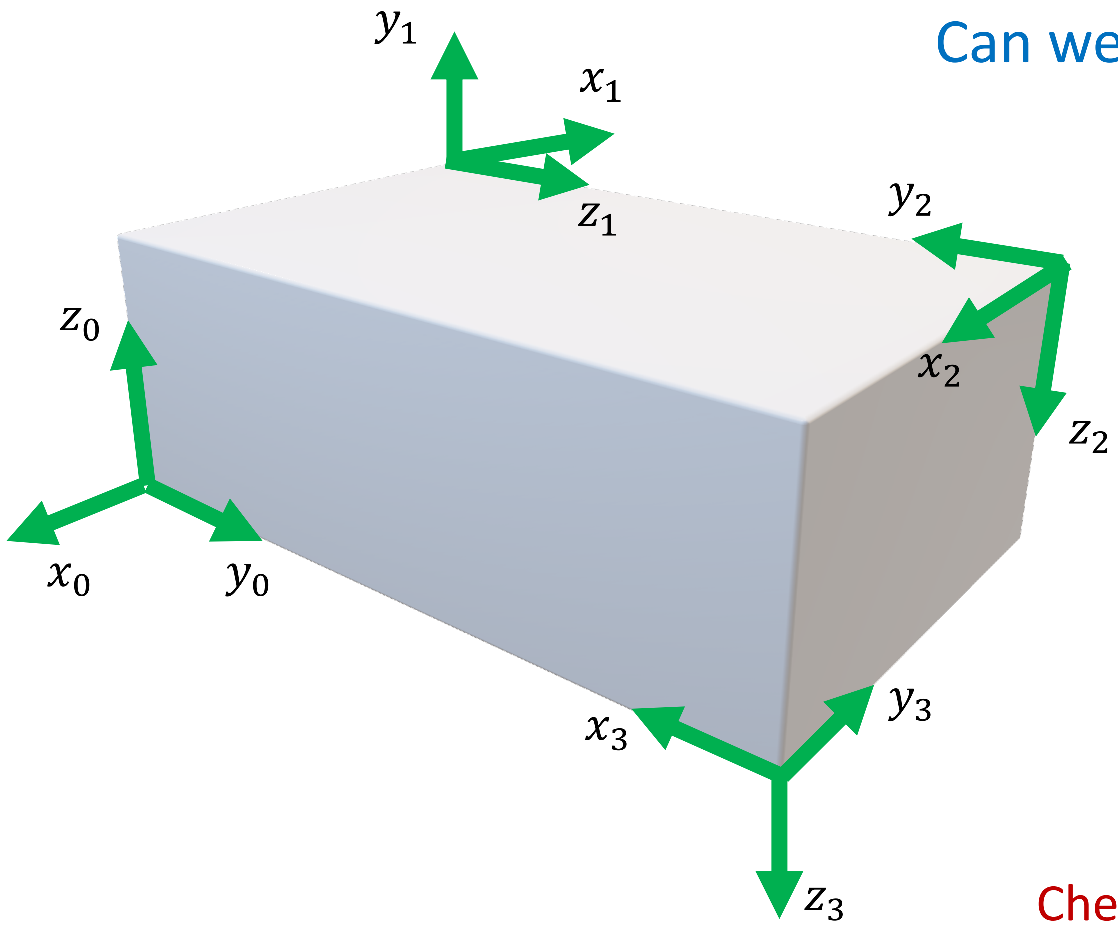
$$\boxed{R_2^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}}$$

This agrees with our earlier result!

A bunch of examples:

A rectangular solid: all angles are multiples of $\pi/2$.

$$R_j^i = \begin{bmatrix} x_j \cdot x_i & y_j \cdot x_i & z_j \cdot x_i \\ x_j \cdot y_i & y_j \cdot y_i & z_j \cdot y_i \\ x_j \cdot z_i & y_j \cdot z_i & z_j \cdot z_i \end{bmatrix}$$



In preceding examples, we have computed R_1^0, R_2^0, R_3^0.

Can we compute R_3^2?

$$R_3^0 = R_2^0 R_3^2$$

$$(R_2^0)^{-1}\, R_3^0 = R_3^2$$

$$(R_2^0)^T\, R_3^0 = R_3^2$$

$$R_0^2 R_3^0 = R_3^2$$

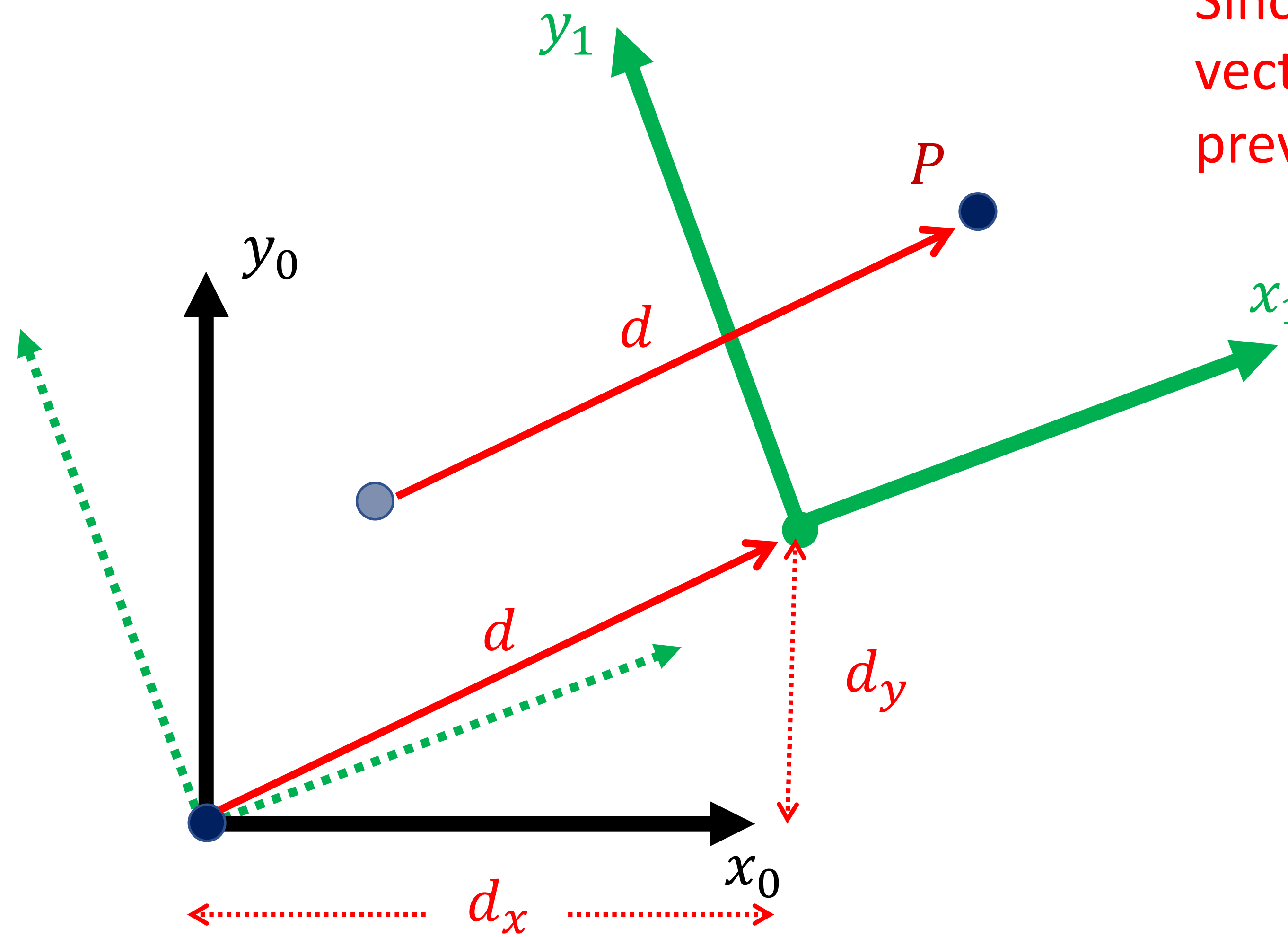
$$R_3^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Check this against the figure by directly determining R_3^2… it works!

Now let’s add translation…

Specifying Pose in the Plane

Suppose we now translate Frame 1 (*no new rotatation*).
What are the coordinates of P w.r.t. Frame 0?



Since we merely translated P by a fixed vector d, simply add the offset to our previous result!

$$\boldsymbol{P^0} = \boldsymbol{R_1^0}\,\boldsymbol{P^1} + \boldsymbol{d^0}$$

$$\boldsymbol{d^0} = \begin{bmatrix} \boldsymbol{d_x} \\ \boldsymbol{d_y} \end{bmatrix}$$

Homogeneous Transformations

We can simplify the equation for coordinate transformations by augmenting the vectors and matrices with an extra row:

This is just our eqn from the previous page

$$\boxed{\begin{bmatrix} \boldsymbol{P}^0 \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_1^0 \boldsymbol{P}^1 + \boldsymbol{d}^0 \\ 1 \end{bmatrix}} = \begin{bmatrix} \boldsymbol{R}_1^0 & \boldsymbol{d}^0 \\ 0_n & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{P}^1 \\ 1 \end{bmatrix}$$

in which $0_n = [0 \quad \cdots \quad 0]$

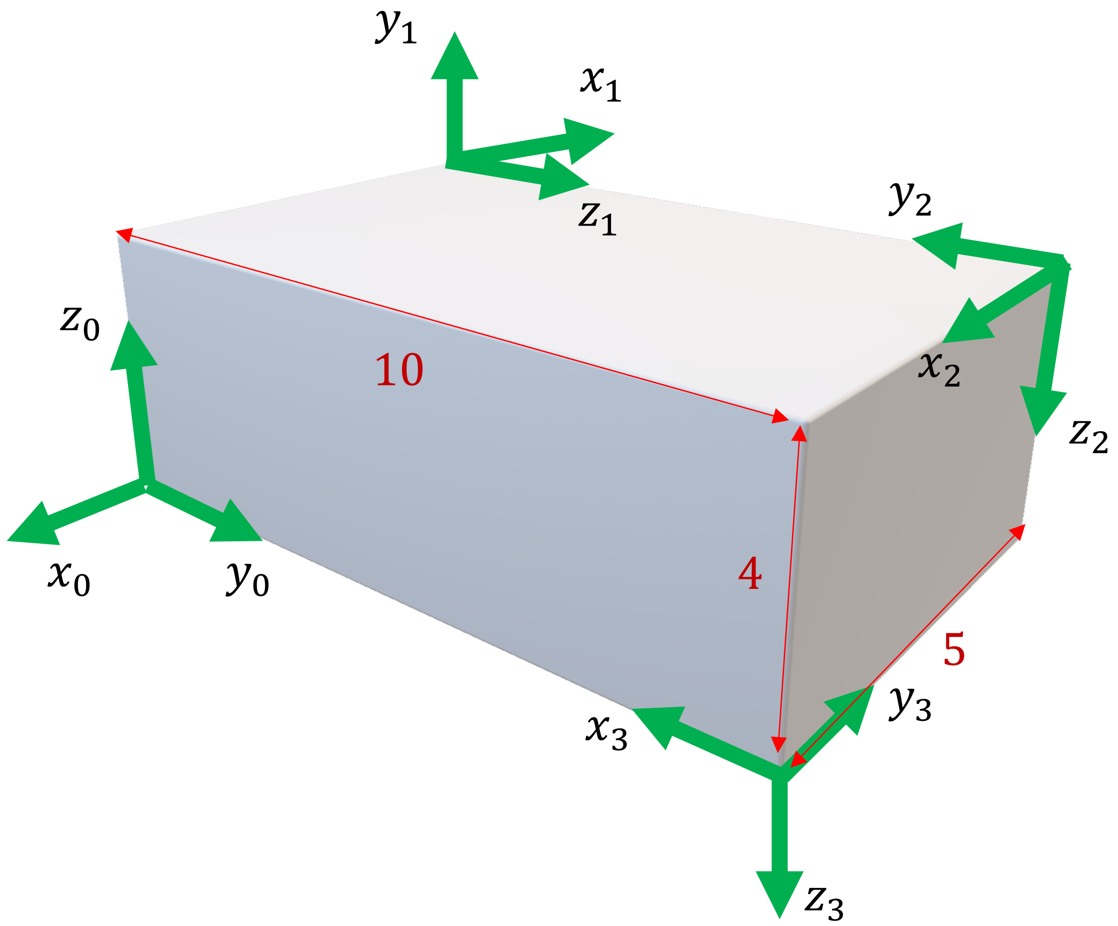
The set of matrices of the form $\begin{bmatrix} R & d \\ 0_n & 1 \end{bmatrix}$, where $R \in SO(n)$ and $d \in \mathbb{R}^n$ is called

the ***Special Euclidean Group of order n***, or $SE(n)$.

A bunch of examples:

A rectangular solid: all angles are multiples of $\pi/2$.

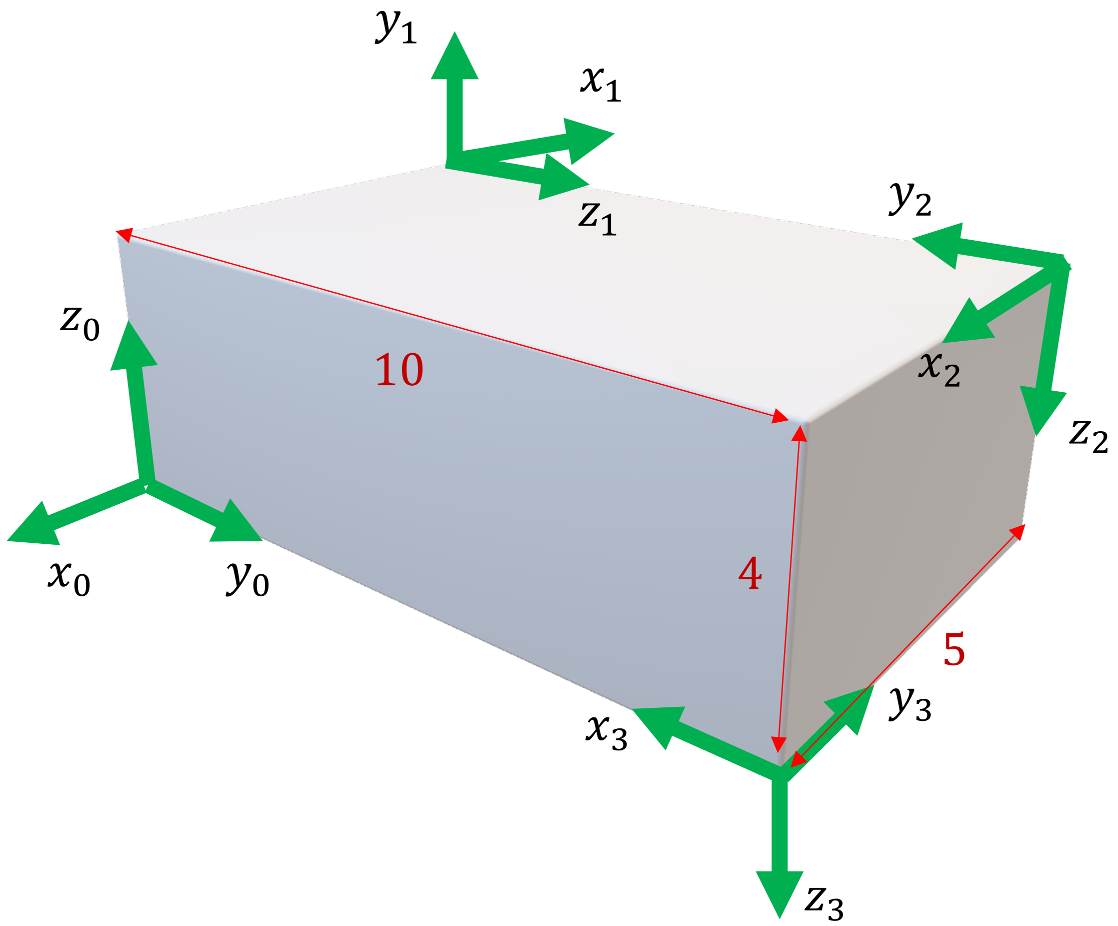
$$R_j^i = \begin{bmatrix} x_j \cdot x_i & y_j \cdot x_i & z_j \cdot x_i \\ x_j \cdot y_i & y_j \cdot y_i & z_j \cdot y_i \\ x_j \cdot z_i & y_j \cdot z_i & z_j \cdot z_i \end{bmatrix}$$



Now let's look at both the relative orientation and relative position of frames.

A bunch of examples:

A rectangular solid: all angles are multiples of $\pi/2$.

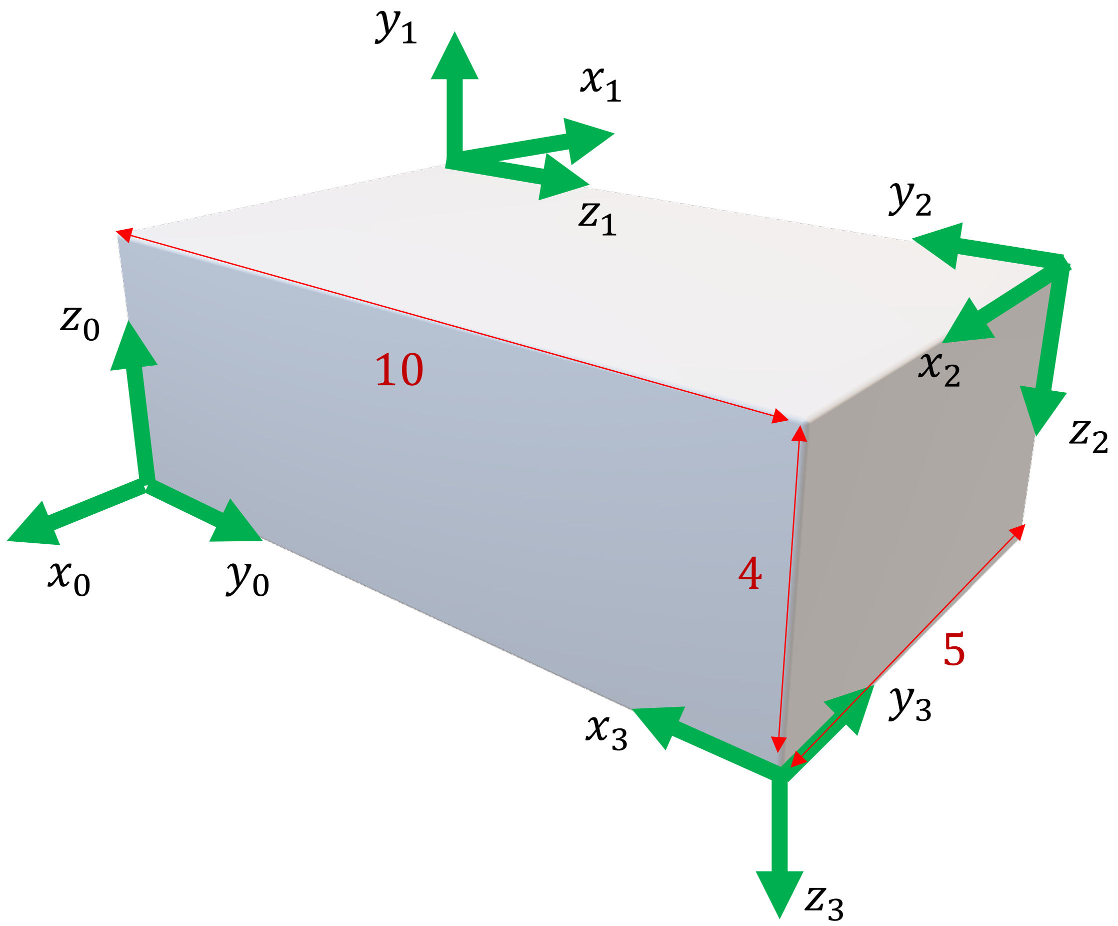


$$R_1^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$T_1^0 = \begin{bmatrix} -1 & 0 & 0 & -5 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A bunch of examples:

A rectangular solid: all angles are multiples of $\pi/2$.

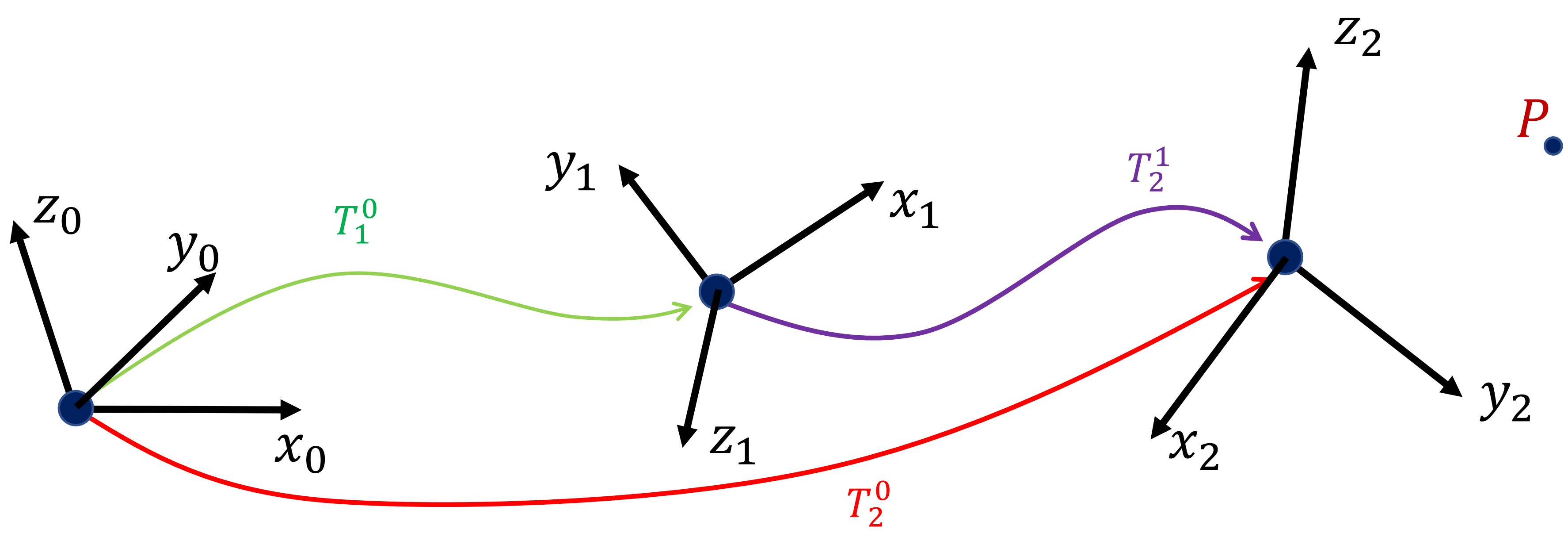


$$R_2^1 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Composition of Transformations

Now, consider the rotation *and the translation!*



$$\tilde{P} = \begin{bmatrix} p_x \\ P_y \\ p_z \\ 1 \end{bmatrix}$$

From our previous results, we know:

$$\left.\begin{aligned} \tilde{P}^0 &= T_1^0 P^1 \\ \tilde{P}^1 &= T_2^1 \tilde{P}^2 \end{aligned}\right\} \Longrightarrow \tilde{P}^0 = T_1^0 T_2^1 \tilde{P}^2$$

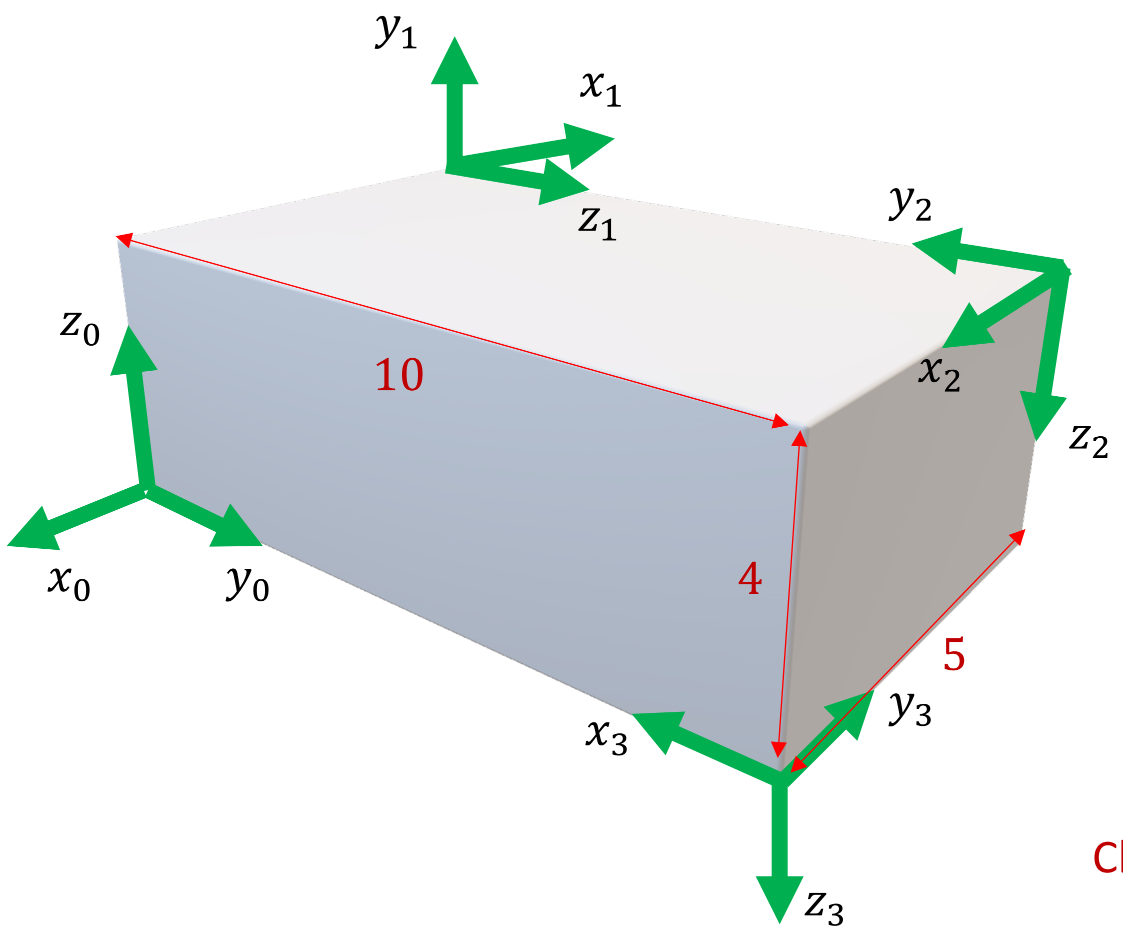
But we also know: $\tilde{P}^0 = T_2^0 \tilde{P}^2$

$$\Longrightarrow T_2^0 = T_1^0 T_2^1$$

<u>This is the composition law for homogeneous transformations.</u>

A bunch of examples:

A rectangular solid: all angles are multiples of $\pi/2$.



$$T_1^0 = \begin{bmatrix} -1 & 0 & 0 & -5 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^0 = \begin{bmatrix} -1 & 0 & 0 & -5 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & -5 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & -1 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Check this by directly determining T_2^0 from the figure… it works!

Inverse of a Homogeneous Transformation

What is the relationship between T_j^i and T_i^j?

In general, $T_k^j = \left(T_j^k\right)^{-1}$ and $\begin{bmatrix} \boldsymbol{R} & \boldsymbol{d} \\ 0_n & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{R}^T & -\boldsymbol{R}^T\boldsymbol{d} \\ 0_n & 1 \end{bmatrix}$

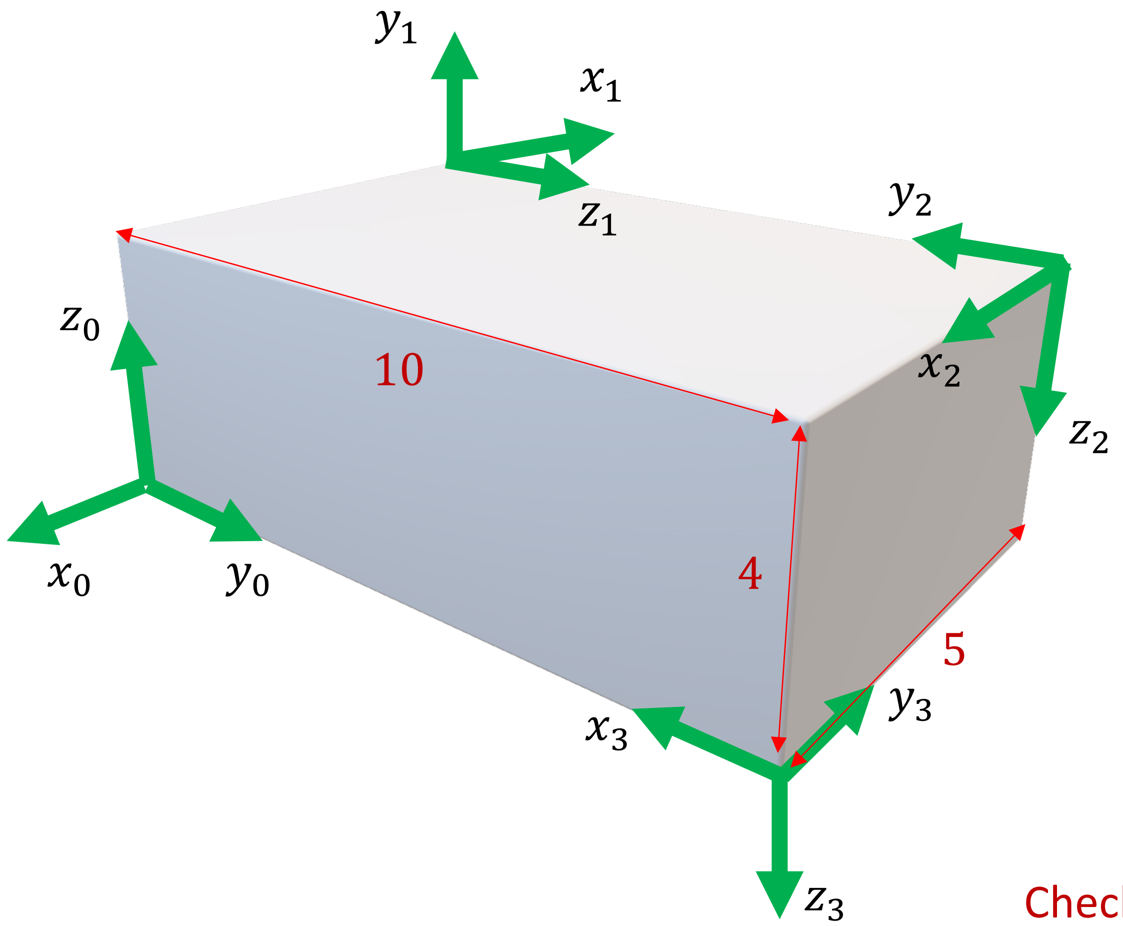
This is easy to verify:

$$\begin{bmatrix} \boldsymbol{R} & \boldsymbol{d} \\ 0_n & 1 \end{bmatrix}\begin{bmatrix} \boldsymbol{R}^T & -\boldsymbol{R}^T\boldsymbol{d} \\ 0_n & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}\boldsymbol{R}^T & -\boldsymbol{R}\boldsymbol{R}^T\boldsymbol{d} + \boldsymbol{d} \\ 0_n & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{n\times n} & \boldsymbol{0}_n \\ 0_n & 1 \end{bmatrix} = I_{(n+1)\times(n+1)}$$

A bunch of examples:

A rectangular solid: all angles are multiples of $\pi/2$.

$$T_2^0 = \begin{bmatrix} 1 & 0 & 0 & -5 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & -1 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

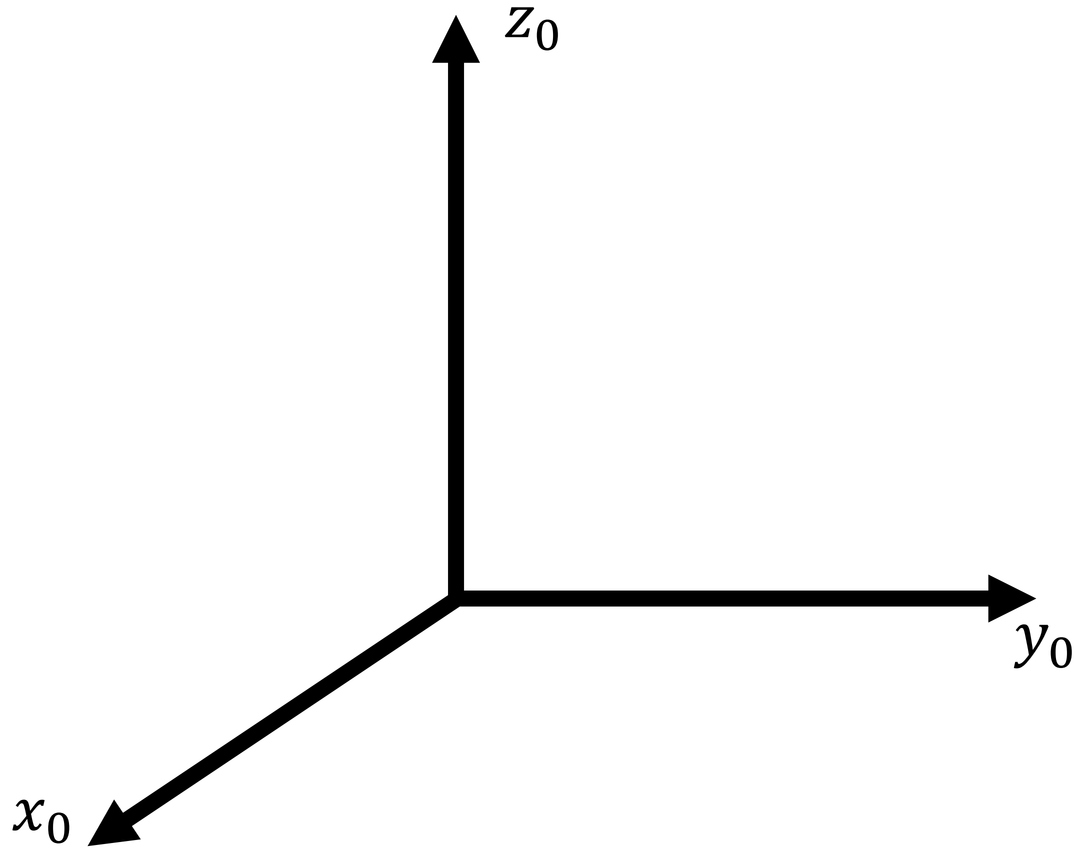


$$(T_2^0)^{-1} = \begin{bmatrix} \boldsymbol{R}^T & -\boldsymbol{R}^T\boldsymbol{d} \\ 0_n & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & -1 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^0(T_2^0)^{-1} = \begin{bmatrix} 1 & 0 & 0 & -5 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & -1 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & -1 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

Check this by directly determining T_0^2 from the figure… it works!

Rotations about Coordinate Axes



$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

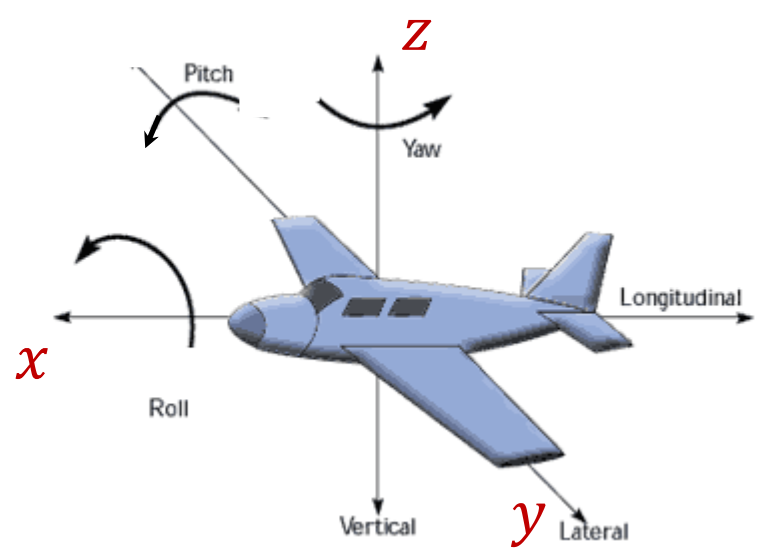
$$R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_{z,\psi} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Roll, Pitch, and Yaw

When we parameterize the rotation matrices using $R = R_{z,\psi} R_{y,\theta} R_{x,\phi}$, the angles are called roll, pitch, and yaw:

- Yaw ψ is a rotation about the world's z-axis.
- Pitch θ is a rotation about the plane's y-axis (note, this axis moves as a function of the yaw angle).
- Roll ϕ is a rotation about the plane's x-axis (note, this axis moves as a function of the yaw angle and the pitch angle).



This coordinate frame assignment is known as *Forward-Left-Up (FLU).*

- The x-axis is the Forward direction.
- The y-axis points to the Left.
- The z-axis points Up.

Parameterization of 3D Rotations

- Consider the three successive rotations: $R = R_{z,\psi} R_{y,\theta} R_{x,\phi} = R_{yaw} R_{pitch} R_{roll}$

$$R_{z,\psi} R_{y,\theta} R_{x,\phi} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

$$= \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix}$$

Parameterization of 3D Rotations

Any rotation matrix can be expressed in this form!

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix}$$

1. Solve for θ using $\mathrm{r}_{31} = -S_\theta$
2. Solve for ϕ using $\mathrm{r}_{32} = C_\theta S_\phi, r_{33} = C_\theta C_\phi$
3. Solve for ψ using $\mathrm{r}_{11} = C_\psi C_\theta, \mathrm{r}_{21} = S_\psi C_\theta$

The function $ATAN2(y, x)$ returns the angle whose tangent is y/x, in the appropriate quadrant. Thus:

$$\phi = ATAN2(r_{32}, r_{33})$$
$$\psi = ATAN2(r_{21}, r_{11})$$

We can parameterize SO(3) using these three angles, ϕ, θ, ψ.

Singularities for Roll, Pitch, Yaw

- For Roll, Pitch, and Yaw, when $S_\theta = 1$, we have:

$$R_{z,\psi}R_{y,\theta}R_{x,\phi} = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix}$$

$$= \begin{bmatrix} 0 & C_\psi S_\phi - S_\psi C_\phi & C_\psi C_\phi + S_\psi S_\phi \\ 0 & S_\psi S_\phi + C_\psi C_\phi & S_\psi C_\phi - C_\psi S_\phi \\ -1 & 0 & 0 \end{bmatrix}$$

When $S_\theta = 1$, there are infinitely many solutions for ψ and ϕ.
Only the difference $\phi - \psi$ is uniquely determined.

$\longrightarrow$

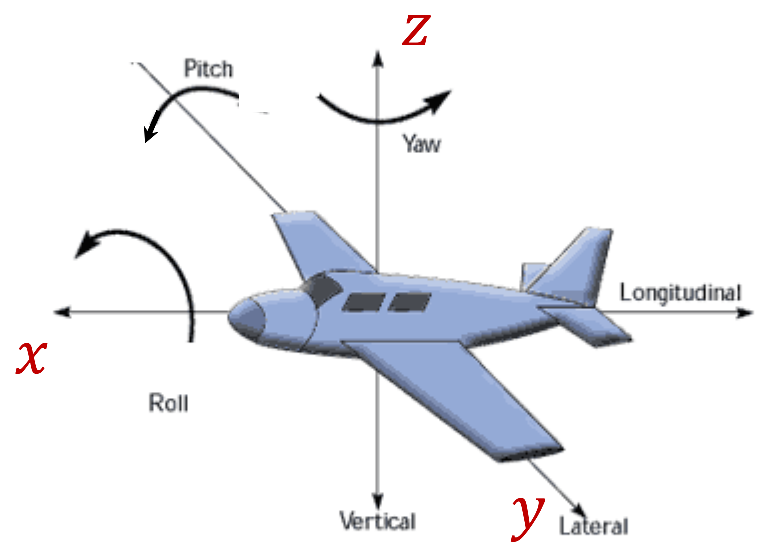
So, this is only a local parameterization.
It works when $r_{31} \neq \pm 1$.

Singularities for Roll, Pitch, Yaw

Things break down when $\theta = \pm\frac{\pi}{2}$ --- but this makes sense!

If $\theta = \pm\frac{\pi}{2}$ then the planes x-axis is aligned with the world's z-axis.

In this case, roll and yaw are rotations about the same axis!



- Roll, Pitch, and Yaw are useful when the plane is roughly horizontal.
- If the plane tips completely up or completely down (i.e., $\theta = \pm\frac{\pi}{2}$), things have already gone very wrong, so it's not such a big deal that the parameterization breaks down for this case.

This coordinate frame assignment is known as *Forward-Left-Up (FLU).*

- The x-axis is the Forward direction.
- The y-axis points to the Left.
- The z-axis points Up.

Other Parameterization of 3D Rotations

Consider the three successive rotations: $R = R_{a,\phi} R_{b,\theta} R_{c,\psi}$
where a, b, c denote coordinate axes and $a \neq b, b \neq c$.

There are lots of possibilities!

$R_{z,\psi} R_{y,\theta} R_{z,\phi}$

$R_{x,\psi} R_{y,\theta} R_{z,\phi}$

$R_{x,\psi} R_{y,\theta} R_{x,\phi}$

$R_{y,\psi} R_{z,\theta} R_{x,\phi}$

$\vdots$

$R_{z,\psi} R_{x,\theta} R_{z,\phi}$

- These are generically referred to as ***Euler angles***.
- Each set of Euler angles admits a local parameterization of $SO(3)$.
- Like Roll-Pitch-Yaw, each Euler angle parameterization is a *local* parameterization, and has problems for certain configurations.
- The configurations where things break down are referred to as ***singularities***.
- These singularities are the reason Oculus was successful in the VR headset market. Previous designs often used Euler angle parameterizations of rotation. Looking directly up caused the display to spin wildly.
- ***The z-y-z Euler angles are commonly used to parameterize the orientation of the wrist mechanism of robot manipulators.***

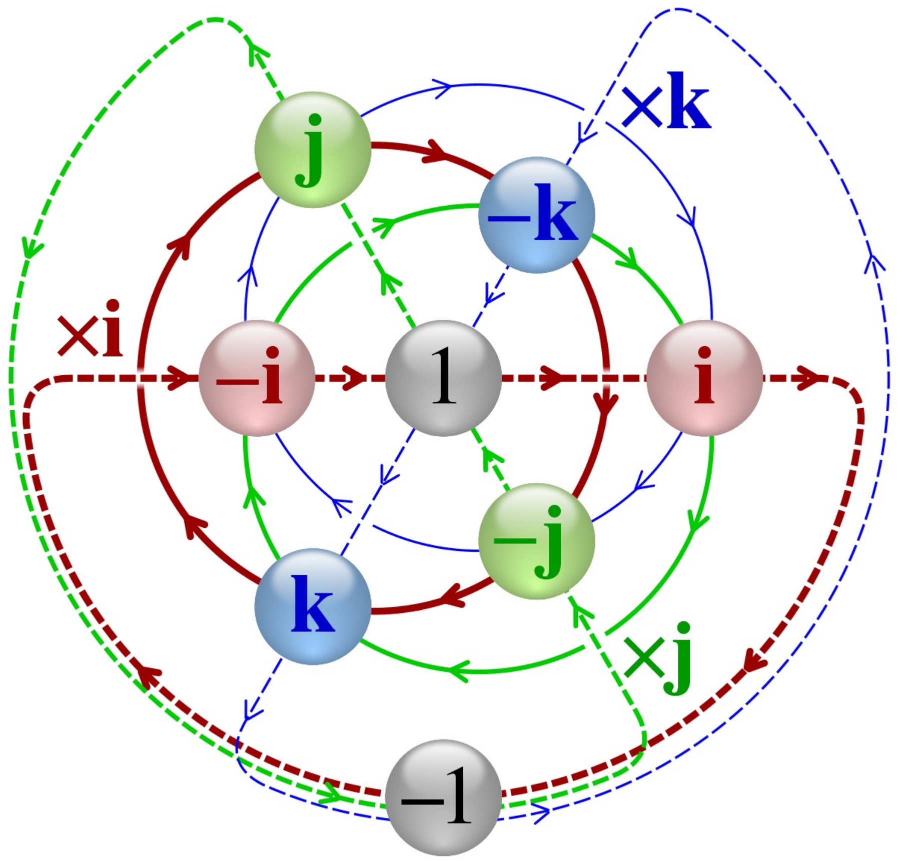
Other Parameterization of 3D Rotations

- 3x3 matrices
- Quaternions: generalization of complex numbers:
- A rotation with an angle θ around the axis defined by the unit vector:

$$\vec{u} = (u_x, u_y, u_z) = u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k}$$

is represented by:

$$\mathbf{q} = e^{\frac{\theta}{2}(u_x\mathbf{i}+u_y\mathbf{j}+u_z\mathbf{k})} = \cos\frac{\theta}{2} + (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})\sin\frac{\theta}{2}$$



GTSAM can do both internally, chosen via compile-time flag.