Exercise 1: Euler angles (ex1.py)

This program displays an interpolation of coordinate frames where rotations are represented by ZYX Euler angles. These, by convention, take on values in the range $[0,2\pi)\times[-\pi/2,\pi/2]\times[0,2\pi)$.

1. Notice that the current linear interpolation function does not interpolate between the two endpoints $(\pi/4,0,0)$ and $(7\pi/4,0,0)$ along a minimal-length curve (a geodesic). Modify the interpolate_euler_angles function so that the path does indeed interpolate along a geodesic. Make sure it also does so for other "simple" interpolations, such as from $(0,0,\pi/4)$ and $(0,0,7\pi/4)$.
2. Specify a different set of interpolation endpoints where simple interpolation of Euler angles fails to produce a geodesic. In your program, take snapshots of the interpolation and describe what is happening.

Exercise 2: Rotation matrices (ex2.py)

This program represents rotations as 3x3 matrices in the format specified in the klampt.so3 module (a list of 9 numbers in column-major order).

To interpolate between two matrices, it is currently converting to a moment representation and interpolating linearly in that space. This does not in general interpolate along a geodesic. Modify the interpolate_rotation function so that it indeed performs geodesic interpolation

Verify that your function is indeed correct by printing out the absolute angle between the interpolated rotation matrix and the endpoints. This angle should prove to be a linear interpolation.