Mapping Dynamic Tennis Ball Movement due to Ball Spin

Sean Yamamoto

PHYS1600 Final Project

Introduction

Professional tennis players hit their serves at blistering speeds up to 150 mph and with spins in different directions depending on the objective of the server. Spin can cause a ball's trajectory to behave in complex manners that seems to defy physics and certainly confuses tennis players. I will model and analyze the ball path of a tennis serve taking into consideration the spin and initial velocity of the ball. The most important characteristics of a tennis serve are the speed of the ball, the x and z values of the ball when it reaches the other end of the court (refer to *Figure 1* for axis orientation). I have broken this path up into 3 critical sections:

1. The path of the ball before the rst bounce. I will use Reynolds number, velocity, and spin to calculate lift and drag force values throughout this section of the ball path. I will then use RK4 to calculate the corresponding trajectory of the ball.
2. The ball bounces on the ground. The bounce direction and spin after bounce is primarily determined by the spin and incident velocity of the ball.
3. The subsequent trajectory of the ball until it hits the ground again. This can be calculated using the same method as Section 1.

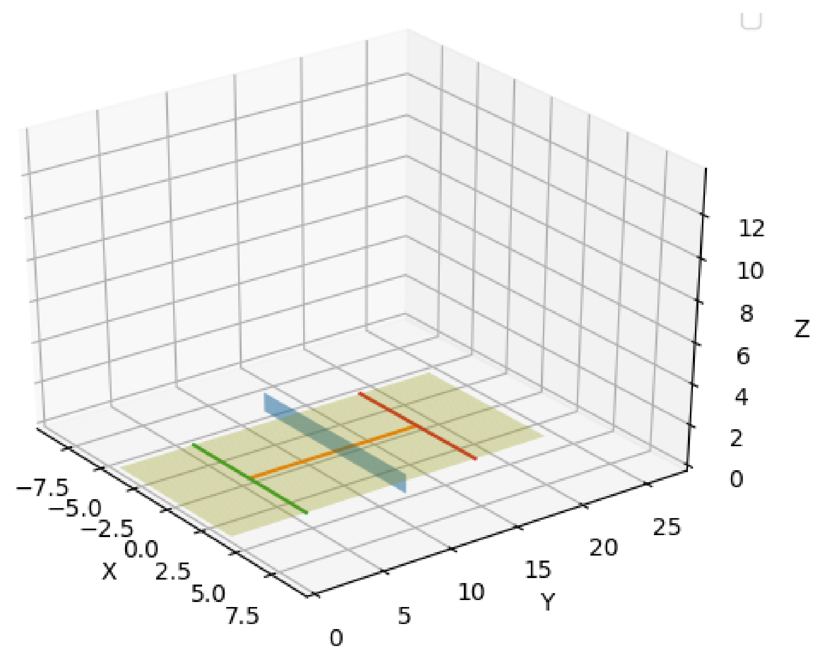

Figure 1: Axes orientation and court model

Background

The forces that act upon the ball while it is mid-air are gravity, drag force, and lift force. Gravity is constant throughout the flight but the drag force and lift forces vary with ball velocity and spin throughout the flight. These forces are generated by the ball's fluid mechanics as it pushes air out of the way in order to continue on its path. Both the drag and lift forces have magnitudes given by *Equation 1* where C is the drag or lift coefficient, A is the cross-sectional area of the ball, p is the density of air, and v is the speed.

$$Equation\ 1:\ F\ =\ CApv^2/2$$

The direction of the drag and lift forces, however, are very different. While the drag force acts in the direction opposing motion, the lift force acts in a direction normal to the velocity and spin of the ball, shown in *Figure 2*. Drag force can be thought of as the response to the ball traveling linearly through a fluid while lift force is the response to the ball rotating in a fluid. Since the drag force acts in the direction opposite of velocity, it does not affect the ball's trajectory nearly as much as the lift force simply due to the fact that the lift force acts in a direction nearly perpendicular to motion, either upwards (backspin) or downwards (topspin). The hardest part about modeling the drag force and lift forces is finding the coefficients C_D and C_L, which will be discussed in the next section.

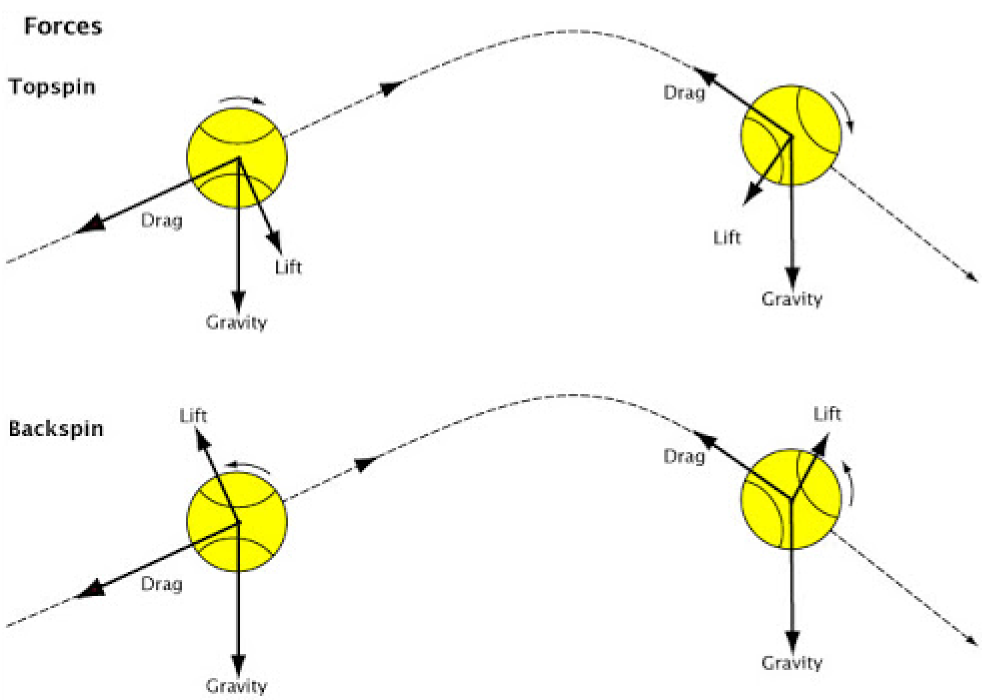

Figure 2: Free body diagrams of topspin and backspin tennis balls from: Cross, Rod. "Tennis Ball Trajectories" *Tennis Warehouse University*, 22 December 2013

The bounce dynamics of the ball is an equally difficult problem but it's governed by the same parameters as the trajectory: velocity and spin. The ball also has a fundamental coefficient of restitution of approximately 0.8. The dynamics of the bounce problem are well described by the following scenario:

Suppose a ball is dropped with an immense amount of spin in the +x direction. When the ball bounces, it does not bounce straight up. It actually bounces up and in the +y direction, which is normal to the spin direction.

This change in direction is caused by the rough tennis ball surface catching on the rough tennis court, creating a friction force that causes the ball to exit the bounce at an angle different than the angle of incidence if there is spin. The magnitude of this force varies greatly with the velocity and spin orientation of the ball. In certain cases, the ball leaves the bounce with spin, which can be greater or less than its incident spin rate.

Methods

I started by modeling the ball's path using a generic model of a spinning projectile given in the Particle3D file. I used the RK4 method to compute this trajectory due to its overall precision for estimating differential equations. This method takes in the initial conditions x_0, v_0, ω_0, along with the differential equations of motion of the system $dx/dt = v$, $dv/dt = F/m$, and outputs the corresponding path. Although RK4 is not a symplectic integrator, this is okay for my simulation since the ball does not conserve energy during its path, being subject to negative work the entire time.

The next step was to improve this model by making it specific to a tennis ball since a tennis ball. I first discovered that the fluid mechanics problem is an extremely difficult problem that physicists are still trying to solve. I decided that the best course of action would be to use an estimated model of these forces determined by experimental data. I found several simple models for drag and lift force online but they were not completely consistent and few were based on thorough experimentation. Upon further research on tennis ball drag and lift forces, I found an academic paper summarizing an experiment focused on defining the drag and lift coefficients of a tennis ball, which was just what I needed. A summary of the information I used from this paper is as follows (Chadwick, November 2003):

- The drag coefficient is best summarized by the value of 0.53
- The lift coefficient is best summarized by the equation $(2 + v/r\omega)^{-1}$

Chadwick reiterated the difficulty in testing these two parameters as they concurrently affect the ball and it can be hard to distinguish them from each other. He used four separate methods to test these parameters, using wind tunnels and precision photography to map his results. The values he suggested using are the ones I used in my program since he determined these results were the most reliable in comparison to his other findings. Arriving at C_D and C_L values let me finalize the trajectory part of my project. I tested my model with a simple test consisting of three tennis balls.

1. Topspin ball: $\omega_x = 5,000\ RPM$
2. Flat ball: $\omega_x = 0\ RPM$
3. Backspin ball: $\omega_x =- 5,000\ RPM$

Each of the balls were given a launch angle of 6°, initial speed of 60 mph, and an initial position of (0, 0, 1). The results of this test were exactly what I expected, with the topspin having the lowest peak height and smallest distance traveled, the backspin ball having the highest peak height and largest distance traveled, and the flat ball right in the middle. These results are shown below in *Figure 3*.

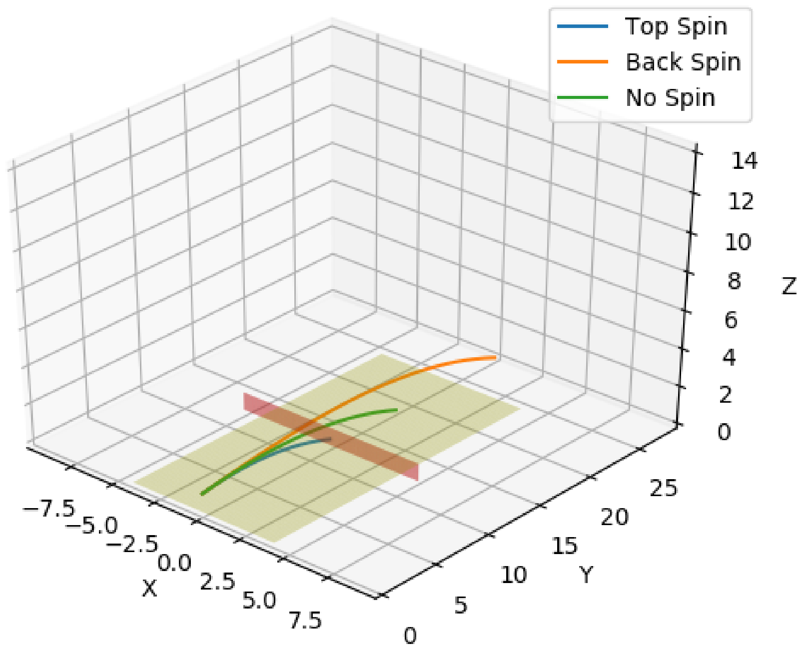

Figure 3: Testing the drag and lift forces

The next thing I had to do was to determine the bounce mechanics of the ball. My method for determining the resulting ball velocity and spin after bouncing is similar to how I determined C_D and C_L. I started by conducting research on the topic and found that this is also an extremely hard situation to model perfectly. The first recordable resource I found was another experiment report (Jafri, May 2004) detailing the impact dynamics of a tennis ball with a flat surface. Two interesting cases detailed in the experiment include:

- Backspin
    - Friction larger than topspin case
        - Skids on the ground
        - Touching ground for longer
    - Backwards spin on ball eliminated in most cases
        - Friction force on ball induces topspin
    - Speed deteriorates
- Topspin where $r\omega >= v$
    - Ball rolls on ground during impact
    - Friction reduced
    - Spin reduced slightly
    - Speed better conserved

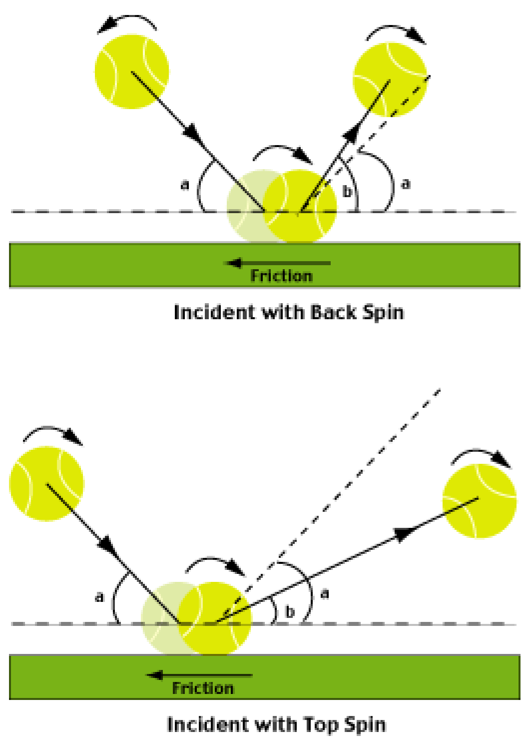

Figure 4: Spinning ball impacting court with reflected ball corresponding to initial spin from: Lindsey, Crawford. "Follow the Bouncing Ball" *Tennis Industry Magazine,* April 2004

While I was able to deduce lots of intuition from this paper, it did not provide a comprehensive model of the bouncing ball. I used the experimental data and my physics knowledge to produce a rough model for the ball bounce.

The first thing I did was I came up with a model for v_x, v_y, and v_z after the bounce. For v_z I simply multiplied it by the coefficient of restitution, 0.8. For v_x and v_y, I wrote functions that depend on spin and velocity. I took into consideration the fact that the magnitude of the effect on the bounce differs for topspin and backspin and the fact that energy is conserved in this system, even if it is impossible to keep track of. The v_z function is given by *Equation 2.* Flat balls (balls with no spin) were also governed by this equation.

$$Equation\ 2\colon\ v_z = 0.8v_{zi}$$

v_x and v_y depend on the spin on the ball. These functions were derived using the difference between translational and rotational energies, using $I = 2mr^2/5$ for the moment of inertia of the ball. For backspin the spin always reduces the velocity and when ω is large enough, it causes the ball to launch backwards, consistent with reality:

<u>Topspin</u>

$$Equation\ 3\colon\ v_x = ((0.8v_{xi})^2 - 2r^2\omega_y^2/5)^{1/2}$$

$$Equation\ 4\colon\ v_y = ((0.8v_{yi})^2 - 2r^2\omega_x^2/5)^{1/2}$$

For topspin the velocity is decreased by an amount proportional to $(r\omega - v)$ since when $(r\omega = v)$ the ball rolls and more of the energy is conserved. The more positive this difference is, the more energy is conserved and the larger v will be. The more negative this difference is, the smaller the resulting v.

<u>Backspin</u>

$$Equation\ 5\colon\ v_x = ((0.8v_{xi})^2 + 2(r\omega_y - v_{xi})/5)^{1/2}$$

$$Equation\ 6\colon\ v_y = ((0.8v_{yi})^2 + 2(r\omega_y - v_{xi})/5)^{1/2}$$

These functions were chosen because they fulfill several key cases that have to do with spin. For one, the x and y velocity functions decrease by a scalar value related to both their current velocity and the spin that affects movement in that direction (for v_x, ω_y). I tested to make sure that for normal spin and velocity values, the magnitude of velocity decreased by approximately 20-50%, which is consistent with the experimental data collected in Jafri's study. While these functions are by no means perfect, they create the type of response that I was looking for.

Next, I determined the values of the resulting spin from the bounce. These values differed by backspin and topspin.

Backspin

$$Equation\ 7:\ \omega\ =\ \omega_0 - 8(5(0.2v_i)^2/2r^2)^{1/2}/10$$

Topspin

$$Equation\ 8:\ \omega\ =\ \omega_0 - 1(5(0.2v_i)^2/2r^2)^{1/2}/10$$

The general effect is that the backspin's resulting spin is more heavily influenced by the friction force as it bounces off the ground since it spends more time on the ground. The function I chose converts backspin to topspin unless backspin is very large and velocity is very small, in which case the ball retains its backspin. The topspin ball loses small amounts of its spin due to the friction force being smaller on it but retains its topspin.

The final component I added was the ball "kick" due to a ball having massive spin in one direction. This is the same as the example I layed out earlier in the report where a ball can actually bounce backwards. The way I did this was I first determined the direction of this perturbation. It turns out that the force is exerted in the direction normal to $(v\ x\ \omega)$and I gave it a proportionality constant of $5(0.2v_i)^2/2r^2)^{1/2}/100$to scale it to a reasonable number.

Results

Figures 5 and 6 show my experimental results plotting the three most spin dependent serves, the topspin serve, slice serve, and kick serve. *Figure 5* does a good job showing the variation of the movement on the x-axis for the different serves. Just like real life, the kick serve "kicks" to the +x direction after it bounces, the slice serve curves to the -x and the spin serve stays fairly constant in direction. Each of these balls were given the same initial speed and position. The only difference was their spin rates, given in *Table 1.*

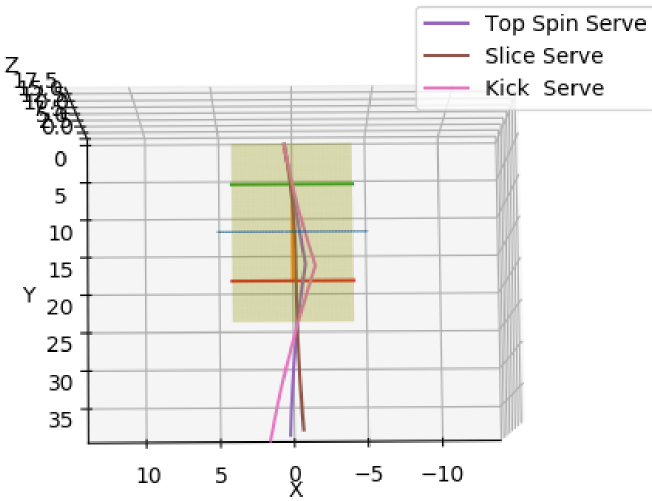

Figure 5: Aerial view of results

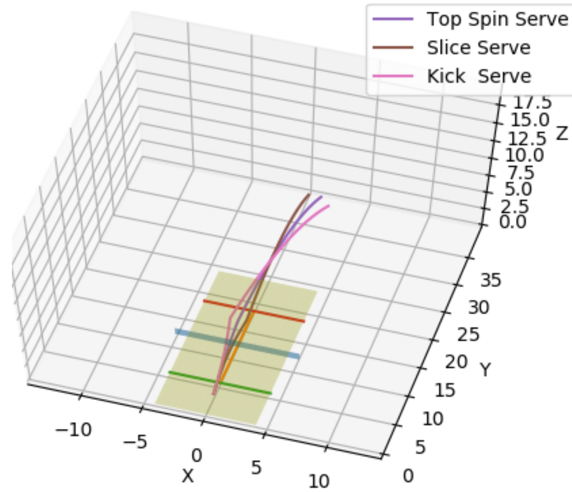

Figure 6: Top/side profile of results

Serve	ω_x (rpm)	ω_y (rpm)	ω_z (rpm)
Top spin	3,000	0	0
Slice	3,000/1.5	0	3,000
Kick	3,000/1.1	0	-3,000

Table 1: Spin rates for shots displayed in *Figures 5 and 6*

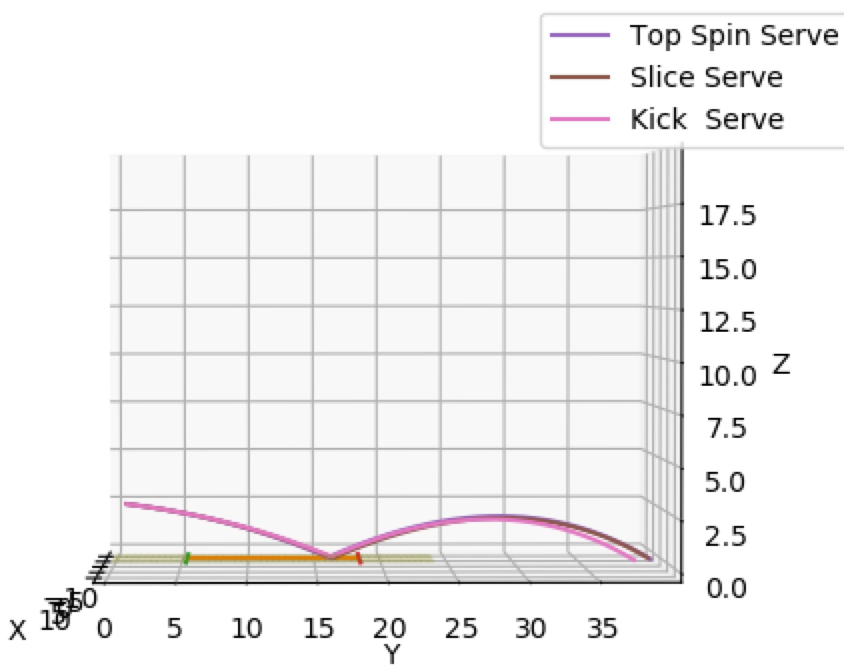

Figure 7: Side view of results

The results I developed were generally consistent with what I have seen for the 10+ years I have been playing tennis. One obvious error, however, is displayed in *Figure 7*, where all the serves seem to have the same Y vs. Z behavior for most of their paths when the top spin serve should 1) land first and 2) have the lowest height. As we can see, the kick serve has the lowest height and the slice serve somehow has the largest height. I know that my drag and lift forces work correctly due to my tests earlier on, before implementing the bouncing mechanics so my bouncing mechanics are obviously not completely correct. I was, however, very happy with the x-axis movement of the serves, especially the kick serve. More error accumulates when one thinks about the ball spin decreasing in magnitude throughout the flight due to air friction. The drag and lift forces, of course, could have been more accurate but even if they were only 90% accurate, they still served their purpose of manipulating the flight in the correct directions at least.

Discussion

As I mentioned above, my general results were very consistent with my conceptual understanding of tennis and its physics. This has long been one of my goals to produce a simulation of a tennis ball with spin. Unfortunately, there is no actual data I can compare my results to. My drag force and lift force approximations match those of Chadwick's thousands of data points that he used to model C_D and C_L. In terms of directions of forces, my simulation was also very accurate, even when it comes to the bouncing mechanics. I understand that the two problems I had to solve in this project were both nearly impossible to model perfectly but the bouncing mechanics I developed were definitely the biggest source of error in my simulation.

Conclusion

I learned that both the fluid dynamics problem and the bounce dynamics of a rough, spinning ball on a rough surface are both still very real problems that have not been solved. Obviously humans have been able to solve them to a certain degree, enabling them to build planes and send rockets to the moon. Given the research I have done on both of these problems, I concluded that in order to obtain the best possible model for these two classical mechanic problems, one should run thorough experiments to plot the exact behaviors of the tennis ball. This is applicable for other types of balls such as basketballs being bounced on the court and shot with backspin, or baseballs being thrown at 100 mph with dynamic spins, since they are all essentially the same problem, just with a different outside surface, volume, mass, and coefficient of restitution. Solving for one will give the general behavior for all of them.

Tennis tournaments use expensive cameras to track shots down to a precision of 3mm, which is an incredible feat in itself. If someone were to gather this data, as well with the spins of each shot, one could easily map out the mechanics of the ball with the millions of data points collected from a single tournament like the US Open. If a model was developed that is good enough, tournaments could switch to using just this program to determine the landing spot of a ball i.e. determine if it was in or out. This type of technology would be a game changer in the world of tennis where millions of dollars can be lost due to a bad line call by a line judge or photography that is not detailed enough to determine the actual landing spot of the ball.

References

1. Chadwick, Stephen George. "The Aerodynamic Properties of Tennis Balls." *The University of Sheffield Department of Mechanical Engineering*, Nov. 2003, pp. 250–257.
2. Cross, Rod. "Tennis Ball Trajectories" *Tennis Warehouse University*, 22 December 2013
3. Jafai, Syed Muhammad Mohsin. "Modeling of Impact Dynamics of a Tennis Ball with a Flat Surface." *Graduate Studies of Texas A&M University*, May 2004, pp. 98–116.
4. Lindsey, Crawford. "Follow the Bouncing Ball" *Tennis Industry Magazine,* April 2004

Appendices

Code can be referenced using the following GitHub link:
https://github.com/syamamo1/PHYS1600/tree/main/finalProject