Pose Control Numerical Robot Solver/Simulator Logic

1) Define constants
    - Link lengths
    - Desired end effector trajectory
    - Desired end effector orientation
    - Desired end effector velocity
    - Accuracy tolerance
    - Time step
2) Define initial guesses
    - Joint angles
    - End effector pose error
3) Enter for loop
    - For each waypoint
4) Enter while loop
    - While error exceeds tolerance
5) Find current point using FK
    - X, Y, and yaw if 2D
    - X, Y, Z, roll, pitch, yaw if 3D
    - Use DH or product of exponentials
6) Find Jacobian
    - 6x6 generally/3D, 2x2 or 3x3 if 2D
    - Use pseudoinverse if not square
7) Calculate error
    - Desired – actual position (may include orientation)
    - Find norm
8) Calculate velocities
    - End effector
    - Joint
9) Calculate new joint angles
    - Use initial joint position, current joint velocities, and time step
10) Exit while loop
    - When error is within tolerance
11) Reset error
    - So next waypoint loop can start
12) Find location of each joint
    - From forward kinematics of current joint positions
    - Use DH or product of exponentials
13) Plot robot
    - 2D or 3D if applicable
    - Set and label axes
    - Title plot
14) Exit for loop
    - When final position reached