Lab 14: Animated Squash

The main assignment for this lab is to animate the trajectory of a ball in the game of squash, which is played with an elastic ball hit against a solid wall. In constructing this code, it is advisable to refer to the 3D pendulum animation presented in class. It may also help to consult the baseball labs.

1. The squash "court" should be a cubical 3D space, 10 m on a side. Set up the initial conditions $(x_0, y_0, z_0, v_{x,0}, v_{y,0}, v_{z,0})$ so that the ball is hit from a front corner of the court from a height of 1 m, roughly toward the center of the back wall (at $y = 10\,\mathrm{m}$), with a low angle and with a speed of 70 $\mathrm{m\,s^{-1}}$ (it may be simplest to specify the angles θ, ϕ for the velocity before defining the velocity vector).

2. Write down the analytic solutions for the ball position $x(t), y(t), z(t)$, including gravity but no other forces yet. Implement these in your code using vectorized calculations, over the time interval from $t = 0$ to $t = t_{\mathrm{wall}}$, where t_{wall} is the time it takes for the ball to reach the far wall. In other words, you are starting with just the trajectory up to the first bounce.

3. Now plot the trajectory in 3D. Click and drag on the plot window to change the viewing direction until you are happy with it. Record the values of azimuth and elevation shown in the lower left of the window. Hardwire these angles into your code using a command like: `ax.view_init(elevation,aximuth)` (note the reversal of the angles).

4. Now comment out your vectorized solution and re-do it as a loop over time, as needed to implement the bounce. Remember to think in terms of indexing x_i, y_i, z_i – even if you are not explicitly looping over i. Set up the y solution as $y(t - t_0)$, where $t_0 = 0$ initially, to prepare for the next step including the bounce.

5. Now add the bounce. Change your t vector to extend to $2\,t_{\mathrm{wall}}$. When the ball reaches the wall within the loop, reverse the y-component of the velocity, and reset y_0 and t_0 to correspond to the bounce. You should now have a plot of the forward and backward parts of the ball trajectory.

6. You are now ready to add the animation. Your best bet to avoid errors is to copy and paste the animation part of the pendulum code, and then to modify it appropriately – e.g., with the bob of the pendulum becoming the squash ball.

7. Now add a "trail" behind the ball during the animation, to show the trajectory of where it has been.

Bonus points: Turn the squash game into racquetball, with bounces permitted from all six "walls."

Submit your working code to Canvas.