The Flight of a Golf Ball

Andrew Reese

April 2025

Introduction

When it comes to golf, people say that it's a game of inches. This is meant in the sense that every detail, no matter how small, is crucial to the overall performance of the player. One of the most important aspects of the game is distance control. How far you can hit the golf ball and the consistency in which you can reproduce that same number can either make or break your final score on a given day.

The reason that distance is so difficult to create and control is due to the myriad of factors that play into how far the ball will go. There are some obvious contributors like the club choice used to hit the ball, the ball's velocity, and even the angle at which the ball launches from. However, there are many more hidden factors such as the spin of the ball, something called the smash factor (which will be elaborated on later), and even weather-related factors like temperature and humidity.

Because of all of these different elements, changes in technology surrounding the game have been implemented to better understand what is going with the flight of a golf ball and how to best control it and make changes to it when necessary. Today, we aim to create a functional, predictive model of golf ball flight, namely to estimate distance.

Historical Background

Golf itself has been around for centuries, albeit quite different than it is played today, but there are records of organized golf clubs dating back to 1744 when a group of Scottish men created the Gentlemen Golfers of Leith. For the interests of this paper, however, we are going to focus on the PGA Tour, which started out in 1926 as the PGA of America and was later renamed in 1975. [1]

Tracking the distance of players' drives and fairway-shots is something that only began in the 1980's. At that point, technology wasn't as advanced as it is today, so the PGA would pay employees known as spotters who would spread out near known-distance markers throughout certain holes and report back with estimated total distances of each players' shot. Today, things are much more streamlined and automated. The PGA has recently contracted a company called TrackMan, who specializes in distance-tracking-technology, to supply such devices for their own use. TrackMan devices are usually in the form of small, TV-like, monitors which use advanced calculations to accurately determine how far each shot will carry and roll out.

Figure 1: TrackMan Launch Monitor.[3]

According to TrackMan's official website, there are over 40 parameters used in the distance calculation process. Some of these parameters are more impactful than others, but the engineers behind the designs insist on maximizing accuracy and find their method to be the best way of doing so. [4]

Modeling Assumptions

Now, to craft a distance-predictive model by hand, there are several assumptions that need to be made. Just as all modeling problems show, things can get extremely complicated without such assumptions, and often times we can still strive for accuracy while being more time-efficient.

1. **Only the flight phase is considered** — swing kinematics such as club-head path and speed are beyond the scope of this model
2. **The ball is treated as a point mass** with lift and drag forces acting on it
3. **Air resistance (drag)** and **lift (Magnus force from spin)** are modeled using standard coefficients.
4. **The spin rate is constant** during flight (i.e. no decay)
5. **Environmental conditions** like temperature, altitude, and wind are assumed to be constant throughout a shot. This model will apply best to things like indoor simulators which essentially assume ideal conditions as far as weather is concerned.
6. Gravity is constant at $g = 9.81\,\mathrm{m/s}^2$.
7. Ball speed, spin rate, and launch angle are measured at impact (i.e., initial conditions).

Derivation of Flight Distance Equation

We'll now derive a model for the carry distance R using physics and dimensional analysis.

Step 1: Define Key Parameters

- v_0: initial ball speed (m/s)
- θ: launch angle (radians)
- ω: backspin (rad/s)
- ρ: air density (kg/m^3)
- C_D: drag coefficient (Between 0.25 and 0.35 for golf balls) [2]
- C_L: lift coefficient (from spin, typically between 0.1 and 0.3)[2]
- A: cross-sectional area ($A = \pi r^2$), $A \approx 1.432 \times 10^-3$ [6]
- m: mass of ball (approx. 0.0459 kg) [6]
- g: gravitational acceleration (9.81 m/s^2)

Step 2: Newton's 2D Equations of Motion

Let:

$$v = \sqrt{v_x^2 + v_y^2}, \quad F_D = \frac{1}{2}\rho C_D A v^2, \quad F_L = \frac{1}{2}\rho C_L A v^2 \quad from^{[2]}$$

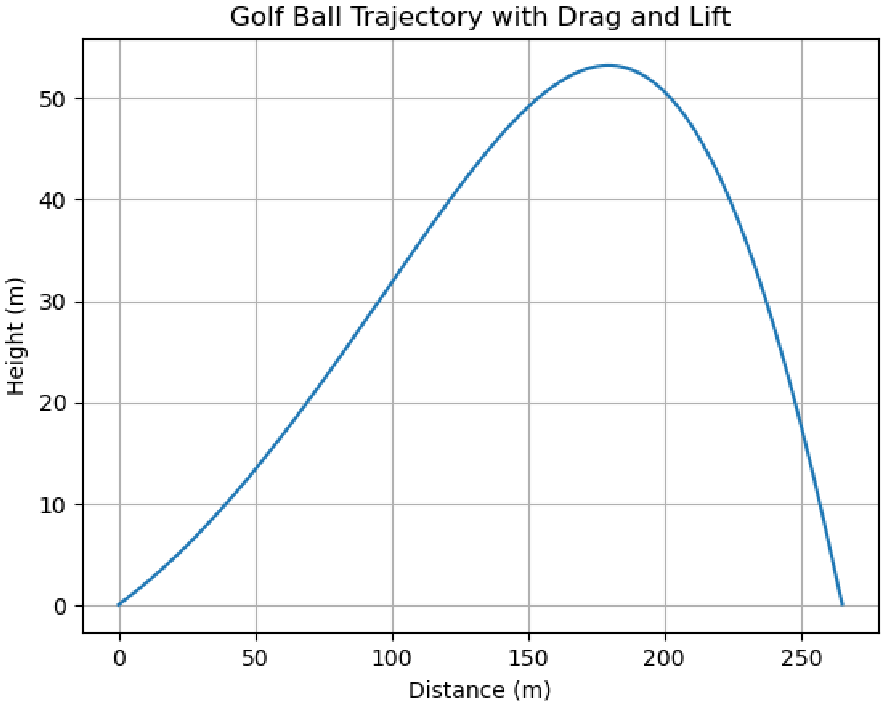
Then the forces on the ball are:

$$m\frac{dv_x}{dt} = -F_D\frac{v_x}{v} + F_L\frac{v_y}{v}$$

$$m\frac{dv_y}{dt} = -mg - F_D\frac{v_y}{v} - F_L\frac{v_x}{v}$$

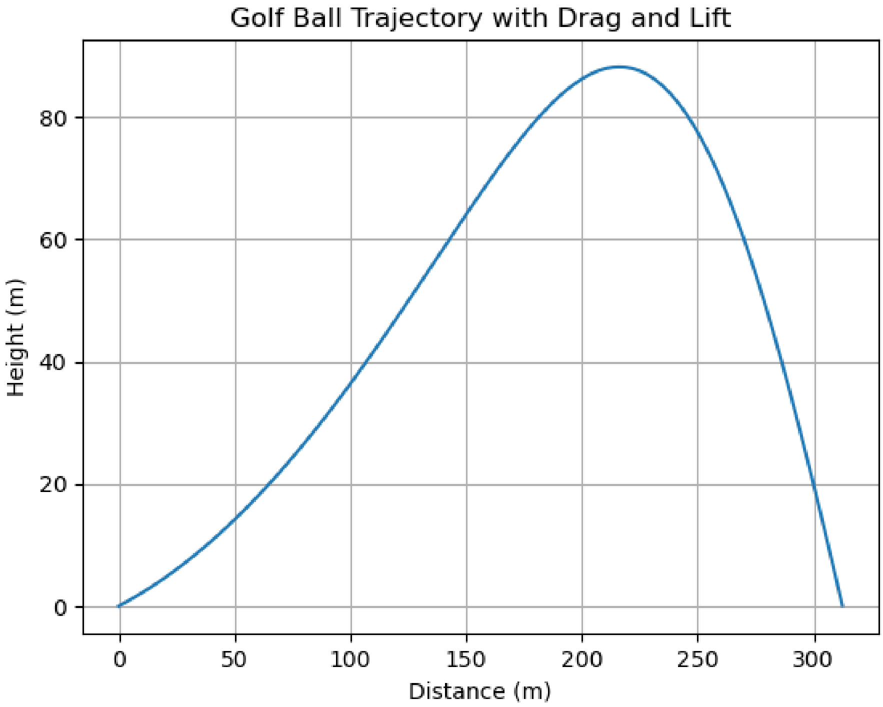
Step 3: Use these ODE's to create outputs in Python

Here is a plot of the curve that a golf ball would make under PGA Tour average ball speed (167 MPH or 74.65 m/s) and spin rate (2,686 rpm)[5]:



Which has **Total Carry = 265.28 Meters (290.11 Yards)** TrackMan devices find these rates to produce a carry of 275 yards, on average.

Here is a plot of the curve that a golf ball would make under World Long Drive average ball speed (214 MPH or 95.66 m/s) and spin rate (2,789 rpm)[5]:



Which has **Total Carry = 312.61 Meters (341.87 Yards)**. TrackMan devices find these rates to produce a carry of 352 yards, on average.

Step 4: Dimensionally-Simplified Estimate

To estimate carry distance R without solving the full system of ODEs, we can use physics principles to create a simplified equation. We can start with the case of no lift and no drag:

$$R_0 = \frac{v_0^2}{g}\sin(2\theta)$$

This equation would be used for projectile motion in a vacuum.

Now consider the addition of lift, which is caused mostly by spin on the golf ball. Lift increases the time the ball remains airborne and therefore increases horizontal distance. The lift force is given by:

$$F_L = \frac{1}{2}\rho C_L A v^2$$

To turn this force into an extra distance term, we can use dimensional analysis. Since distance has units of meters, and lift is a force, a good scaling estimate is:

$$\text{lift bonus} \propto \frac{F_L}{mg} \cdot v_0$$

Substituting F_L into this expression gives:

$$\text{lift bonus} = \left(\frac{1}{2} \cdot \frac{\rho A}{mg} \cdot C_L \cdot v_0\right)$$

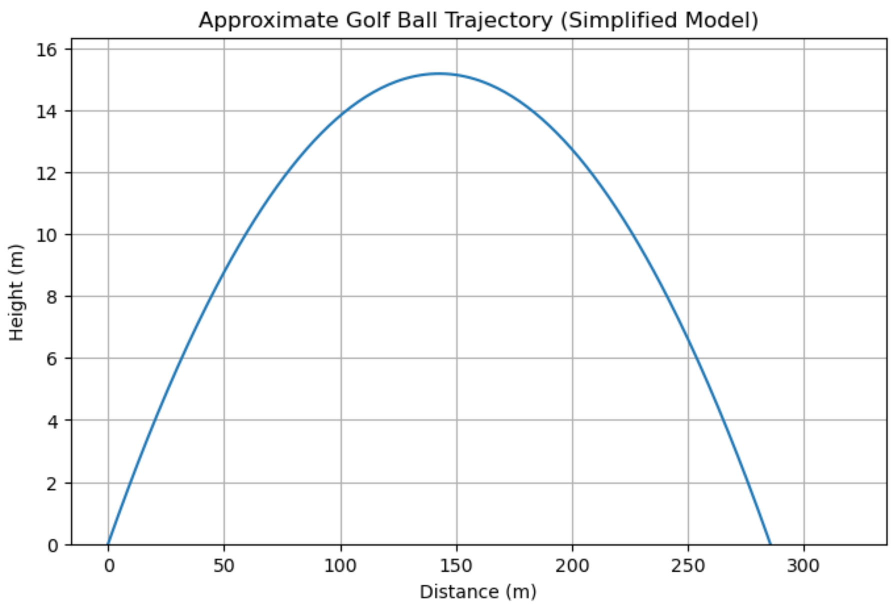
So the full approximate carry distance becomes:

$$R \approx \frac{v_0^2}{g}\left(\sin(2\theta) + \frac{\rho A}{2mg} C_L v_0\right)$$

The first term models standard projectile motion, while the second term accounts for additional range due to lift. This simplified equation allows dominant parameters such as ball speed and lift to determine a rough estimate of carry distance while keeping this to a mid-level of complexity.

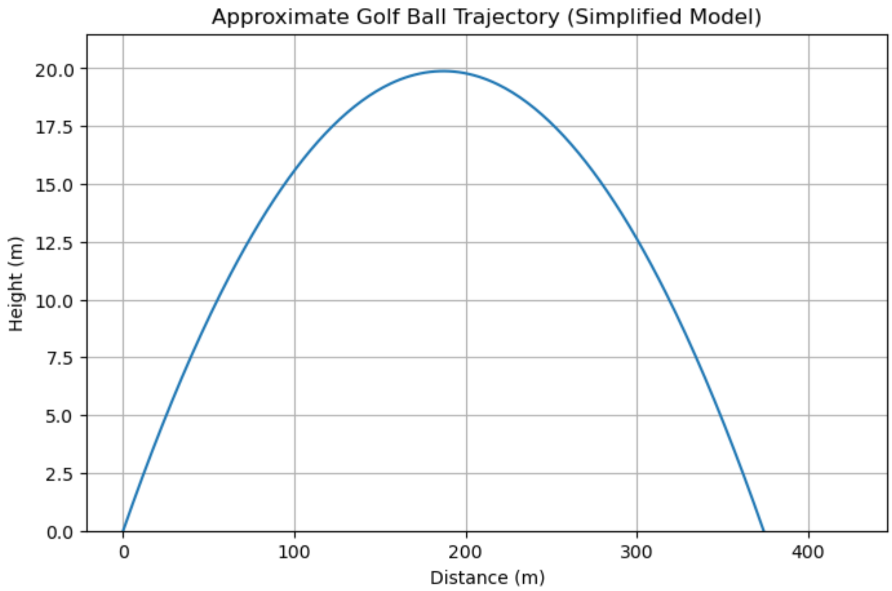
Step 5: Using the simplified equation to create output in Python

Here is a plot of the curve that a golf ball would make under PGA Tour average ball speed (167 MPH or 74.65 m/s) and spin rate (2,686 rpm)[5]:



Which has **Total Carry = 285.15 Meters (311.84 Yards)** TrackMan devices find these rates to produce a carry of 275 yards, on average.

Here is a plot of the curve that a golf ball would make under World Long Drive average ball speed (214 MPH or 95.66 m/s) and spin rate (2,789 rpm)[5]:



Which has **Total Carry = 374.52 Meters (409.58 Yards)**. TrackMan devices find these rates to produce a carry of 352 yards, on average.

Comparison of the two models

Unsurprisingly, the first model was more accurate. It estimated that the average PGA Tour pro would drive 290.11 yards and the average World Long Drive pro would drive 341.87 yards. The PGA estimation is 15 yards longer than TrackMan data shows, and the WLD estimation is roughly 11 yards short.

The simplified model showed PGA pros driving roughly 35 yards further, and WLD pros driving 57 yards further than the TrackMan estimations.

Reasoning through the error

Both models are slightly inaccurate in their calculations. This error can almost definitely be attributed to the assumptions that were made at the start of the model. As mentioned previously, devices like TrackMan use up to 40 different parameters to precisely predict outcomes on the golf course. We used only a handful in this case which is certainly a reason why the numbers produced by either model are not the same as those produced by TrackMan.

For our first model, assuming that environmental conditions are ideal and that only the flight phase should be considered are two large contributors to error. A large portion of a ball's flight is determined pre-contact and also by weather conditions, which is why more powerful, computer-based models like the TrackMan take these things into account.

The simplified model proved to over-estimate carry distances by a significant margin. Taking things like air resistance out of the equation definitely inhibits the accuracy of such a model.

Despite the errors involved, there are times and places where both of these used models can be useful. If you want a more complex model which accounts for things like drag and spin, the first model does a reasonably proficient job at predicting ball flight. If you are just looking for quick, re-producible statistics based on only ideal conditions, the simplified model is a good way to go. In some sense, this model describes what is physically possible if everything is perfect.

References

1. Encyclopdia Britannica. (2024). *Golf.* Retrieved from https://www.britannica.com/sports/golf
2. Penner, R. (2002). *Physics of Golf*, pp. 143-150. Retrieved from http://raypenner.com/golf-physics.pdf
3. TrackMan. (n.d.). *What is a golf launch monitor?* Retrieved from https://www.trackman.com/golf/blog/what-is-a-golf-launch-monitor
4. TrackMan. (January 14th, 2020). *40 TrackMan Parameters.* Retrieved from https://www.trackman.com/blog/golf/40-trackman-parameters
5. TrackMan. *Analysis of TrackMan data gathered at World Long Drive.* Retrieved from https://assets-us-01.kc-usercontent.com/c42c7bf4-dca7-00ea-4f2e-373223f80f76/550ab9f6-5dd3-45b0-b58c-75
R07%20-%20Analysis%20of%20Trackman%20data%20gathered%20at%20World%20Long%20Drive.pdf
6. USGA. *Golf Ball Specifications.* Retrieved from https://www.usga.org/equipment-standards/golf-ball.html

Appendix A: Code for Step 3

```python
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

# --- Physical Constants ---
g = 9.81  # gravity (m/s^2)
rho = 1.225  # air density at sea level (kg/m^3)
r = 0.02135  # radius of golf ball (m)
A = np.pi * r**2  # cross-sectional area (m^2)
m = 0.0459  # mass of golf ball (kg)
Cd = 0.3  # drag coefficient
Cl = 0.2   # lift coefficient (depends on spin)

# Initial Conditions (Different for PGA vs World Long Drive etc)
v0 = 74.655  # initial speed (m/s)
theta = np.radians(11.2)  # launch angle in radians
omega = 2789 * 2*np.pi/60  # spin in rad/s (e.g. 3000 rpm)

# Initial velocity components
v0x = v0 * np.cos(theta)
v0y = v0 * np.sin(theta)

def golf_ode(t, y):
    x, y_pos, vx, vy = y
    v = np.sqrt(vx**2 + vy**2)

    # Drag and lift forces
    Fd = 0.5 * rho * Cd * A * v**2
    Fl = 0.5 * rho * Cl * A * v**2

    # Unit velocity vector
    vx_unit = vx / v
    vy_unit = vy / v

    # Unit vector perpendicular to velocity (lift is perpendicular)
    vx_perp = -vy_unit
    vy_perp = vx_unit

    # Drag components (opposite to velocity)
    ax_drag = -Fd * vx_unit / m
    ay_drag = -Fd * vy_unit / m
```

```python

    # Lift components (perpendicular to velocity)
    ax_lift = Fl * vx_perp / m
    ay_lift = Fl * vy_perp / m

    # Total acceleration
    ax = ax_drag + ax_lift
    ay = ay_drag + ay_lift - g

    return [vx, vy, ax, ay]


# Solve the ODE
y0 = [0, 0, v0x, v0y]
t_span = (0, 10)
t_eval = np.linspace(t_span[0], t_span[1], 1000)

sol = solve_ivp(golf_ode, t_span, y0, t_eval=t_eval, rtol=1e-8)

# Stop the simulation when the ball hits the ground
x_vals = sol.y[0]
y_vals = sol.y[1]
ground_idx = np.argmax(y_vals < 0)  # first time y < 0
x_final = x_vals[ground_idx]
t_final = sol.t[ground_idx]

print(f"Carry Distance: {x_final:.2f} meters")

# Plot Trajectory
plt.plot(x_vals[:ground_idx], y_vals[:ground_idx])
plt.title("Golf Ball Trajectory with Drag and Lift")
plt.xlabel("Distance (m)")
plt.ylabel("Height (m)")
plt.grid()
plt.show()
```

Listing 1: Python code for simulating golf ball trajectory using ODEs

Appendix B: Code for Step 5

```python
import numpy as np
import matplotlib.pyplot as plt

# Constants
g = 9.81
rho = 1.225
r = 0.02135
A = np.pi * r**2
m = 0.0459

# Inputs
v0 = 83  # m/s
theta_deg = 12
theta = np.radians(theta_deg)
Cl = 0.3

# Simplified carry distance formula
def simplified_range(v0, theta, Cl):
    return (v0**2 / g) * (np.sin(2 * theta) + (rho * A * Cl * v0) / (2 * m * g))

# Estimate total carry and flight time
R = simplified_range(v0, theta, Cl)
vx = v0 * np.cos(theta)
flight_time = R / vx


# Print carry distance
print(f"Estimated Carry Distance: {R:.2f} meters ({R * 1.0936:.2f} yards)")


```

```python
# Simulate trajectory using projectile motion
t = np.linspace(0, flight_time, 300)
x = vx * t
y = v0 * np.sin(theta) * t - 0.5 * g * t**2

# Plot
plt.figure(figsize=(8, 5))
plt.plot(x, y)
plt.title("Approximate Golf Ball Trajectory (Simplified Model)")
plt.xlabel("Distance (m)")
plt.ylabel("Height (m)")
plt.grid(True)
plt.ylim(bottom=0)
plt.show()
```

Listing 2: Python code for plotting carry distance using simplified equation