MEAM 520

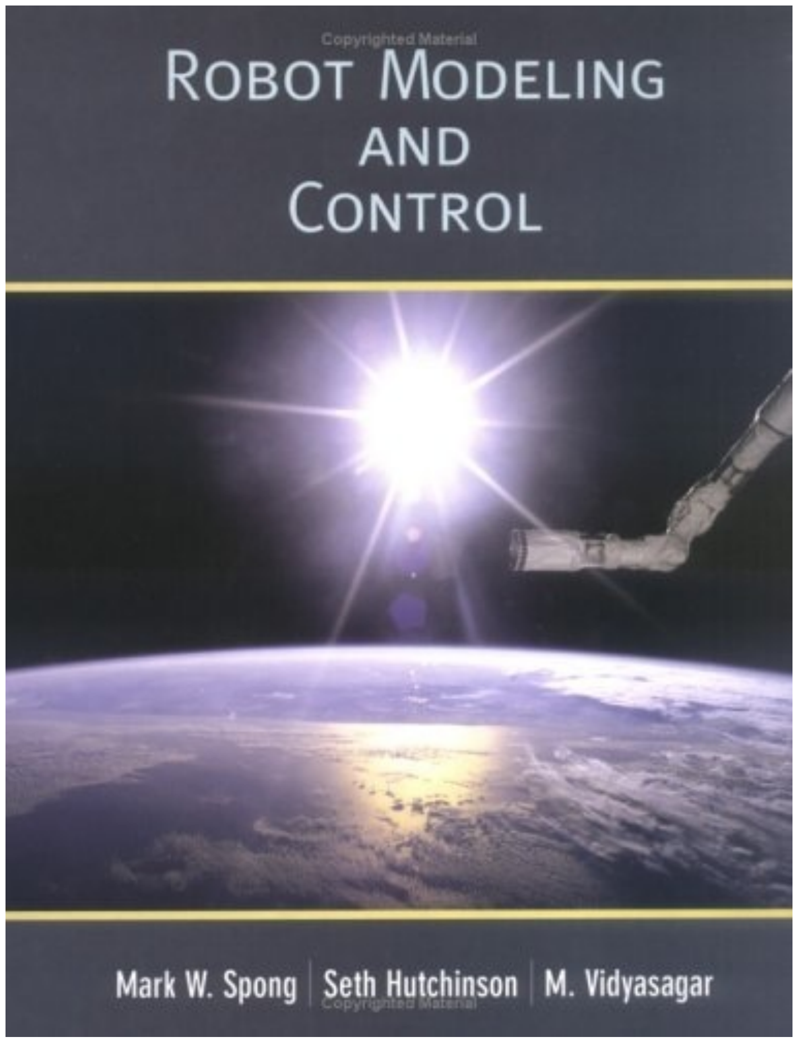
Rotational Parameterizations and Homogeneous Transformations

Katherine J. Kuchenbecker, Ph.D.
Mechanical Engineering and Applied Mechanics Department
SEAS, University of Pennsylvania

GRASP
LABORATORY

Lecture 4: September 9, 2014

Reading



For today,
read Sections 2.4 and 2.5.

Homework

Homework 2:
Robotic Manipulators and Rotation Matrices

MEAM 520, University of Pennsylvania
Katherine J. Kuchenbecker, Ph.D.

September 4, 2013

This paper-based assignment is due on Thursday, September 11, by midnight (11:59:59 p.m.) You should aim to turn it in during class that day. If you don't finish until later in the day, you can turn it in to Professor Kuchenbecker's office, Towne 224, in the assignment submission box or under the door. Late submissions will be accepted until Sunday, September 14, by midnight (11:59:59 p.m.), but they will be penalized by 10% for each partial or full day late, up to 30%. After the late deadline, no further assignments may be submitted.

You may talk with other students about this assignment, ask the teaching team questions, use a calculator and other tools, and consult outside sources such as the Internet. To help you actually learn the material, what you write down must be your own work, not copied from any other individual or a solution manual. Any submissions suspected of violating Penn's Code of Academic Integrity will be reported to the Office of Student Conduct. If you get stuck, post a question on Piazza or go to office hours!

These problems are a mix of custom problems and augmented problems from the textbook, *Robot Modeling and Control* by Spong, Hutchinson, and Vidyasagar (SHV). Please follow the extra clarifications and instructions when provided. Write in pencil, show your work clearly, box your answers, and staple together all pages of your assignment. This assignment is worth a total of 25 points.

1. Custom problem – **Kinematics of Baxter** *(2 points)*
   Rethink Robotics sells a two-armed manufacturing robot named Baxter. Watch YouTube videos of Baxter (e.g., "Baxter Robot Folds a Shirt", "Intera 3 Baxter Software Update Signature Moves") to learn about its kinematics. Draw a schematic of the serial kinematic chain of Baxter's left arm (the one the woman is touching in the picture below.) Use the book's conventions for how to draw revolute and prismatic joints in 3D.

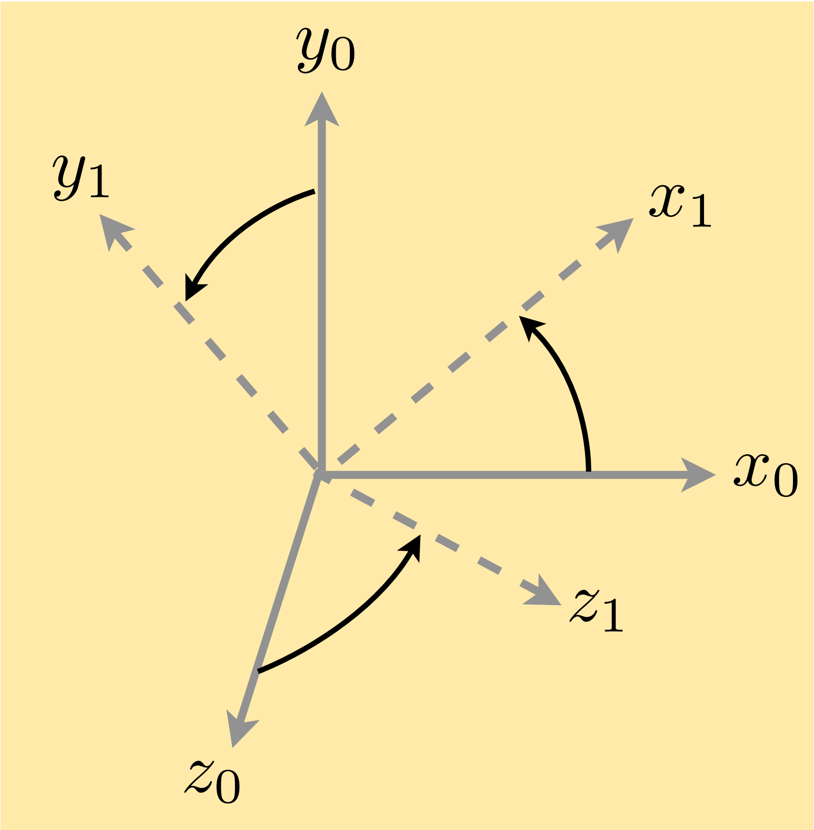
YouTube
Browse
Rethink Robotics: Meet Baxter
RethinkRobotics
Subscribe
1 video

Assignment continued on reverse.

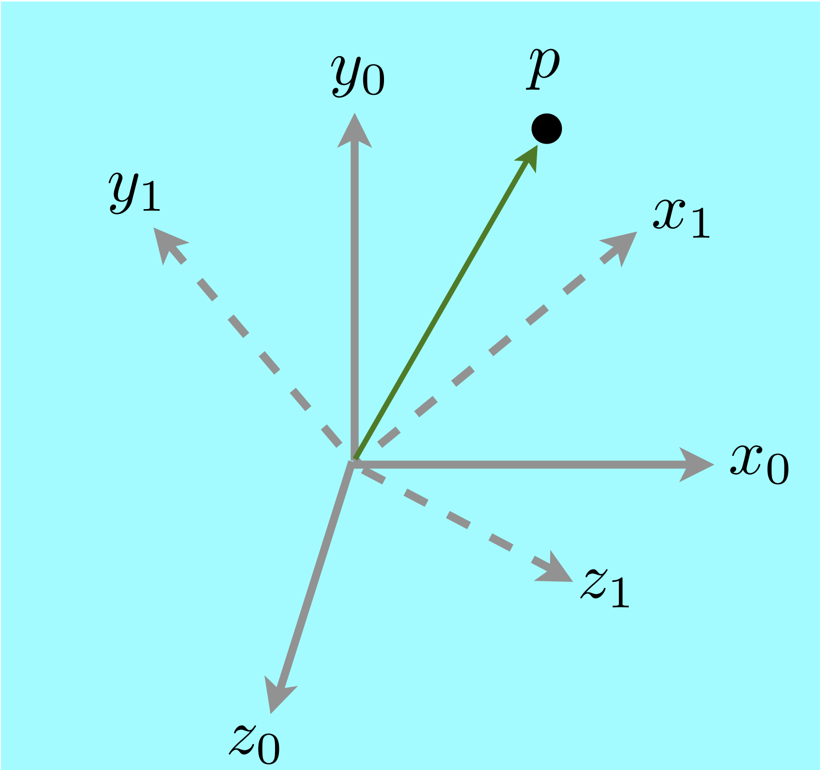
Homework 2 is due by
11:59 p.m. on Thursday.

Interpretations of Rotation Matrices

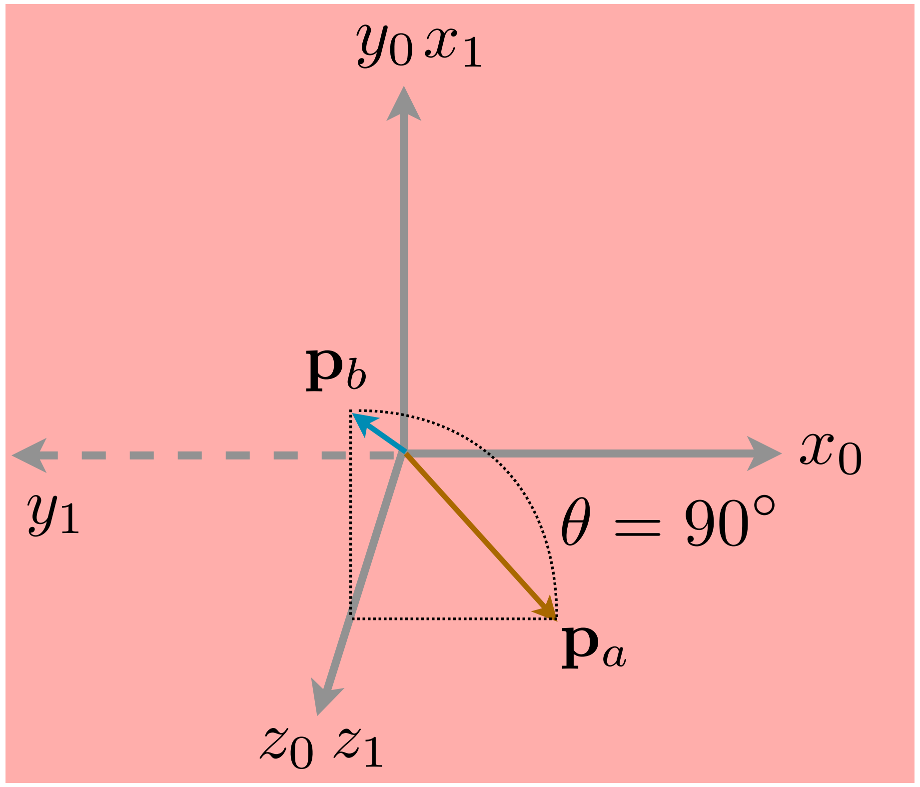
$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$



Orientation of one coordinate frame with respect to another frame



Coordinate transformation relating the coordinates of a point p in two different frames



Operator taking a vector and rotating it to yield a new vector in the same coordinate frame

A tool to help you understand 3D rotation matrices:

visualizeR.m
plotCoordinateFrame.m
plotVector.m

note

stop following 42 views

Actions

Posted: a Matlab script to help you understand rotation matrices in 3D

Dear all,

I created a new Matlab script to help you learn how rotation matrices work in three-dimensional space. It is called visualizeR.m, and it's posted under General Resources here on Piazza. Here is the direct link:
https://piazza.com/class_profile/get_resource/hxuqr6o59c446c/hzt6ljbnrnf7a7

The download is a zip file containing three Matlab scripts. Unzip it and run visualizeR.m. The other two files are helper functions for plotting. All three need to be in the same folder on your computer.

This script will generate a random rotation matrix, R_1^0, and show you all three interpretations, as described by SHV on page 47:

Upper-left plot: Gives the numbers in the 3x3 matrix.

Upper-right plot: Shows the orientation of a transformed coordinate frame (frame 1) with respect to a fixed coordinate frame (frame 0).

Lower-left plot: Shows a coordinate transformation relating the coordinates of a point p in frames 0 and 1, via $p^0 = R_1^0 p^1$

Lower-right plot: Shows an operator taking a vector v_a^0 and rotating it to give a new vector v_b^0 in the same coordinate frame, via $v_b^0 = R_1^0 v_a^0$.

Every time you run the script, you have the option of creating a new random matrix or using the one you already have. Use the "Rotate 3D" tool in the figure window to rotate any of the graphs; they will rotate together. You are also welcome to look at the script to understand how it works and edit it to make it better.

I encourage everyone to download this script and play with it. With practice, you should be able to imagine all three interpretations only by looking at the numbers in the rotation matrix. Start by making sure you under stand the upper-right plot, then move on to the two below.

Please let me know what questions and comments you have!

- KJK

hw2 textbook matlab

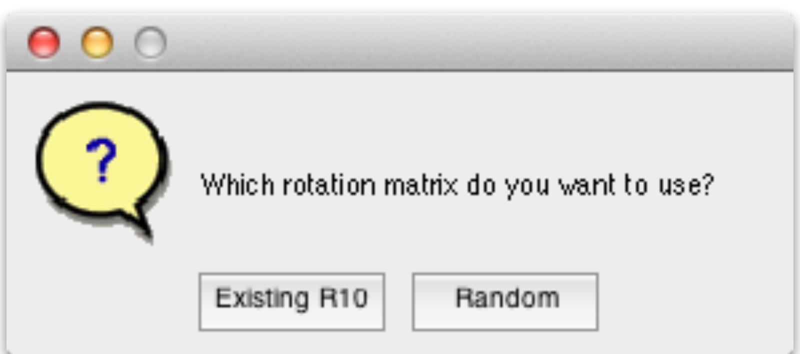
Editor – /Users/kuchenbe/Documents/teaching/meam 520/lectures/04 transformations/visualizeR/visualizeR.m

EDITOR PUBLISH VIEW

visualizeR.m

```matlab
%% Visualize a rotation matrix through three interpretations.
%
% Written by Katherine J. Kuchenbecker, University of Pennsylvania
% For MEAM 520: Introduction to Robotics
% September 2014
% Version 1


%% Find out which matrix to use

% If the matrix already exists, ask if we should use it.
if exist('R10','var')
    button = questdlg('Which rotation matrix do you want to use?','','Existing R10
else
    button = 'Random';
end


%% Create a random three-by-three rotation matrix if needed

if strcmp(button, 'Random')
    % This method was suggested on this MATLAB Central thread:
    % http://www.mathworks.com/matlabcentral/newsreader/view_thread/298500
    % The source credited there is "How to generate random matrices from
    % the classical compact groups", Francesco Mezzadri, Notices of ACM,
    % Volume 54, Number 5, 2007.

    % Create a 3 x 3 matrix of normally distributed pseudorandom numbers.
    random_matrix = randn(3);
```

script | Ln 6 | Col 12


Which rotation matrix do you want to use?
Existing R10
Random

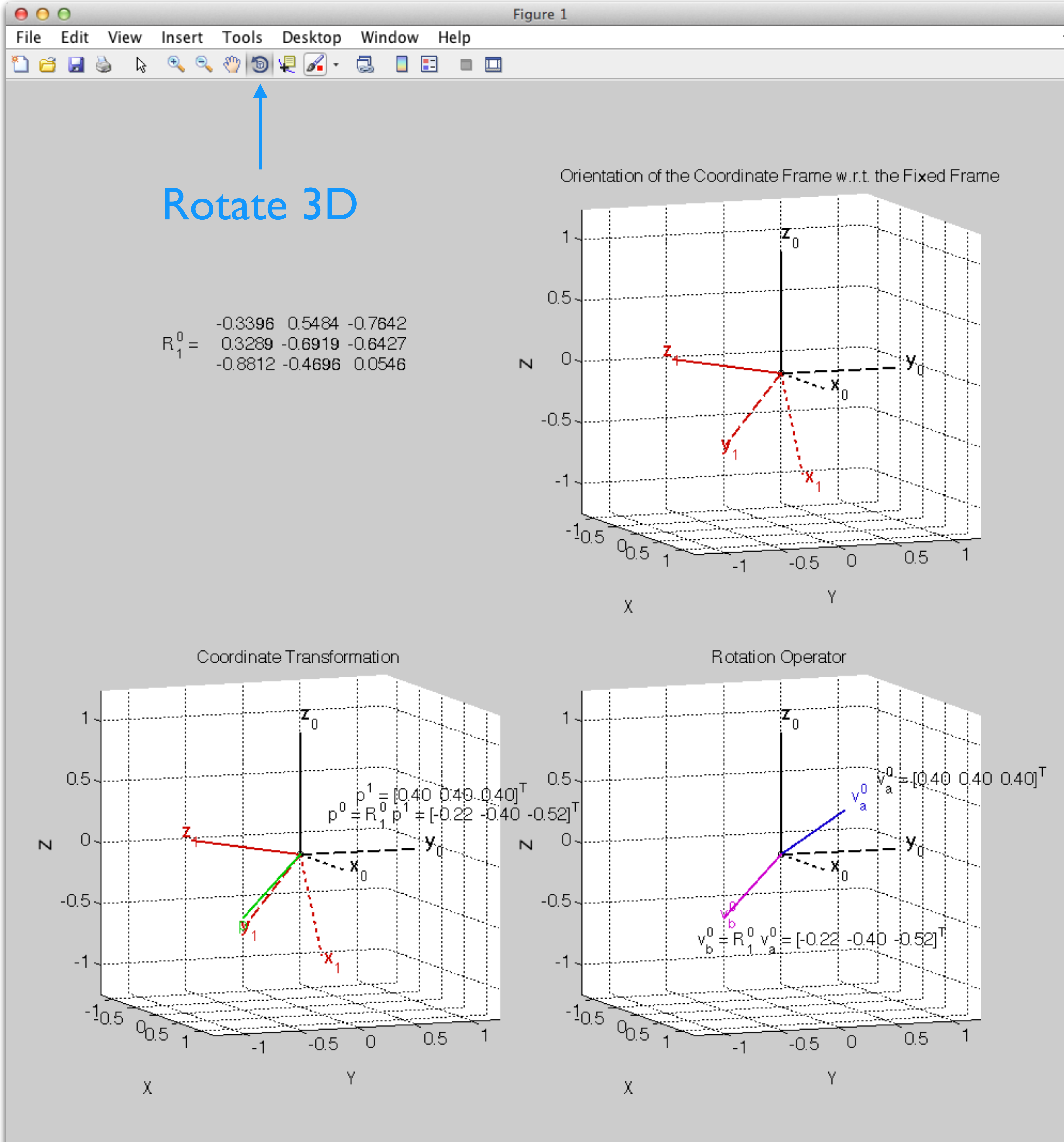
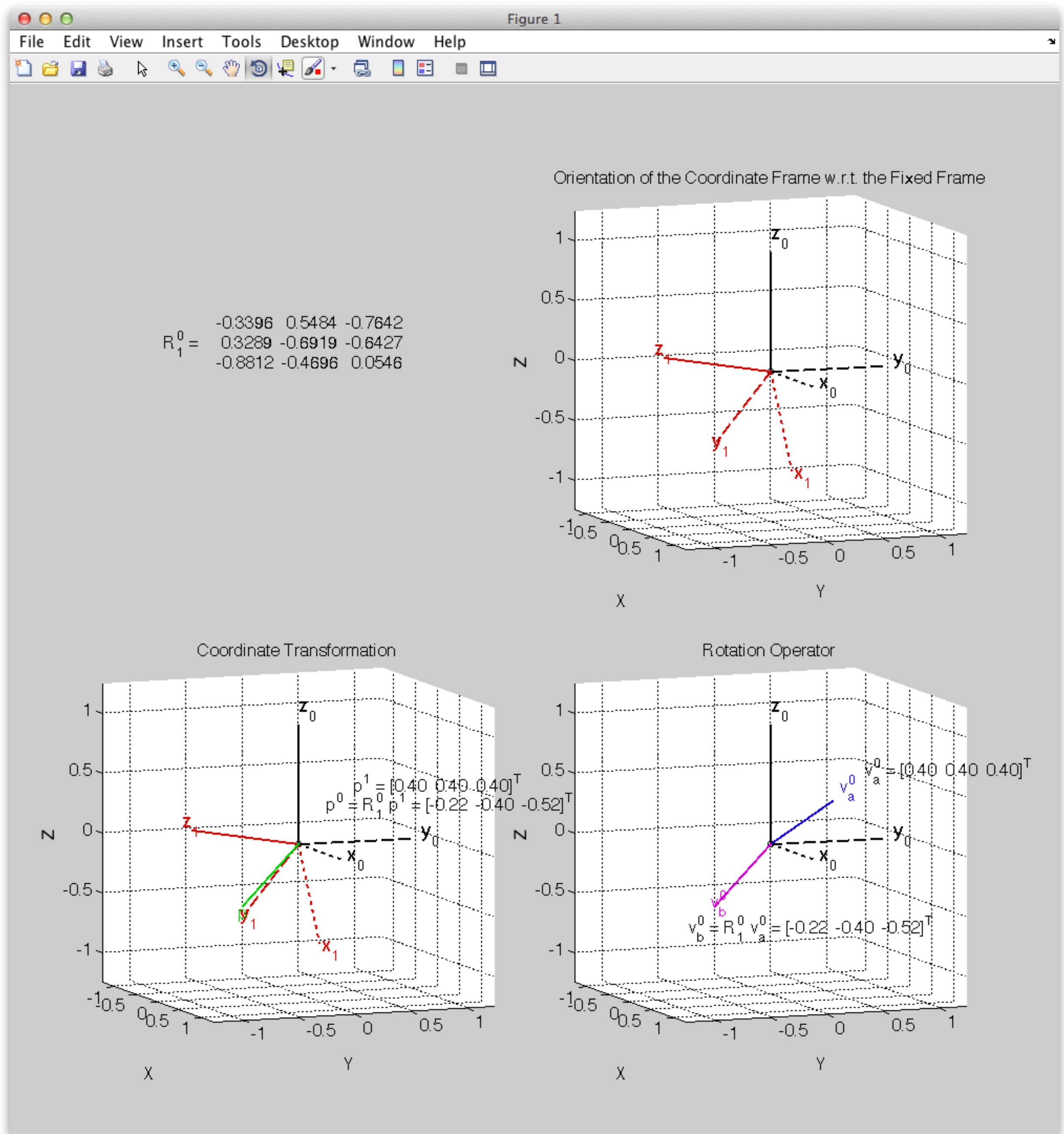


Figure 1
File Edit View Insert Tools Desktop Window Help
Rotate 3D
R =
-0.3396 0.5484 -0.7642
0.3289 -0.6919 -0.6427
-0.8812 -0.4696 0.0546
Orientation of the Coordinate Frame w.r.t. the Fixed Frame
Coordinate Transformation
Rotation Operator
X
Y
Z

What questions do you have?



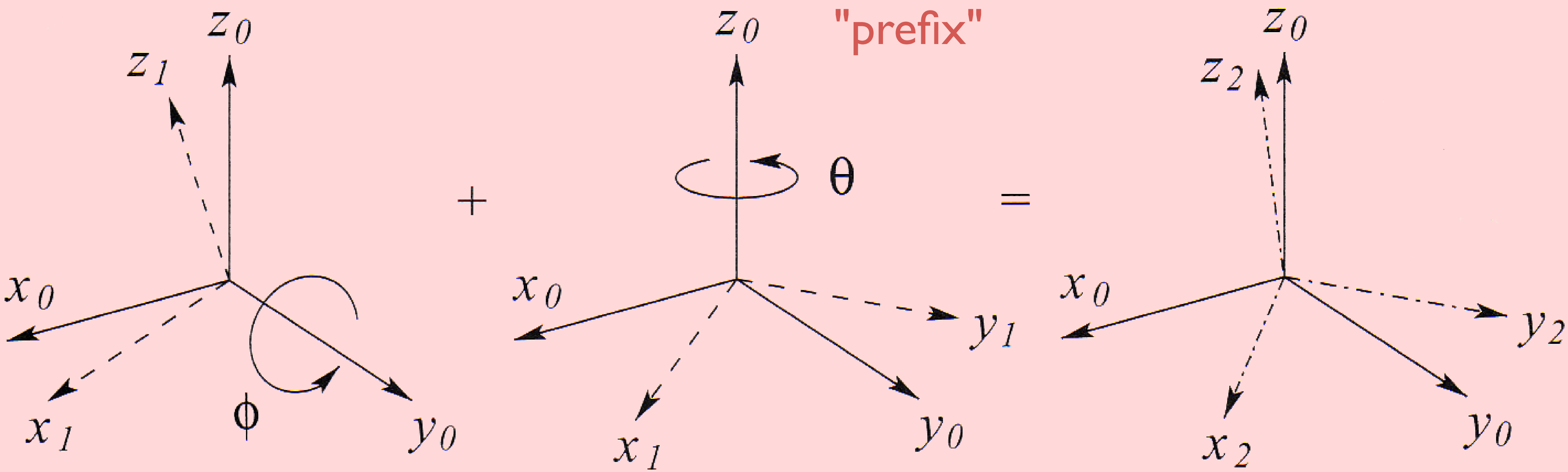
$$\mathbf{R}_1^0$$

$$p^0 = \mathbf{R}_1^0 \, p^1$$

$$v_b^0 = \mathbf{R} \, v_a^0$$

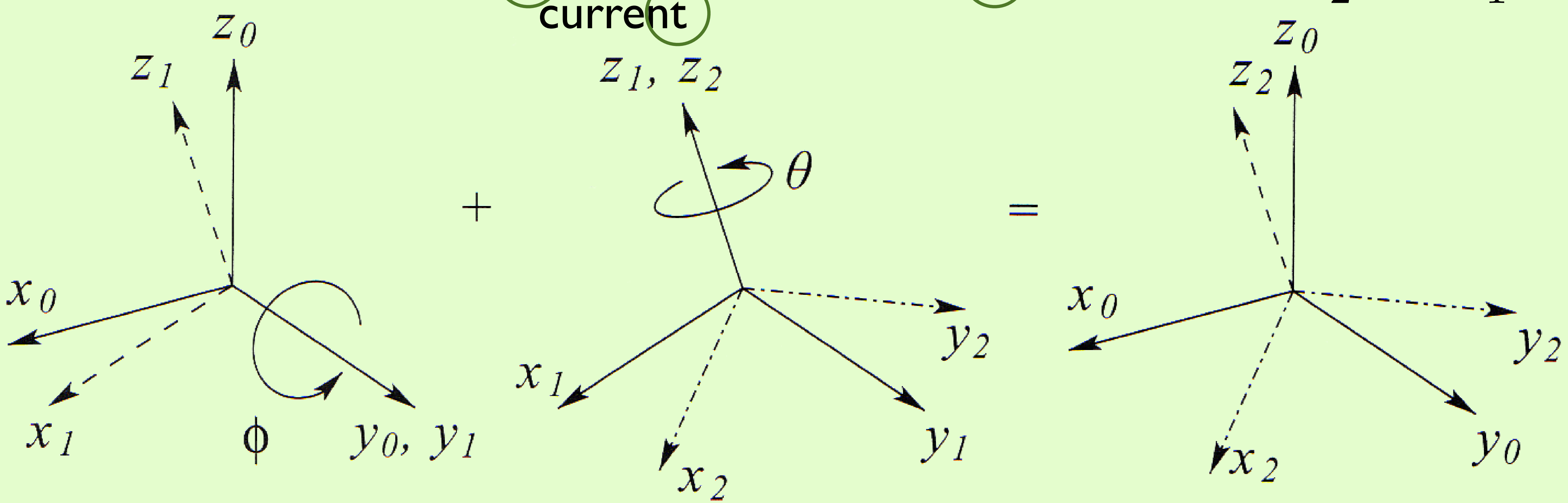
successive rotation about fixed frame? **pre-multiply** $\mathbf{R}_2^0 = \mathbf{R}\,\mathbf{R}_1^0$

"prefix"

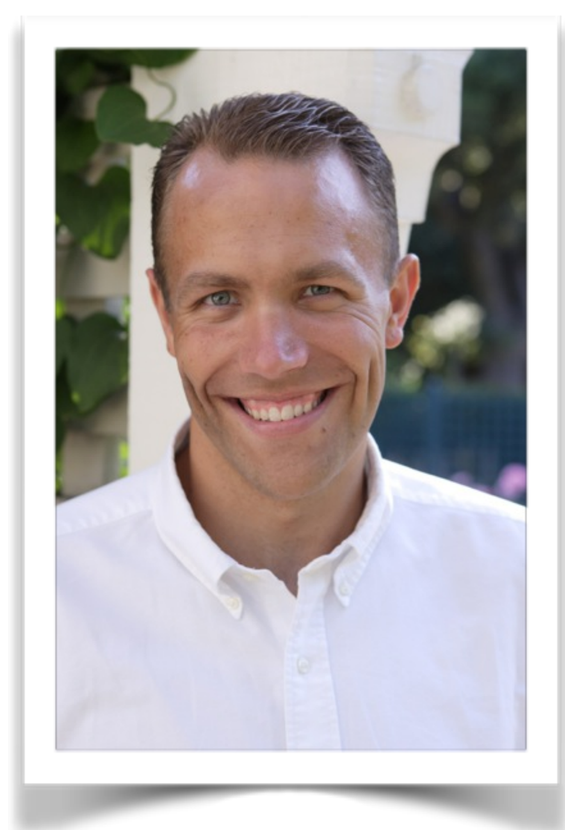


successive rotation about intermediate frame? **post-multiply** $\mathbf{R}_2^0 = \mathbf{R}_1^0\,\mathbf{R}_2^1$

current



Many of today's slides are adapted from ones created by Jonathan Fiene for MEAM 520 in Spring 2012.



Parameterizing Rotations

SHV 2.5

Parameterization of Rotations

$$\mathbf{R}_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix}$$

In three dimensions, no more than 3 independent values are needed to specify an arbitrary rotation.

Thus, the 9-element rotation matrix has 6 redundancies.

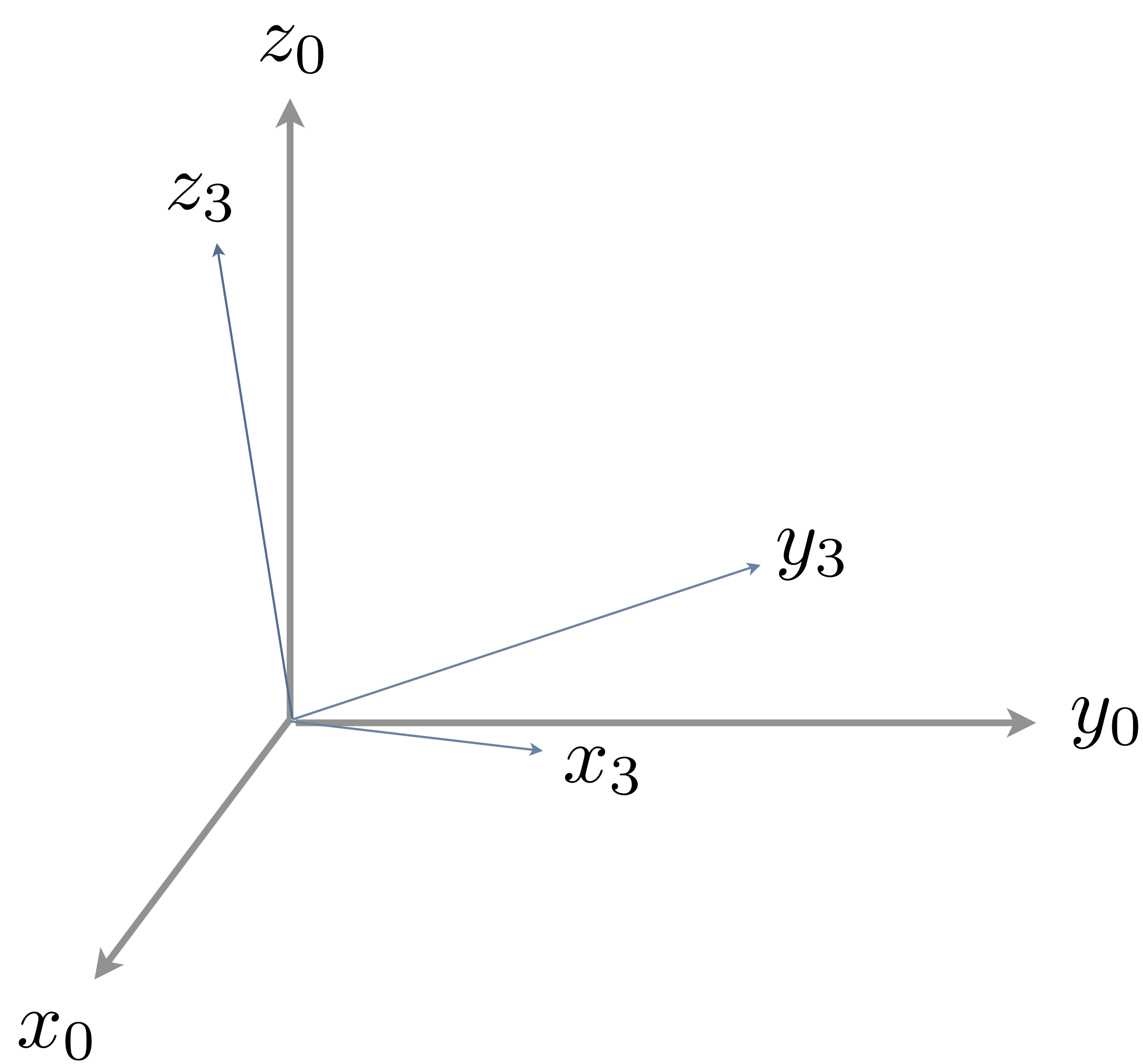
Numerous methods have been developed to represent rotation/orientation more efficiently.

Three common examples are the following:

Euler Angles | Roll, Pitch, Yaw Angles | Axis/Angle Representation

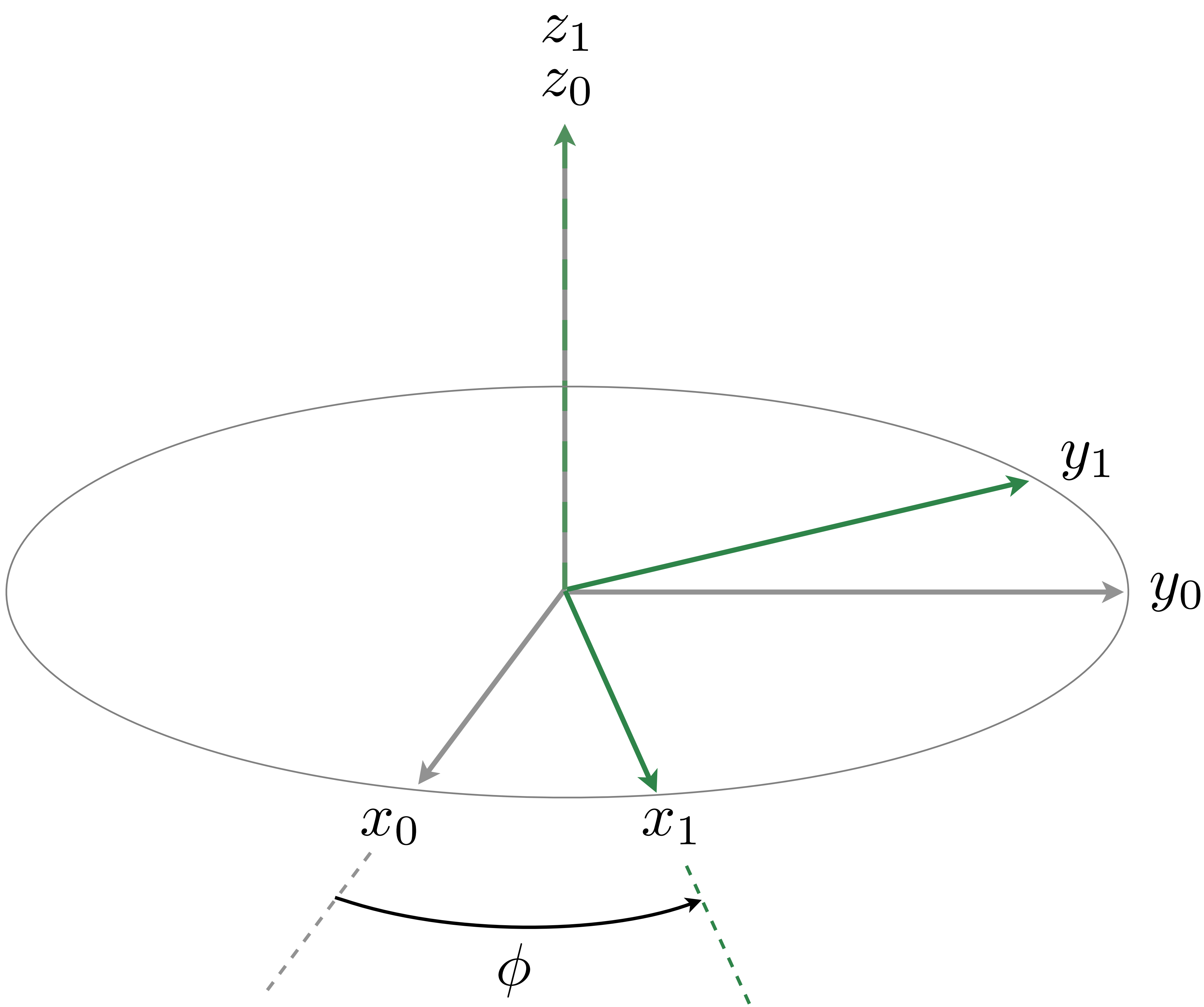
Conventions vary, so always check definitions!

Define a set of three **intermediate** angles, ϕ, θ, ψ , to go from $0 \rightarrow 3$

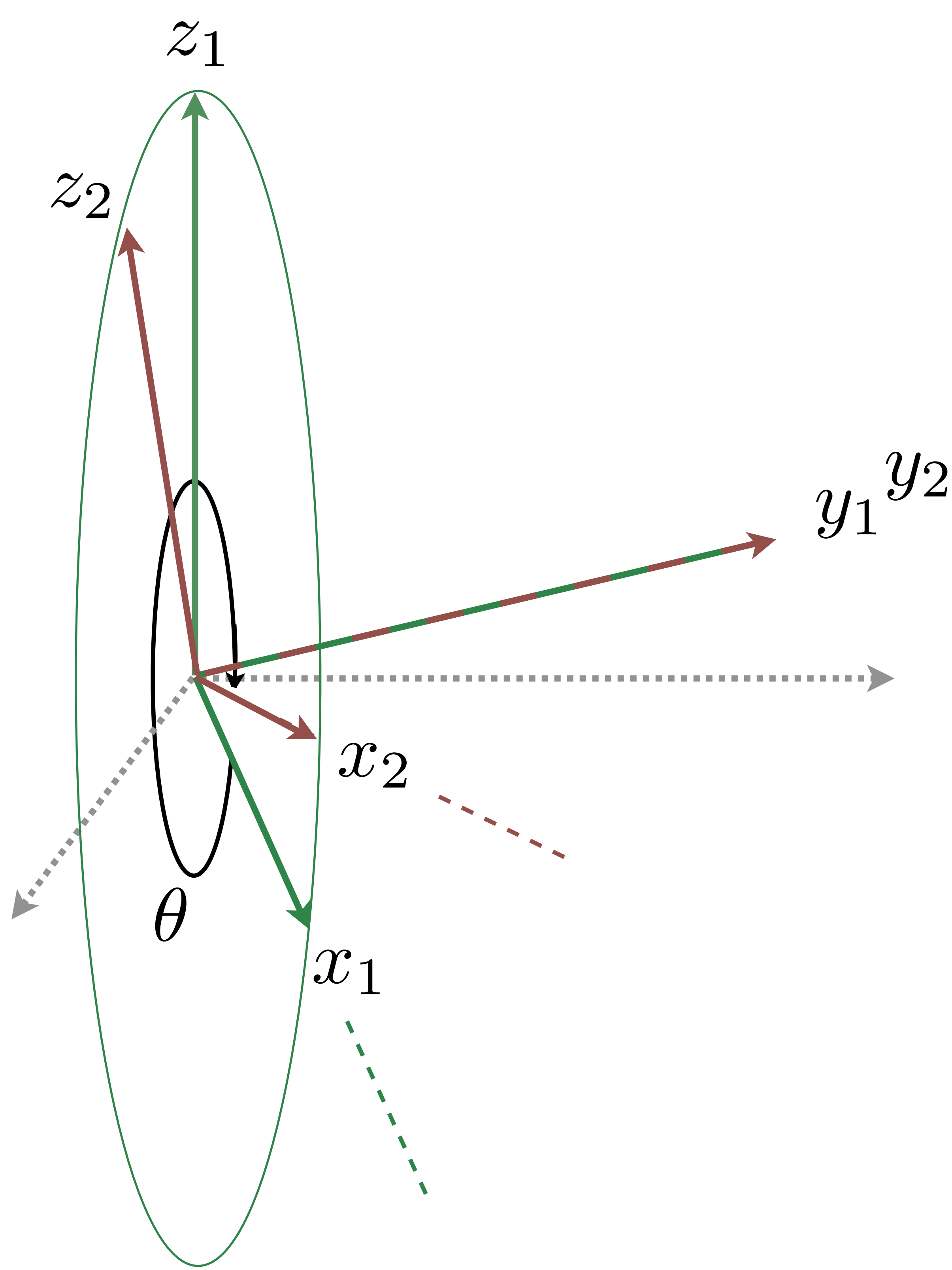


Our book uses a Z-Y-Z convention for Euler angles.

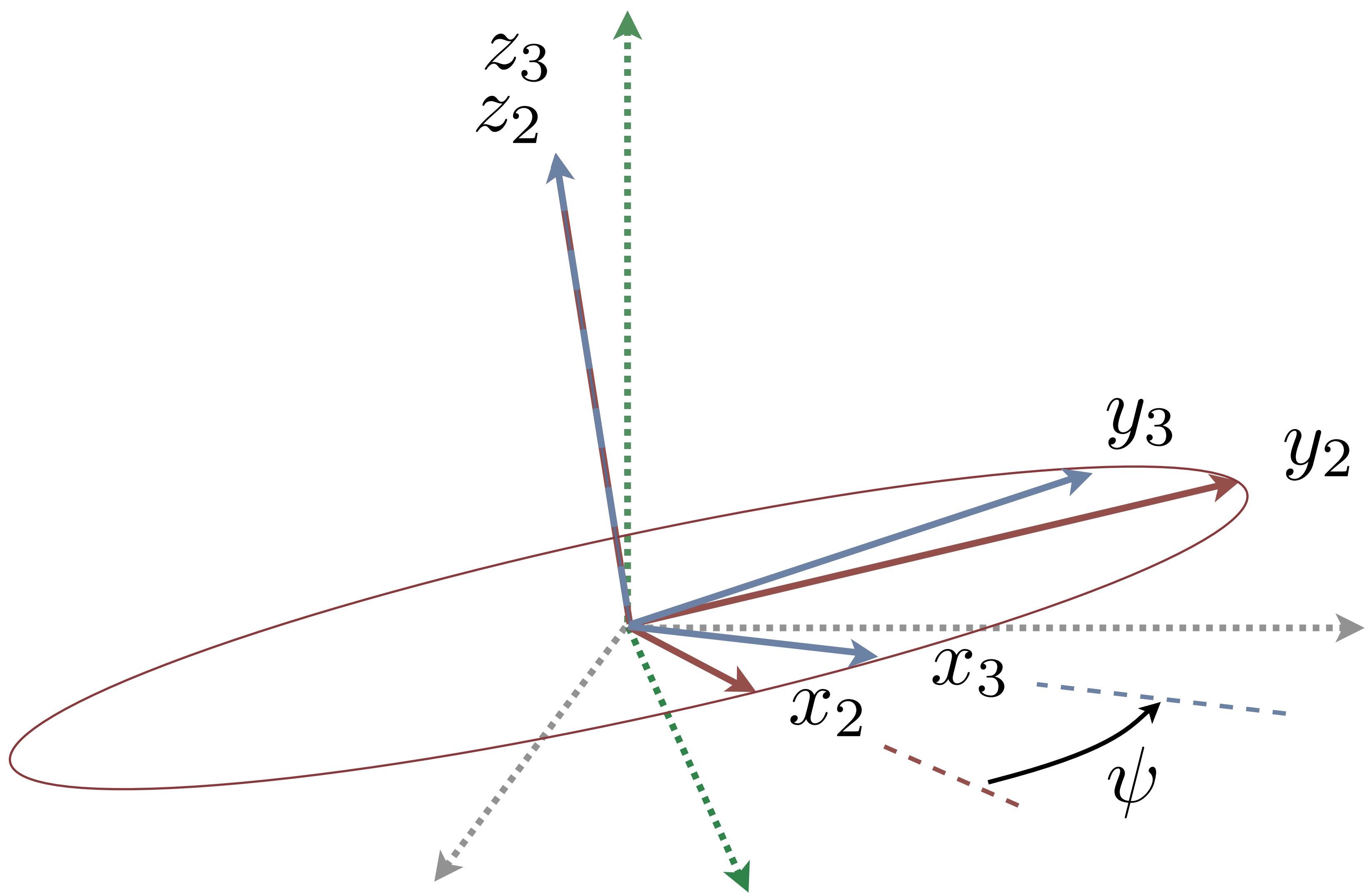
step 1: rotate by ϕ about z_0



step 2: rotate by θ about y_1

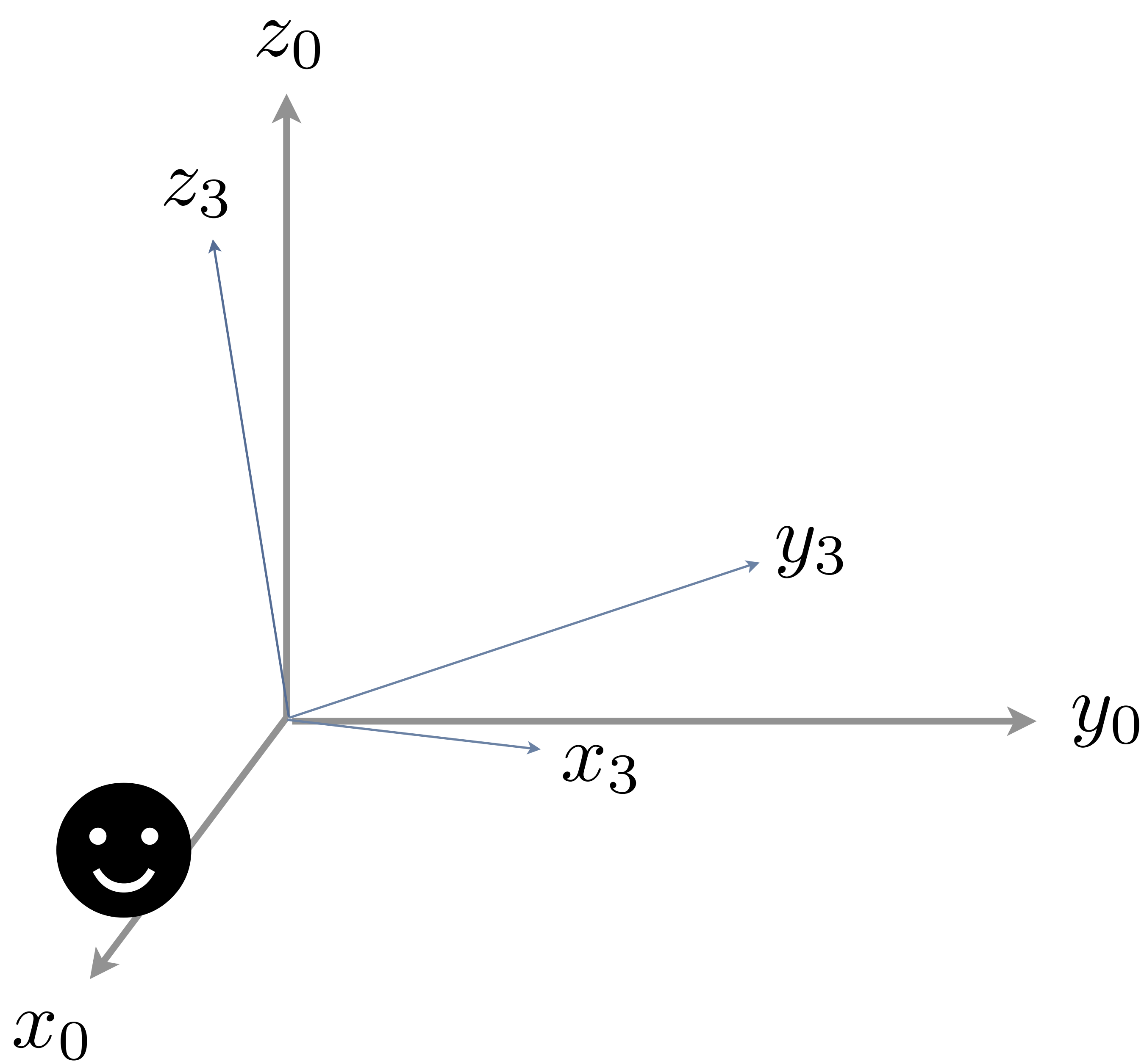


step 3: rotate by ψ about z_2



Euler Angles

Define a set of three **intermediate** angles, ϕ, θ, ψ , to go from $0 \rightarrow 3$



Think about looking out the x-axis with your head, looking left/right, then tilting up/down, then spinning your head around its long axis.

Should we pre- or post-multiply the successive rotations?

Euler Angles to Rotation Matrix

(**post**-multiply using the **basic rotation matrices**)

$$\mathbf{R} = \mathbf{R}_{z,\phi}\,\mathbf{R}_{y,\theta}\,\mathbf{R}_{z,\psi}$$

$$s_\theta = \sin\theta$$
$$c_\theta = \cos\theta$$

$$= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix}$$

Rotation Matrix to Euler Angles?

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

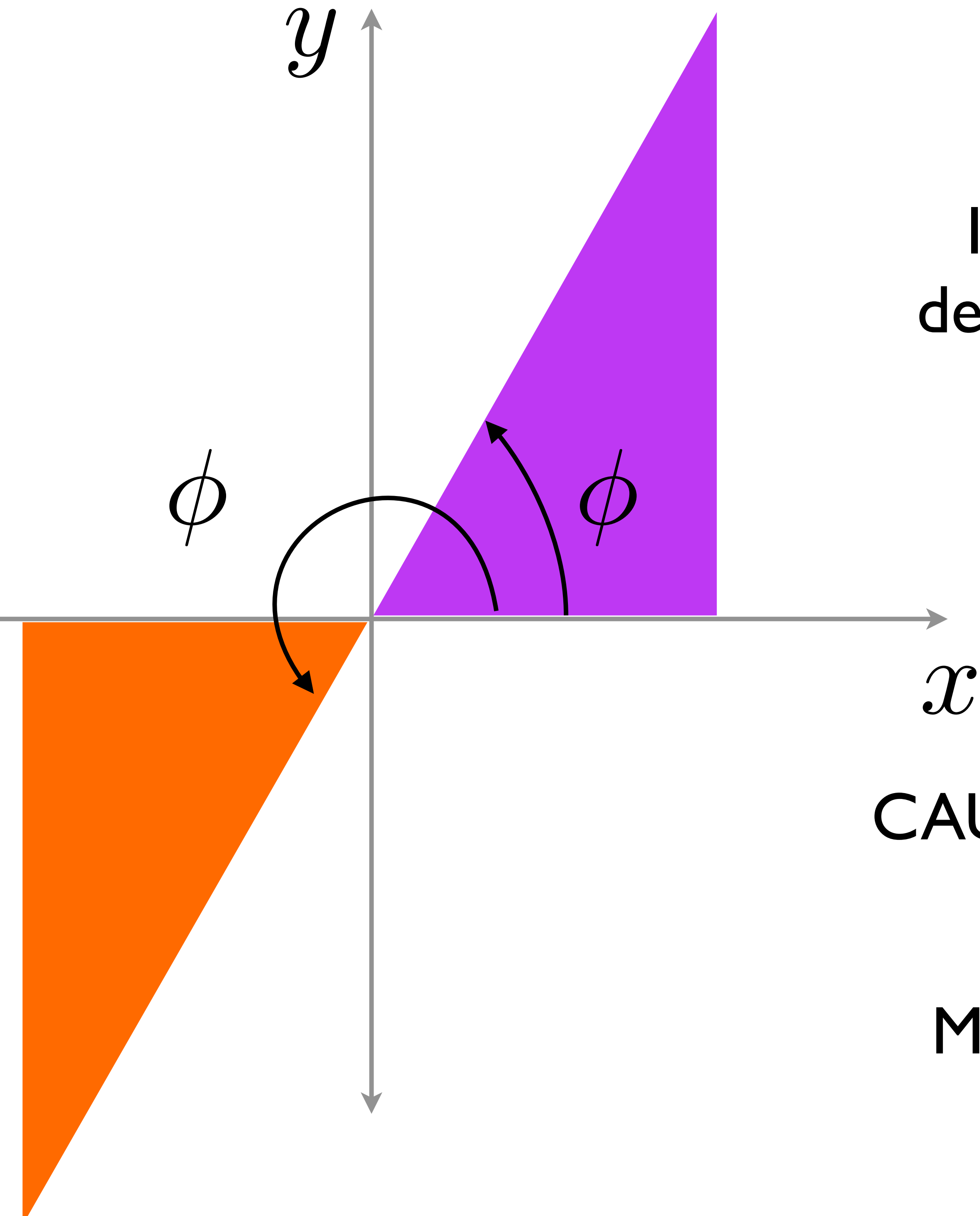
Use atan2 to determine ϕ for both θ options

$$= \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix}$$

Use atan2 to determine ψ for both θ options

Two solutions for θ because sign of s_θ is not known.

$$\phi = ? \quad \theta = ? \quad \psi = ?$$



atan2 is the two-argument inverse tangent function.

It preserves the signs of the numerator and denominator, returning the angle in the correct quadrant, instead of in only two quadrants.

Read Appendix A.1 to learn more.

CAUTION: Our book lists atan2 arguments in the opposite order of MATLAB.

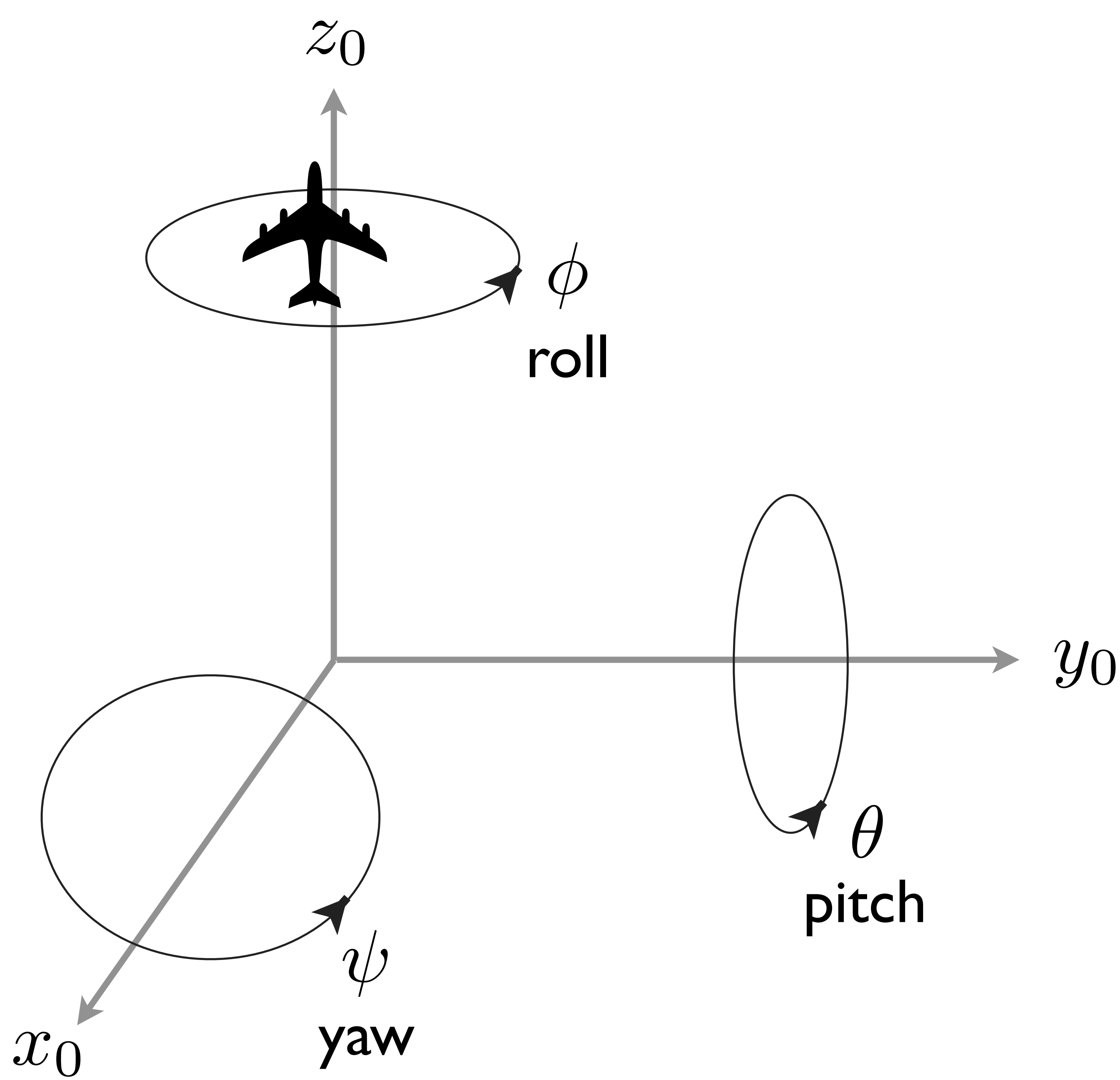
MATLAB expects atan2(num, den) or similarly atan2(y, x).

SHV uses atan2(den, num) or similarly atan2(x, y).

I recommend writing atan2(num/den).

Roll, Pitch, Yaw Angles

a set of three angles that define rotations about **fixed** axes



Our book uses an X-Y-Z convention for Yaw, Pitch, Roll angles.

Think about being a plane flying in the z-axis direction. Yaw is turning left/right, pitch is tilting up/down, and roll rotates around travel direction.

Should we pre- or post-multiply the successive rotations?

Roll, Pitch, Yaw Angles to Rotation Matrices

(**pre-**multiply using the **basic rotation matrices**)

$$\mathbf{R} = \mathbf{R}_{z,\phi}\, \mathbf{R}_{y,\theta}\, \mathbf{R}_{x,\psi}$$

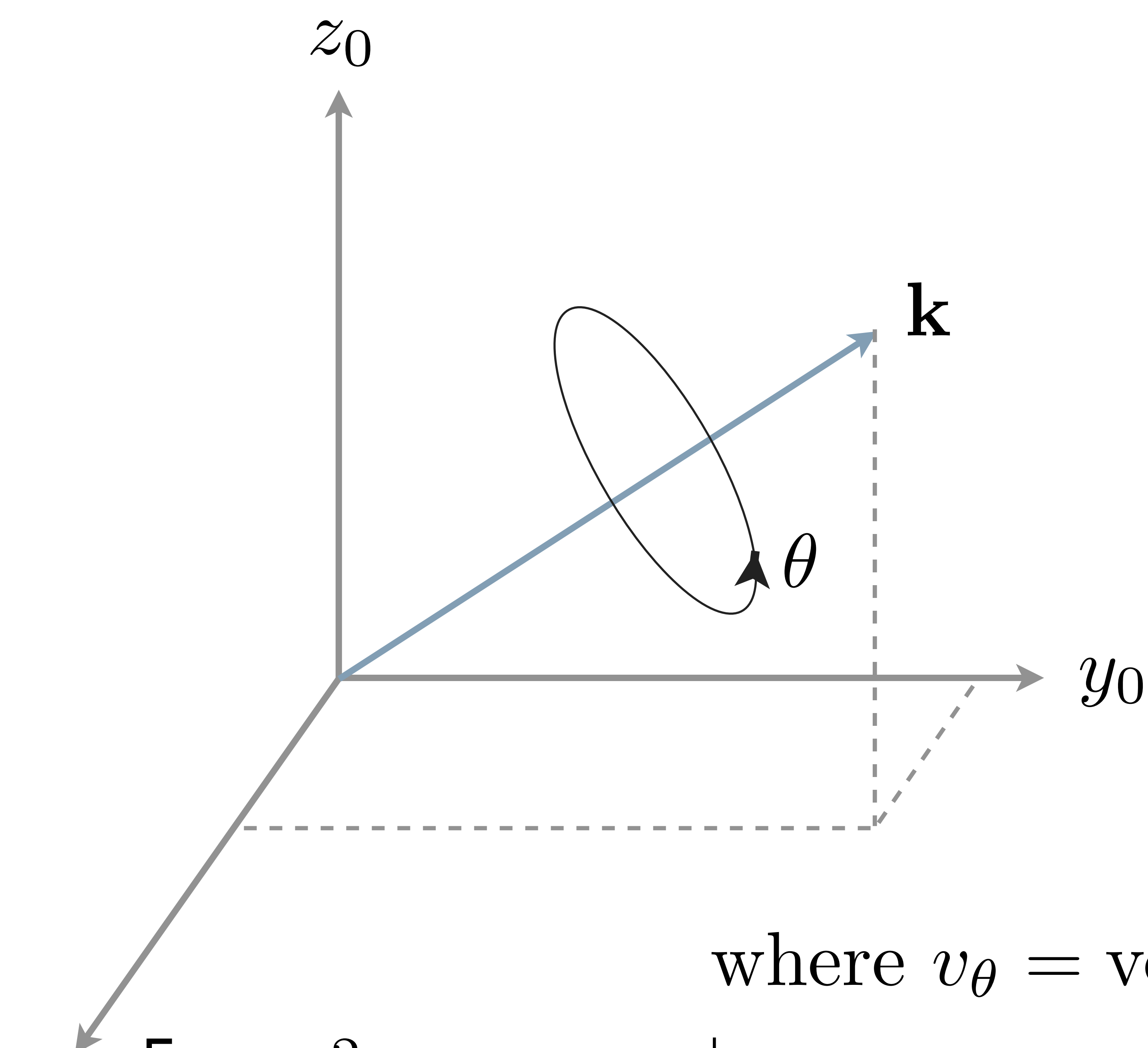
$$= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi & c_\psi \end{bmatrix}$$

$$= \begin{bmatrix} c_\phi c_\theta & c_\phi s_\theta s_\psi - s_\phi c_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ s_\phi c_\theta & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix}$$

You can convert from a rotation matrix to these angles in a manner similar to the procedure for Euler angles.

Axis/Angle Representation

rotation by an angle about an axis in space



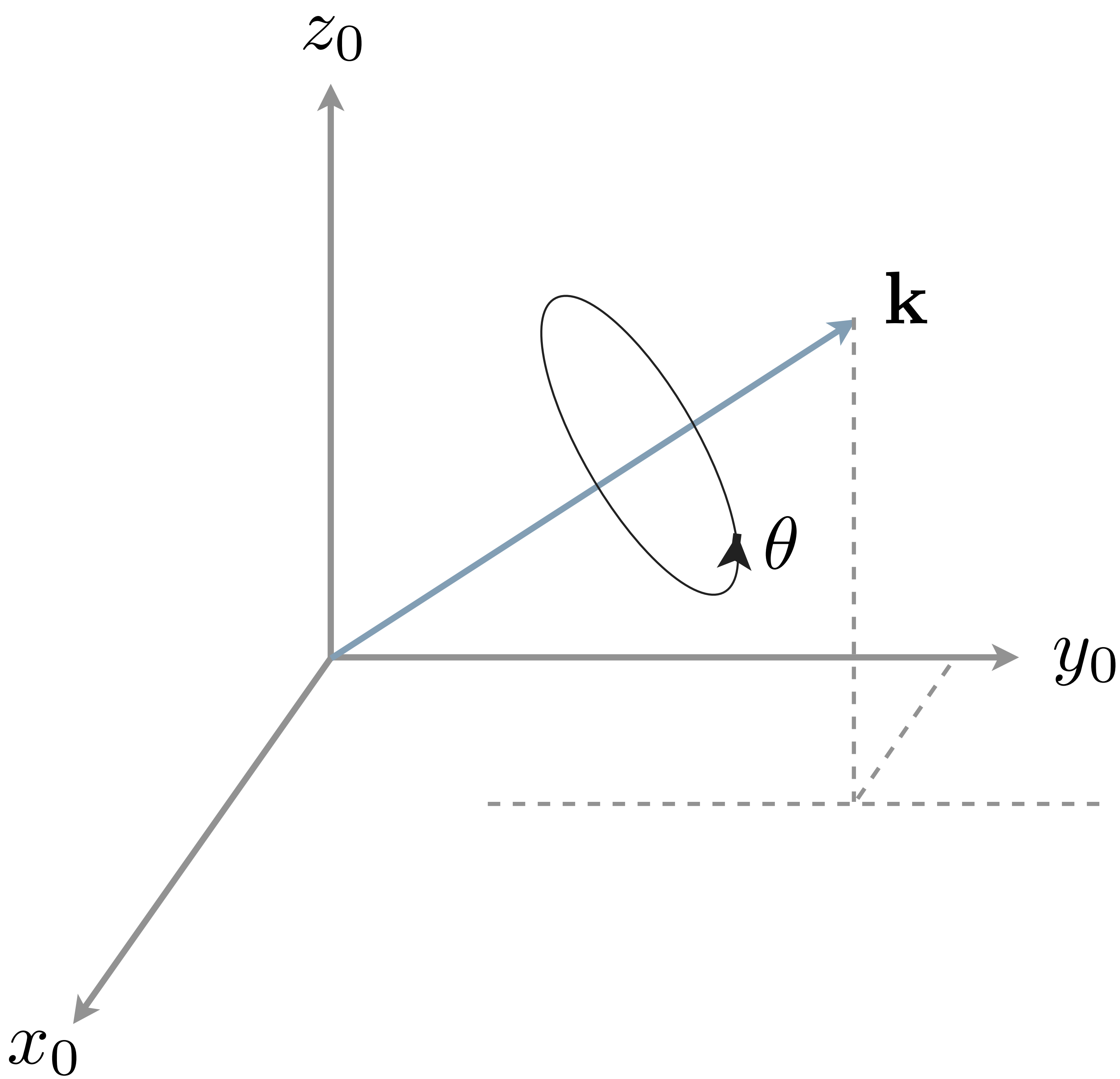
$$\mathbf{k} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}$$

$$k_x^2 + k_y^2 + k_z^2 = 1$$

$$\text{where } v_\theta = \operatorname{vers}\theta = 1 - \cos\theta$$

$$R_{k,\theta} = \left[\begin{array}{c|c|c} k_x^2 v_\theta + c_\theta & k_x k_y v_\theta - k_z s_\theta & k_x k_z v_\theta + k_y s_\theta \\ k_x k_y v_\theta + k_z s_\theta & k_y^2 v_\theta + c_\theta & k_y k_z v_\theta - k_x s_\theta \\ k_x k_z v_\theta - k_y s_\theta & k_y k_z v_\theta + k_x s_\theta & k_z^2 v_\theta + c_\theta \end{array}\right]$$

any rotation matrix can be represented this way!



$$R = R_{k,\theta}$$

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

Is axis/angle solution unique?

No. $R_{k,\theta} = R_{-k,-\theta}$

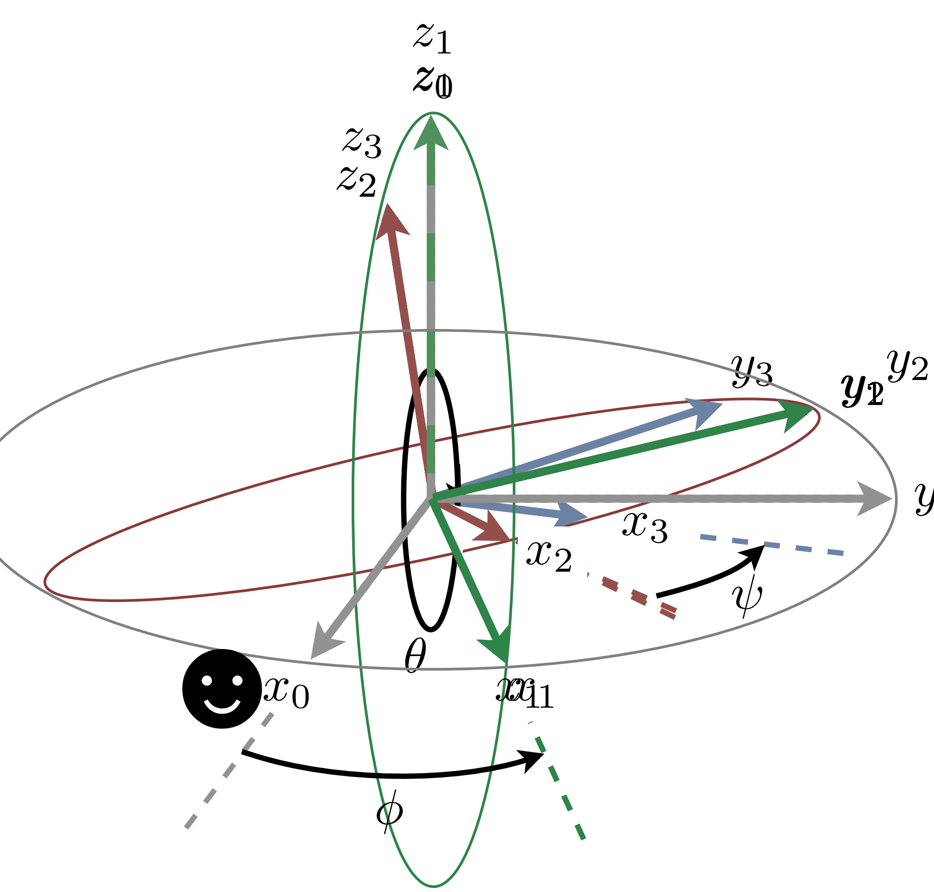
$$\theta = \cos^{-1}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right)$$

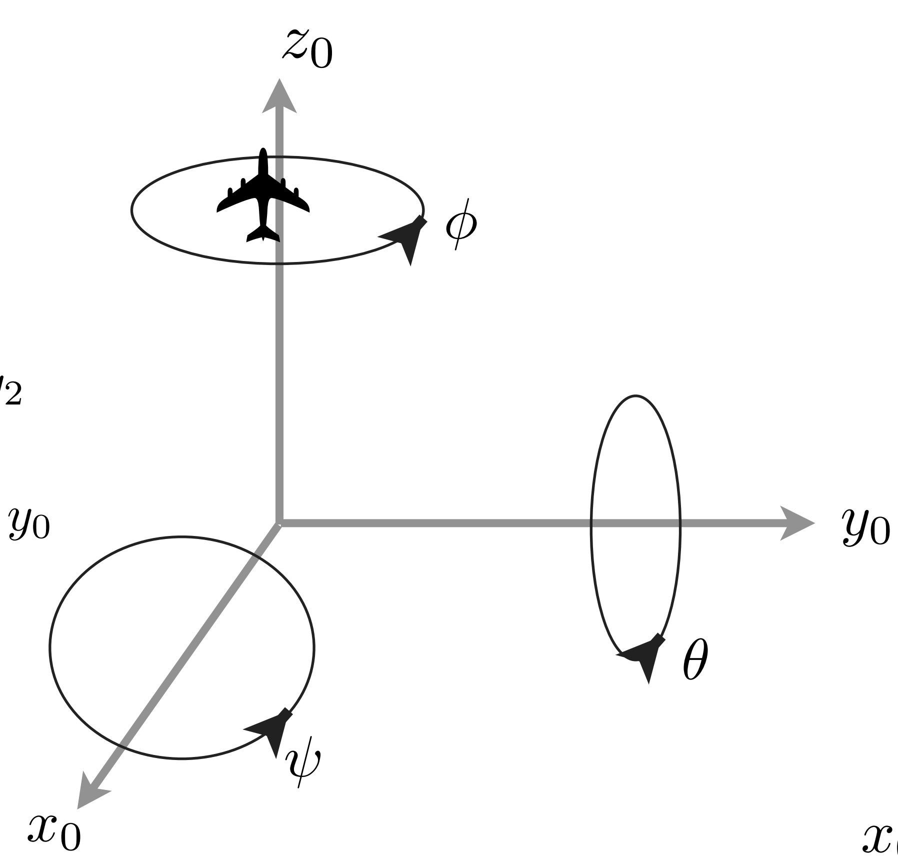
$$k = \frac{1}{2\sin\theta}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

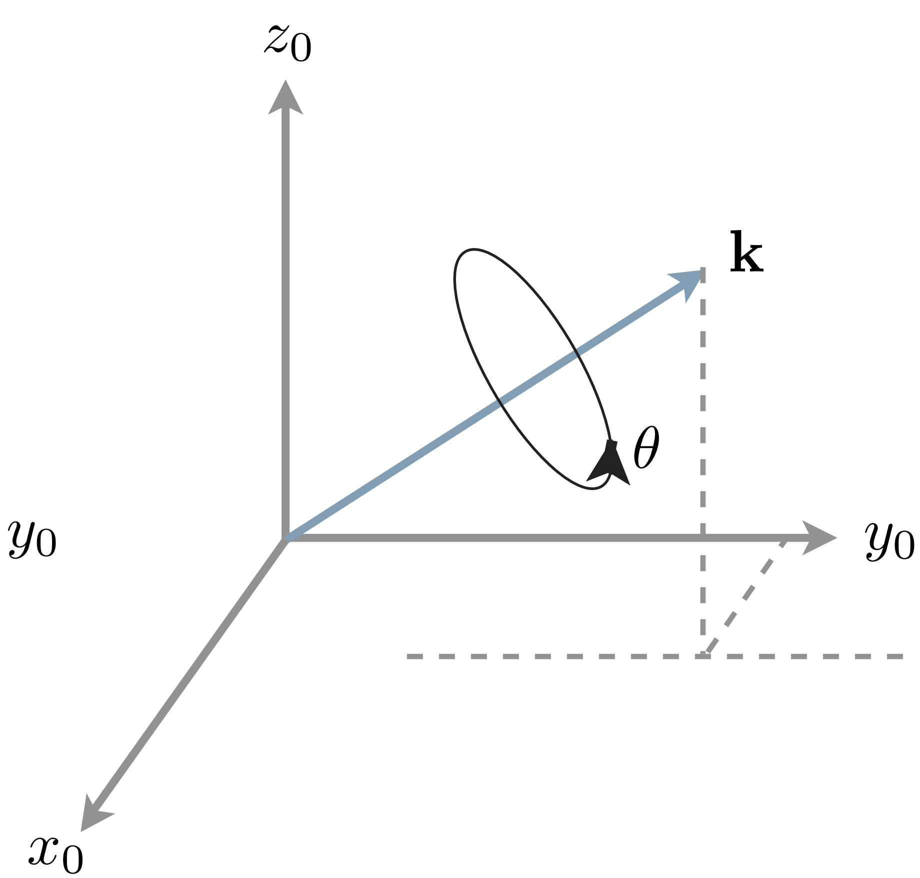
Talk to the person next to you.

Explain one of the three parameterization approaches to your partner, then switch.

Talk about the third one together.



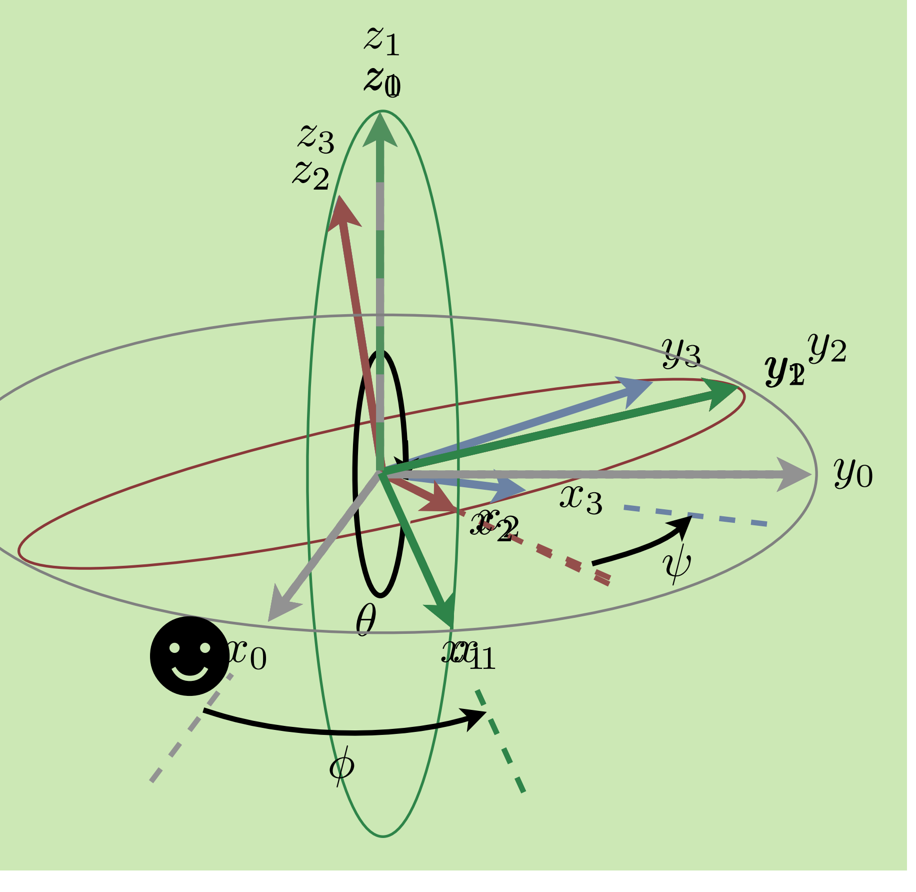




What questions do you have?

Euler Angles

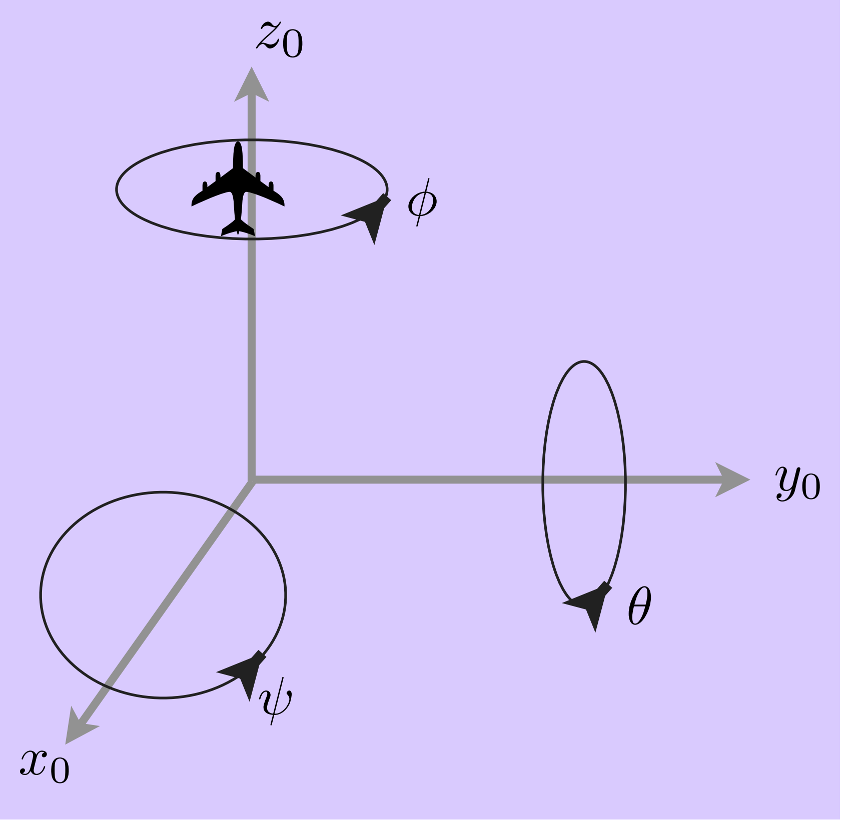
three rotations about intermediate frames



Our book uses ZYZ

Roll, Pitch, Yaw Angles

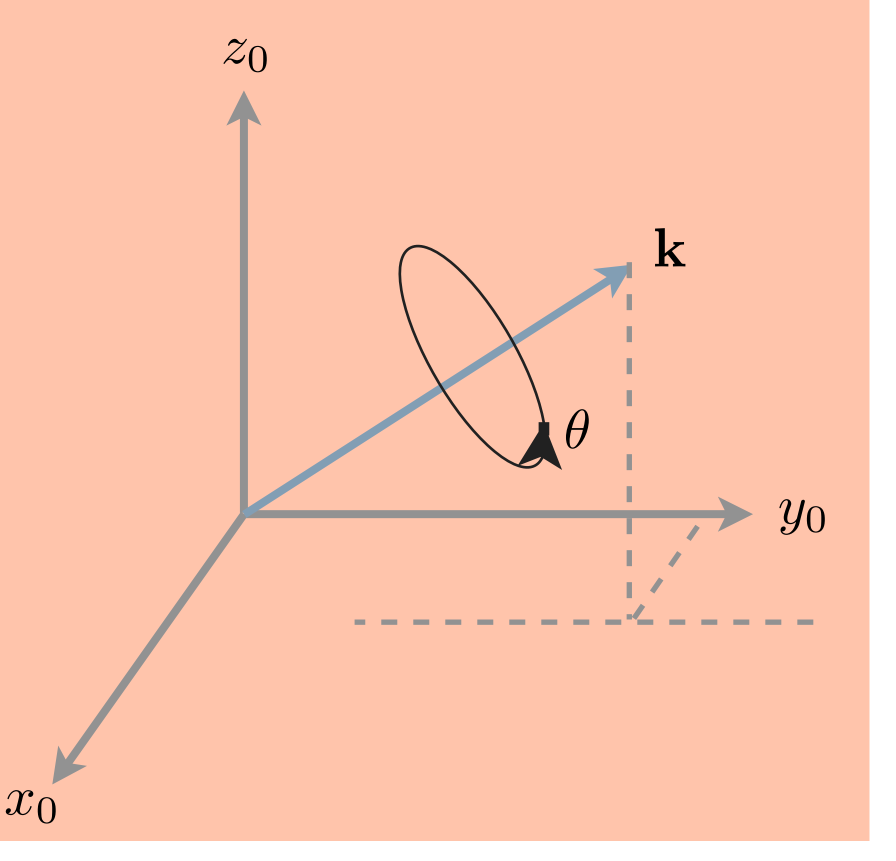
three rotations about the fixed frame



Our book uses XYZ

Axis/Angle

a unit vector (axis) and one angle



A tool to help you understand these three common parameterizations of 3D rotation matrices:

What's a p-file in MATLAB?

parameterizationsOfR.p

plotCoordinateFrame.m

plotVector.m

Why am I giving you a p-file
instead of the m-file?

Posted: a Matlab script to help you understand parameterizations of rotation matrices

Dear all,

I created a new Matlab tool to help you three common parameterizations of rotation matrices in three-dimensional space. We'll cover this material during today's lecture. The tool is called parameterizationsOfR.p, and it's posted under General Resources here on Piazza. Here is the direct link: https://piazza.com/class_profile/get_resource/hxuqr6o59c446c/hzvb42p5tok56i

The download is a zip file containing one Matlab p-file and two helper functions for plotting. Unzip it and run parameterizationsOfR in Matlab, i.e., change the current directory to the unzipped folder and type the name of the function (parameterizationsOfR) on the command line, then hit enter.

When called without any arguments, this function chooses a random 3x3 rotation matrix and depicts the orientation of the associated coordinate frame relative to the fixed base frame. Examine the frames using the "Rotate 3D" tool. Press a key when you are ready to continue.

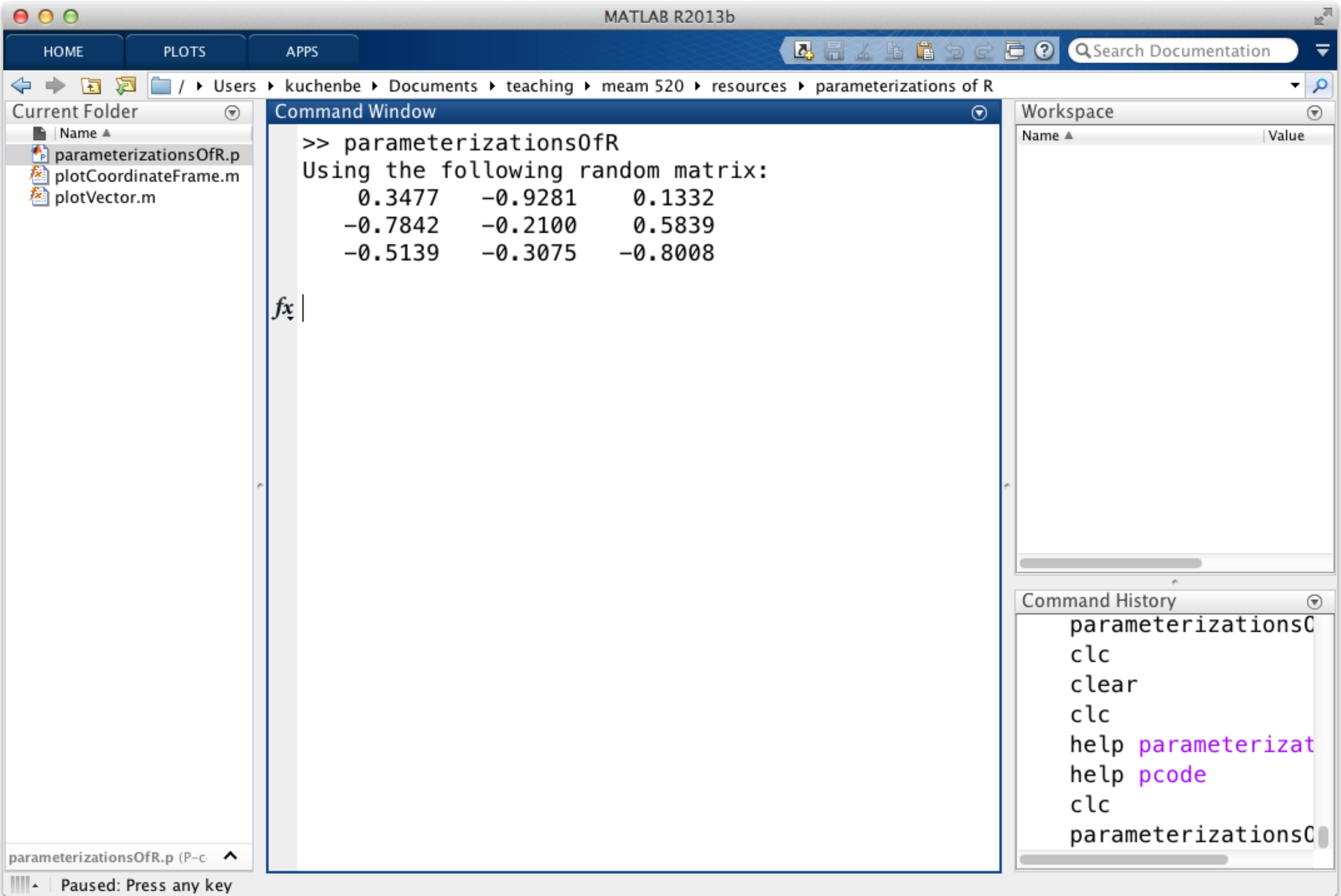
Then the script will ask you which parameterization you would like to see, from the three options covered in SHV (ZYZ Euler Angles, XYZ Yaw/Pitch/Roll Angles, and Axis/Angle). After you pick one, the graph will animate to show you each rotation in order. You will need to press a key to advance. You can keep using the "Rotate 3D" tool.

The function takes two optional arguments and returns one:

```
R = parameterizationsOfR;
R = parameterizationsOfR(nsteps);
R = parameterizationsOfR(nsteps,R);
```

The first argument is `nsteps`, the number of steps to show in the animation. The default value is 40. Increase this value to slow down the animation, and decrease it to speed up the animation.

The second optional argument is `R`, a 3x3 rotation matrix. This capability is provided so that you can visualize specific rotation matrices or re-run the code on the same matrix multiple times. Note that the script fails if you pass in the identity matrix because I didn't program all of the situations where there are zero elements in the matrix. I will update this if I have time. It's made to work with random matrices.


MATLAB R2013b
HOME
PLOTS
APPS
Search Documentation
/ Users kuchenbe Documents teaching meam 520 resources parameterizations of R
Current Folder
Name
parameterizationsOfR.p
plotCoordinateFrame.m
plotVector.m
Command Window
>> parameterizationsOfR
Using the following random matrix:
0.3477 -0.9281 0.1332
-0.7842 -0.2100 0.5839
-0.5139 -0.3075 -0.8008
Workspace
Name
Value
Command History
parameterizationsC
clc
clear
clc
help parameterizat
help pcode
clc
parameterizationsC
parameterizationsOfR.p (P-c
Paused: Press any key

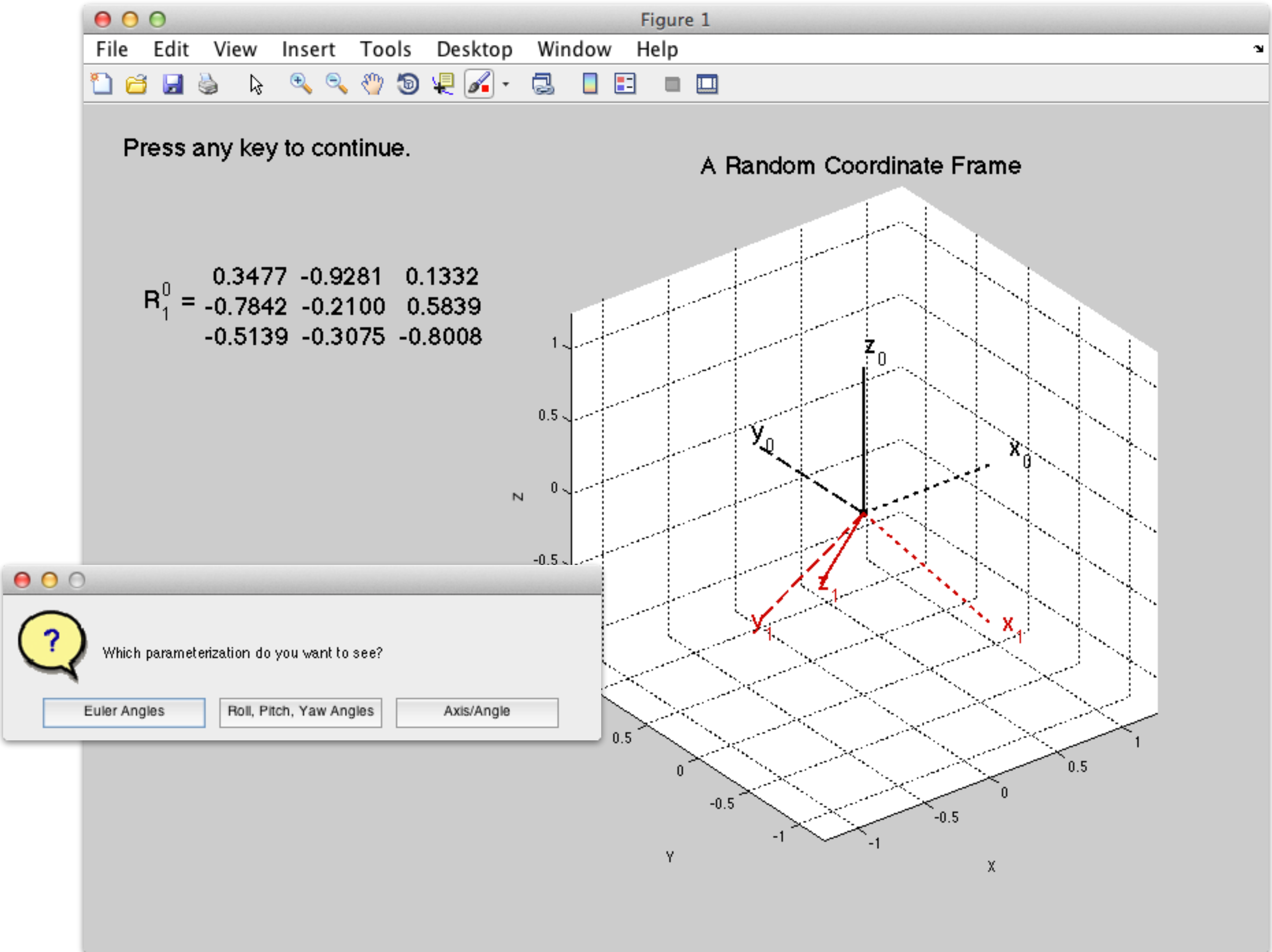


Figure 1
File Edit View Insert Tools Desktop Window Help
Press any key to continue.
A Random Coordinate Frame
$R_1^0 = \begin{matrix} 0.3477 & -0.9281 & 0.1332 \\ -0.7842 & -0.2100 & 0.5839 \\ -0.5139 & -0.3075 & -0.8008 \end{matrix}$
z_0
y_0
x_0
z_1
y_1
x_1
Which parameterization do you want to see?
Euler Angles
Roll, Pitch, Yaw Angles
Axis/Angle
Z
Y
X

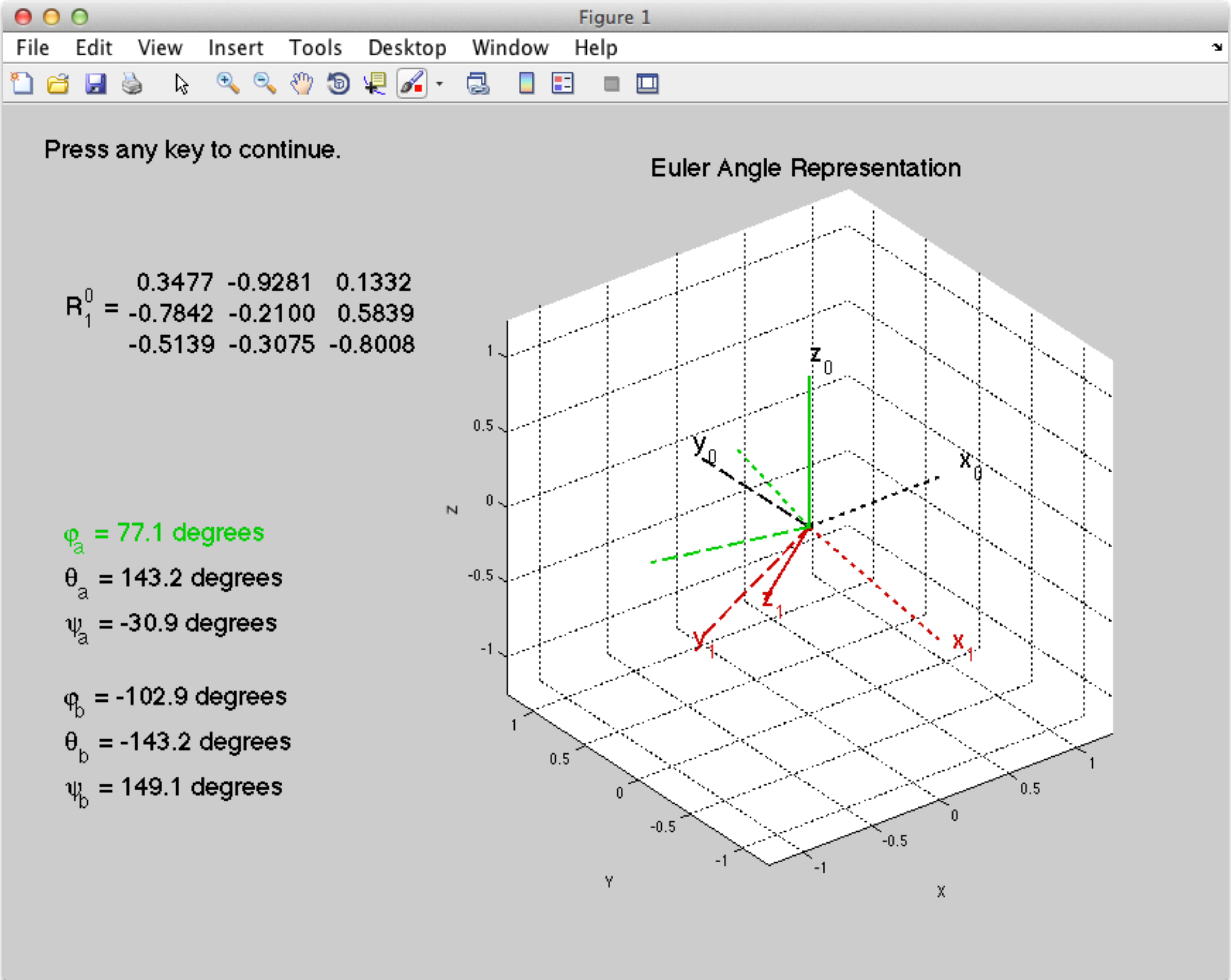


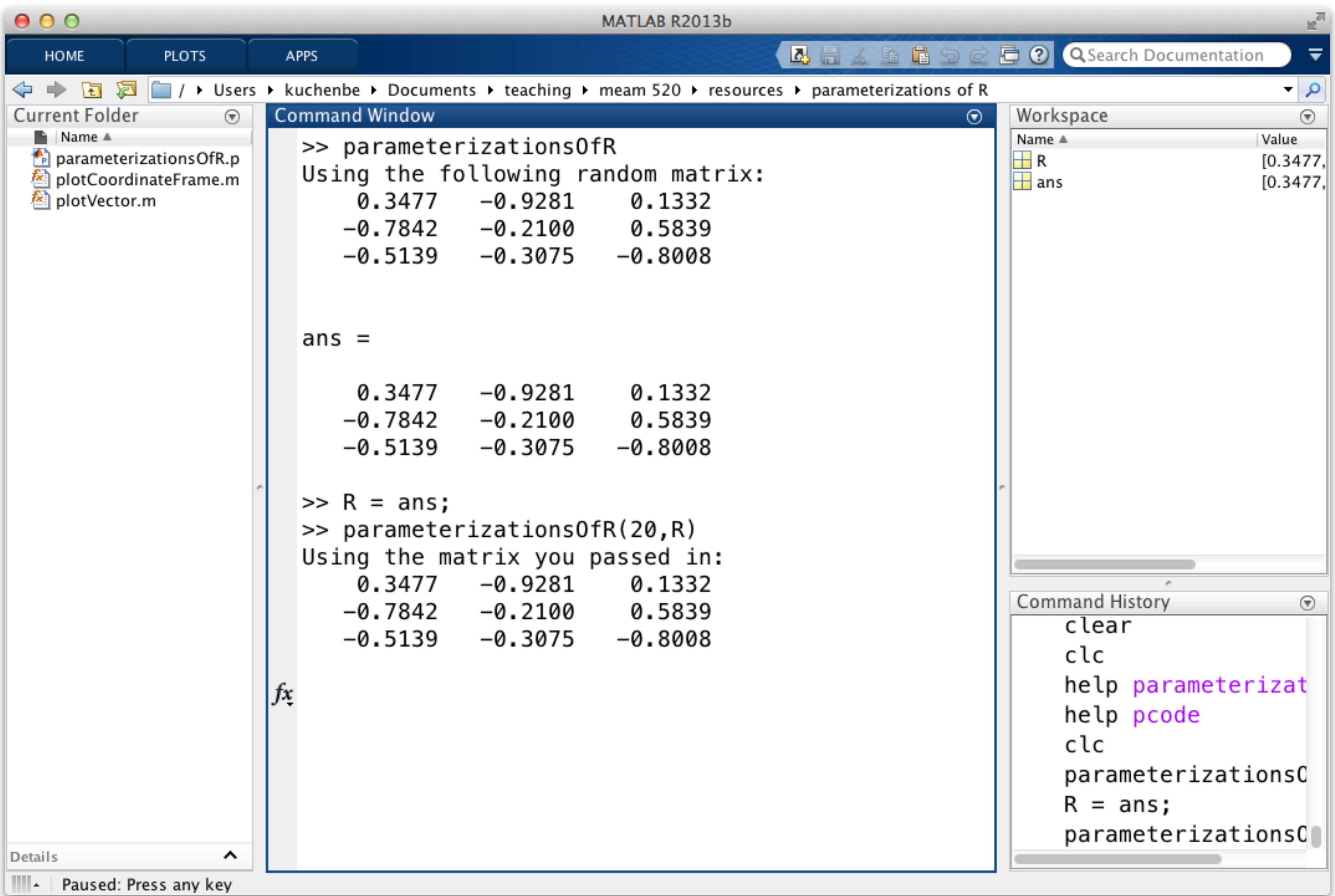
Figure 1
File Edit View Insert Tools Desktop Window Help
Press any key to continue.
Euler Angle Representation
R₁⁰ =
0.3477 -0.9281 0.1332
-0.7842 -0.2100 0.5839
-0.5139 -0.3075 -0.8008
φa = 77.1 degrees
θa = 143.2 degrees
ψa = -30.9 degrees
φb = -102.9 degrees
θb = -143.2 degrees
ψb = 149.1 degrees
z0
y0
x0
z1
y1
x1
Z
Y
X
1
0.5
0
-0.5
-1

```
parameterizationsOfR(nsteps)
```

number of steps in the animation: larger is slower, and default is 40

```
parameterizationsOfR(nsteps,R)
```

the rotation matrix to use: if unspecified, a random rotation matrix is used

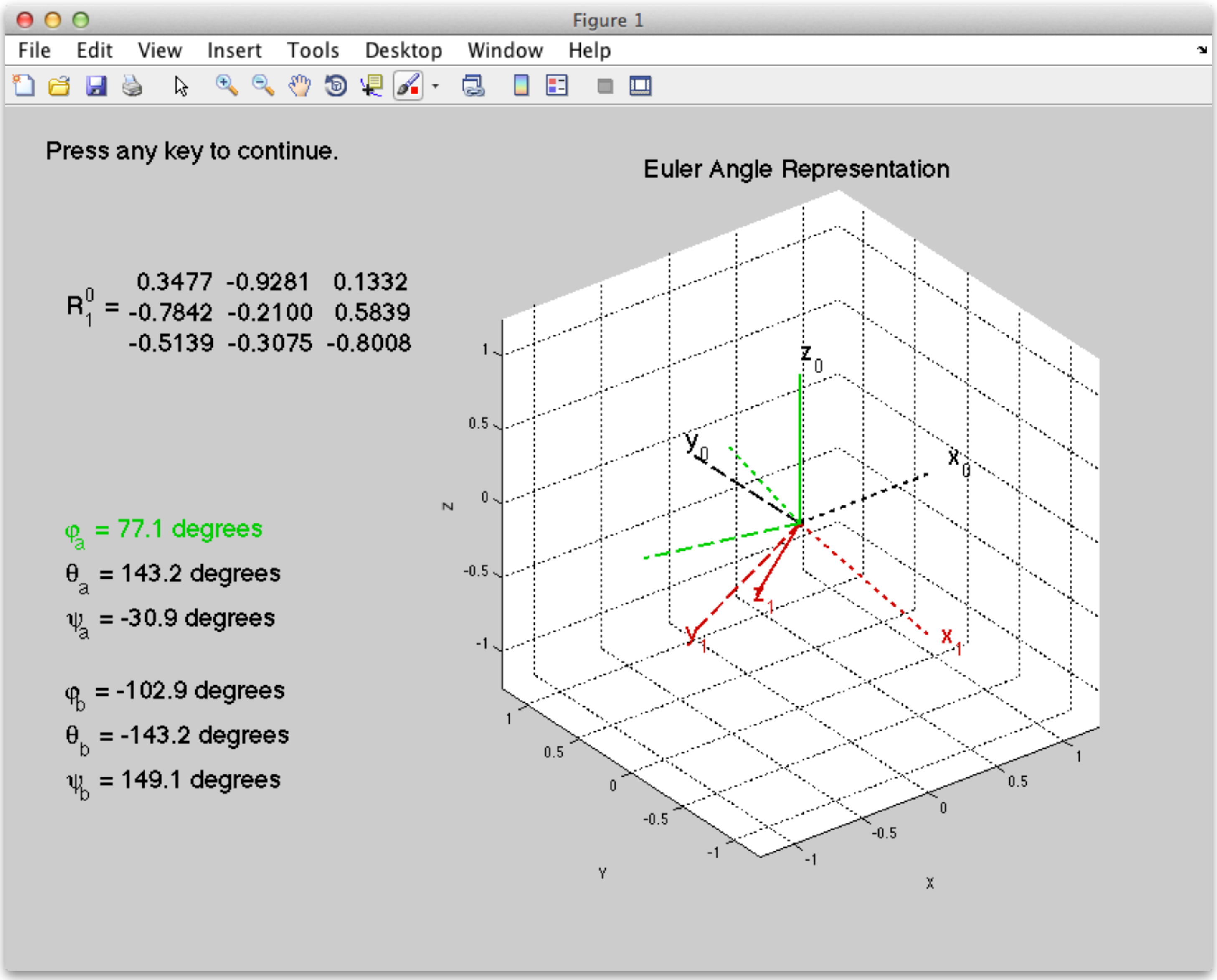


I strongly recommend that you play with this tool to hone your skills at visualizing rotation matrices and their parameterizations.

Matrices that contain zeros are not currently supported in this tool.

If you find bugs or errors in my code, report them on Piazza with the specific steps needed to reproduce the problem.

What questions do you have?



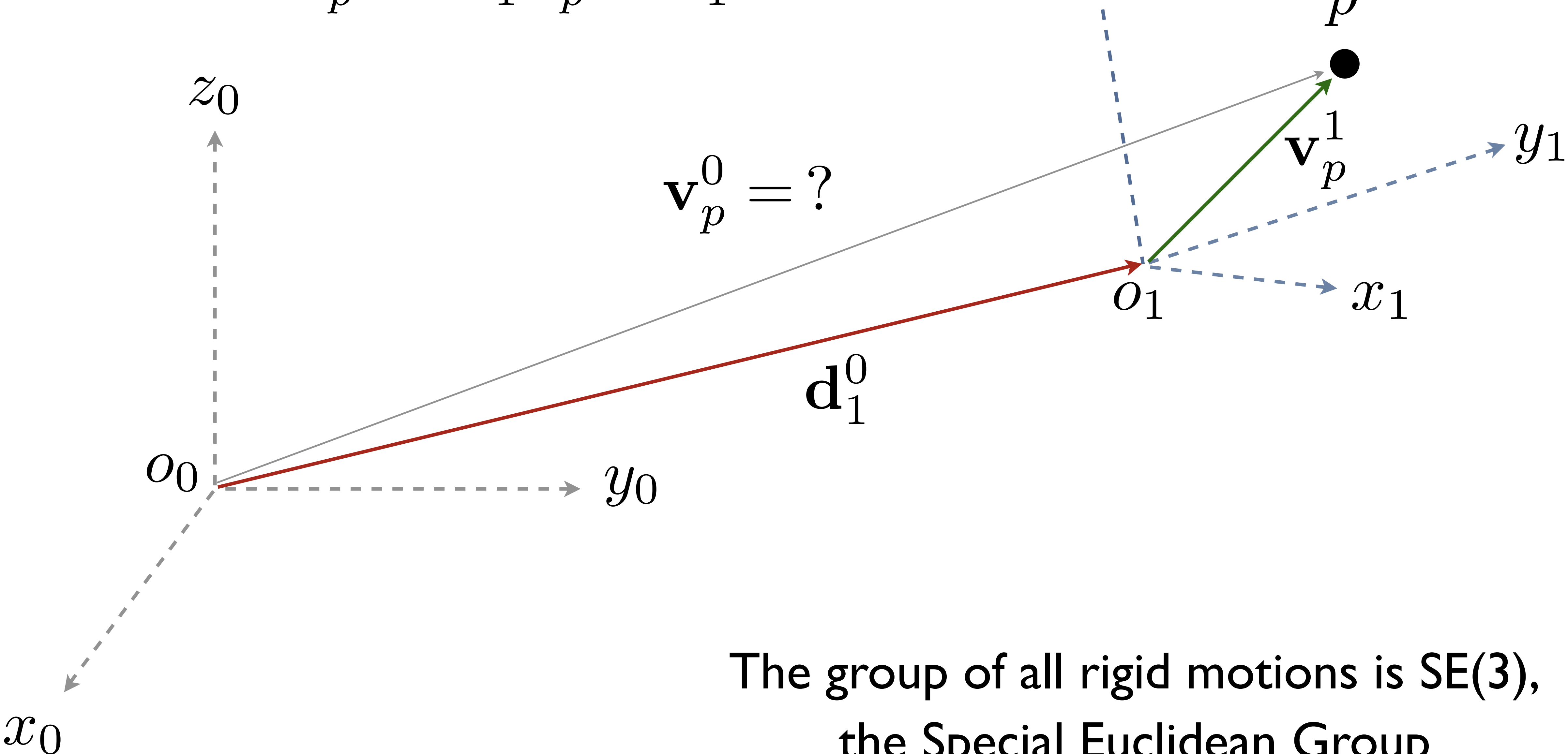
Rigid Motions & Homogeneous Transformations

SHV 2.6 and 2.7

a **rigid motion** couples pure translation with pure rotation

rigid motions can be expressed as

$$\mathbf{v}_p^0 = \mathbf{R}_1^0 \mathbf{v}_p^1 + \mathbf{d}_1^0$$



The group of all rigid motions is SE(3), the Special Euclidean Group

Reading Assignment

Within Chapter 2: Rigid Motions and Homogeneous Transformations

- Read Secs. 2.6, 2.7, and 2.8

Deadline: Thursday lecture