Numerical Methods for Projecting Baseball Trajectories

Tyler Headley, Kenji Horigome

Abstract— **In this project, we utilize a computational approach to modeling baseball trajectories based on physical principles. We implement numerical methods to solve the differential equations that govern baseball flight, accounting for gravity and air resistance. Using Statcast batting data for validation, we demonstrate how our model accurately predicts landing locations from launch parameters. We find that exit velocity and launch angle significantly impact flight paths, with optimal home run parameters occurring at approximately 105-110 mph exit velocity and 38-42° launch angle. Our approach provides interpretable predictions that can inform game strategy, player development, and defensive positioning.**

I. INTRODUCTION

The flight trajectory of a baseball after contact is determined by a combination of physical forces, initial conditions, and environmental factors. Understanding these trajectories has applications in baseball analytics, player development, and game strategy. In this paper, we develop a computational model for baseball flight paths using physics-based principles and numerical methods to provide predictions of landing locations based on launch parameters.

As the size and complexity of baseball data continue to grow, computational approaches have become essential for yielding actionable insights. Modern baseball analysis has evolved from simple statistics to sophisticated tracking systems that capture detailed information about every play. Statcast, implemented across all Major League Baseball (MLB) stadiums since 2015, uses high-resolution cameras and radar to measure previously unquantifiable aspects of the game.

Our project draws on this data to implement numerical methods to solve the differential equations that describe baseball flight. We focus primarily on the effects of gravity and air resistance, which are the dominant forces affecting trajectory. By simulating how these forces interact with the ball's initial conditions—specifically exit velocity, launch angle, and spray direction—we can predict landing locations and flight characteristics. We employ a Runge-Kutta-Fehlberg method (RK45), which provides an adaptive step-size approach to numerical integration, balancing accuracy and computational efficiency.

This model offers the advantage of being physically interpretable, which is beneficial to explain how certain launch parameters lead to specific outcomes. The model's applications extend to multiple aspects of baseball, including player development, game strategy, and equipment design. This work contributes to the growing intersection of sports, physics, and computational methods, applying classical mechanics for practical applications with America's pastime.

II. MATHEMATICAL FOUNDATIONS

In this section, we present the mathematical foundations of baseball trajectory modeling. We begin with the fundamental equations of motion that govern projectile flight, incorporate air resistance effects, and describe the numerical methods used to solve these equations.

A. Physical Model

The motion of a baseball can be described by a system of differential equations based on Newton's laws of motion. We model the baseball as a point mass subject to gravitational and aerodynamic forces. The state of the system at any time is represented by the position vector $\mathbf{r} = (x, y, z)$ and the velocity vector $\mathbf{v} = (v_x, v_y, v_z)$.

1) Gravitational Force Model: The simplest model considers only gravity, represented by the constant acceleration vector $\mathbf{g} = (0, 0, -g)$, where $g \approx 9.81$ m/s^2. This leads to the differential equation:

$$\frac{d\mathbf{v}}{dt} = \mathbf{g} \tag{1}$$

Although this basic model provides a reasonable approximation for short distances and low velocities, it fails to account for the significant aerodynamic effects that influence baseball flight at game-typical speeds of 70-110 mph. In the absence of air resistance, the trajectory would follow a perfect parabola, which is not observed in actual baseball trajectories.

2) Incorporating Drag Forces: For accurate trajectory prediction, we must account for air resistance (drag), which opposes the motion and is proportional to the square of velocity. Incorporating drag, the equation becomes:

$$\frac{d\mathbf{v}}{dt} = \mathbf{g} - \frac{1}{2}\rho C_D \frac{A}{m} V \mathbf{v} \tag{2}$$

where:

- ρ is the air density (approximately 1.225 kg/m^3 at sea level)
- C_D is the drag coefficient (typically 0.2-0.5 for a baseball)
- A is the cross-sectional area of the baseball (πr^2, where $r \approx 0.037$ m)
- m is the mass of the baseball (approximately 0.145 kg)
- $V = \|\mathbf{v}\|$ is the magnitude of the velocity

For example, drag forces using this model on flying balls have been studied using numerical methods in [4]. The drag coefficient C_D is an important parameter that varies based on several factors, including Reynolds number, ball surface roughness, and potentially air conditions. Research by Adair [1] and Nathan [2] suggests that for typical baseball velocities, C_D generally falls between 0.2 and 0.5, with lower values corresponding to smoother balls or conditions that promote laminar flow.

The drag force is proportional to the square of velocity, making it particularly significant for well-hit balls. At high exit velocities (>100 mph), drag can reduce the distance traveled by 20-30% compared to the idealized parabolic trajectory of the gravity-only model. This substantial effect shows importance of including air resistance in any realistic model of baseball flight. This is the model we employ for our computational experiments.

3) Advanced Modeling Considerations: For even more accurate models, additional factors could be incorporated:

a) Magnus Force: The spin of the baseball creates a pressure differential perpendicular to both the velocity and spin axis, resulting in lateral movement. For a spinning baseball, this can be modeled as:

$$\mathbf{F}_{\text{Magnus}} = \frac{1}{2}\rho C_L(S)\frac{A}{m}V(\hat{\omega}\times\mathbf{v}) \tag{3}$$

where $C_L(S)$ is the lift coefficient (a function of spin rate S), and $\hat{\omega}$ is the unit vector along the spin axis. This factor is modeled by [3] for baseball pitch trajectories, which experience even greater Magnus forces due to ball spin.

b) Environmental Factors: Wind velocity, altitude, temperature, and humidity all affect trajectory through their influence on air density and drag. Air density decreases approximately 3% per 1,000 feet increase in altitude, with corresponding reductions in drag force. This explains the well-known "Coors Field effect" at the Colorado Rockies' stadium, situated at 5,280 feet above sea level, where batted balls travel much farther than at sea-level parks.

c) Ball Construction: Recent controversies in MLB regarding "juiced" and "de-juiced" balls highlight the sensitivity of trajectories to small changes in ball construction. Seam height, internal composition, and surface properties can all affect drag coefficient and carry distance.

B. Numerical Methods

The system of differential equations describing baseball flight does not have a closed-form analytical solution due to the nonlinear drag term. Therefore, we employ numerical integration techniques to approximate the solution. In this section, we discuss the limitations of simpler methods and our rationale for selecting a higher-order adaptive algorithm.

1) Euler's Method and Its Limitations: The simplest approach to solving our differential equations would be Euler's method, a first-order numerical technique that approximates the solution using:

$$y_{i+1} = y_i + hf(t_i, y_i), \quad i = 0, \ldots, n-1 \tag{4}$$

where h is a fixed step size and $f(t, y)$ represents the right-hand side of our differential equation. For the baseball trajectory problem, y would contain the six state variables (three position and three velocity components).

While conceptually straightforward, Euler's method has limitations for our application:

- It requires small step sizes to maintain accuracy for rapidly changing functions
- It cannot handle the stiffness potentially introduced by the nonlinear drag terms
- The global error is proportional to the step size ($O(h)$), making it computationally expensive to achieve high accuracy

These limitations motivate the use of higher-order methods with adaptive step size control.

2) Runge-Kutta Methods: Runge-Kutta methods improve upon Euler's method by evaluating the function at multiple points within each step interval. The general form of a Runge-Kutta method can be represented using a Butcher tableau:

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} \tag{5}$$

where c contains the nodes (evaluation points), A is a matrix defining how intermediate values are combined, and b contains the weights for the final combination.

For example, the classic fourth-order Runge-Kutta method (RK4) uses four function evaluations per step:

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \tag{6}$$

This method provides significantly better accuracy than Euler's method, with global error of $O(h^4)$, but still uses a fixed step size. RK4 is employed by [5] for modeling golf ball hit trajectories.

3) Adaptive RK45 Method: For our baseball trajectory simulations, we employ the Runge-Kutta-Fehlberg 4(5) method (RK45), implemented via SciPy's `solve_ivp` function. This method combines a 4th-order and a 5th-order Runge-Kutta method to provide both a solution and error estimate for adaptive step size control.

The adaptation strategy uses the error estimate to automatically adjust the step size:

$$h_{\text{new}} = h_{\text{current}} \cdot \left(\frac{\epsilon \cdot y_{\text{current}}}{||error||}\right)^{1/5} \tag{7}$$

where ϵ is the desired error tolerance. This approach allows large steps where the solution changes slowly (e.g., near the apex of the trajectory) and small steps where accuracy is critical (e.g., during rapid changes in velocity due to drag).

The general form of the RK45 step is:

$$y_{n+1} = y_n + h\sum_{i=1}^{6} b_i k_i \tag{8}$$

where k_i are intermediate slope evaluations:

$$k_1 = f(t_n, y_n) \quad (9)$$
$$k_2 = f(t_n + c_2 h, y_n + h(a_{21}k_1)) \quad (10)$$
$$k_3 = f(t_n + c_3 h, y_n + h(a_{31}k_1 + a_{32}k_2)) \quad (11)$$
$$\vdots \quad (12)$$

The 4th-order approximation uses one set of b_i coefficients, while the 5th-order solution uses another set. The difference between these two approximations provides the error estimate. The adaptive step size control makes RK45 well-suited for baseball trajectory simulation, providing a good balance between accuracy and computational efficiency.

III. EXPERIMENTS

In this section, we test our model with various input parameters, validate against Statcast data, and analyze the effects of physical parameters on trajectory outcomes.

A. Sample Trajectories

We simulated three archetypal batted ball types to validate the qualitative behavior of our model and explore how different launch parameters affect trajectory characteristics. This analysis focuses on the effect of exit velocity, launch angle, and drag effects for batted ball outcomes including a home run, line drive, and ground ball.

1) Home Run Trajectory: For the home run simulation, we selected parameters representative of a well-hit ball in MLB competition:

- Exit velocity: 115 mph (51.4 m/s)
- Launch angle: 27° (near the empirically observed optimum)
- Spray angle: 0° (straightaway center field)
- Drag coefficient: 0.35

The resulting trajectory exhibited the following characteristics:

- Distance: 119.9 m (393.2 ft)
- Maximum height: 20.5 m (67.3 ft)
- Flight time: 4.02 s
- Terminal velocity: approximately 41.8 m/s at landing

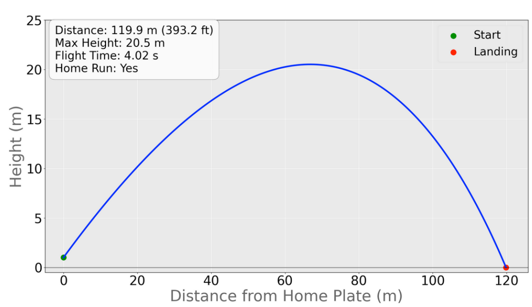

Fig. 1. Side view (y-z plane) trajectory of a simulated home run with exit velocity of 115 mph and launch angle of 27°. The curve demonstrates the characteristic asymmetric shape of a drag-affected projectile, with a steeper descent than ascent due to continuous deceleration from air resistance.

Figure 1 shows the side view (y-z plane) of this trajectory. The curve demonstrates the characteristic asymmetric shape of a drag-affected projectile, with a steeper descent than ascent. The asymmetry is due to drag forces continuously reducing the ball's velocity, causing it to travel less distance in the descent phase than would be expected in a vacuum.

The launch angle of 27° creates substantial vertical displacement, allowing the ball to clear a standard outfield fence (typically 2.4 m or 8 ft tall). The 393.2 ft distance exceeds our home run threshold of 350 ft, which approximates the minimum distance to clear the fence in most MLB stadiums.

2) Line Drive Trajectory: For the line drive simulation, we used parameters typical of solid contact with less vertical launch:

- Exit velocity: 95 mph (42.5 m/s)
- Launch angle: 12° (characteristic of a line drive)
- Spray angle: -10° (slightly pulled to right field)
- Drag coefficient: 0.35

The simulation yielded:

- Distance: 61.0 m (200.0 ft)
- Maximum height: 4.5 m (14.8 ft)
- Flight time: 1.80 s
- Terminal velocity: approximately 36.3 m/s at landing

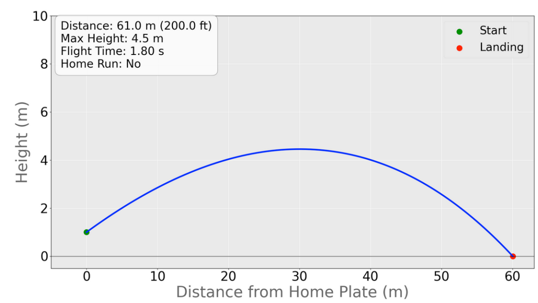

Fig. 2. Trajectory of a simulated line drive with exit velocity of 95 mph and launch angle of 12°. The maximum height of 4.5 m is sufficient to clear infielders but stays well below outfield fence height for most of its path.

Figure 2 illustrates this trajectory, showing the much flatter path characteristic of line drives. The maximum height of 4.5 m is sufficient to clear infielders but stays well below outfield fence height for most of its path. The 200 ft distance places this hit in the shallow outfield, typical of a single in game situations.

3) Ground Ball Trajectory: For the ground ball simulation, we selected parameters representing weak contact with a downward angle:

- Exit velocity: 88 mph (39.3 m/s)
- Launch angle: -5° (negative angle directed toward ground)
- Spray angle: 15° (opposite field)
- Drag coefficient: 0.35

This produced:

- Distance: 8.6 m (28.3 ft)
- Maximum height: 1.0 m (3.3 ft)
- Flight time: 0.23 s

Figure 3 shows this trajectory, which is characterized by a brief flight phase before the ball contacts the ground. Unlike

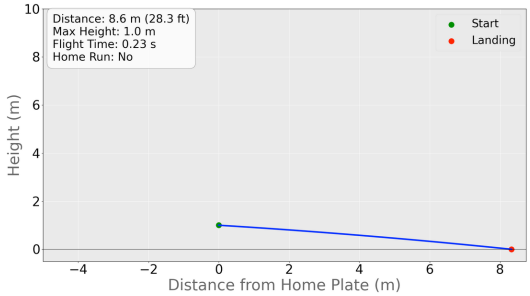

Fig. 3. Trajectory of a simulated ground ball with exit velocity of 88 mph and negative launch angle of -5°. The negative launch angle directs the ball immediately downward, with the initial vertical velocity component accelerated further by gravity.

the previous cases, this trajectory is primarily determined by the initial conditions rather than aerodynamic effects, as the ball is in flight for only 0.23 seconds before first bounce.

The three trajectory types described above demonstrate different path characteristics based on their launch parameters.

B. Parameter Studies

1) Effect of Drag Coefficient: We investigated how drag coefficient affects baseball trajectories by simulating a ball hit with fixed initial conditions (40 m/s, 25° launch angle) while varying the drag coefficient from 0.2 to 0.6. We found that lower drag coefficients result in significantly longer distances:

- $C_D = 0.2$: Distance = 98.1 m
- $C_D = 0.3$: Distance = 88.7 m
- $C_D = 0.4$: Distance = 81.2 m
- $C_D = 0.5$: Distance = 75.1 m
- $C_D = 0.6$: Distance = 70.0 m

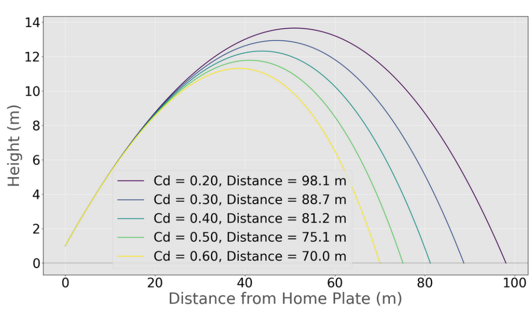

Fig. 4. Effect of varying drag coefficient on baseball trajectory. All trajectories use the same initial conditions (exit velocity = 40 m/s, launch angle = 25°), with only the drag coefficient changing. Lower drag coefficients result in significantly increased distance and altered trajectory shape.

Figure 4 illustrates how varying the drag coefficient affects trajectory while keeping all other parameters constant. The substantial variation in distance highlights the importance of accurately modeling aerodynamic effects in baseball trajectory prediction.

2) Effect of Launch Angle: We also examined how launch angle affects distance by simulating trajectories with fixed exit velocity (45 m/s) and drag coefficient (0.35) while varying launch angles from 34° to 46°.

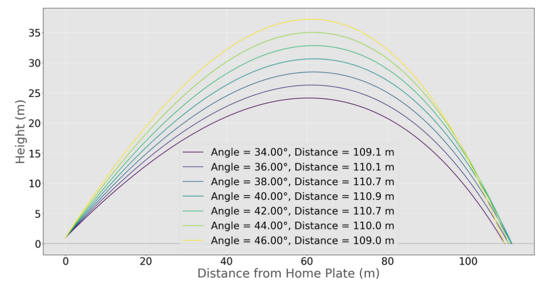

Fig. 5. Effect of varying launch angle on baseball trajectory while maintaining constant exit velocity (45 m/s) and drag coefficient (0.35). The optimal launch angle for maximum distance is approximately 40°, with a relatively flat optimality curve between 38° and 42°.

Figure 5 shows the effect of launch angle on trajectory. The results demonstrate the existence of an optimal launch angle for maximizing distance, which depends on both exit velocity and drag characteristics. We found the optimal launch angle range to be between 38° and 42°, with our maximum distance achieved at 40° (110.9 m). Although this might be the optimal launch angle for our model with all else equal, player biomechanics likely make it difficult to swing upwards in this manner, explaining why the general recommendation is considered to target 25°-35°.

C. Model Validation

To assess our model's predictive capabilities, we validated it by comparing simulation results against empirical data from MLB games. For this study, we used a random sample of 117 Statcast batted balls from the 2024 MLB regular season, taking the relevant initial conditions and simulating the ball flight with our model. Statcast also provides data for the x and y "hit coordinate" of the ball, giving us a way of comparing our models prediction with actual data. However, this hit coordinate indicates where the ball ended up once it was fielded rather than where it first made contact with the ground, introducing a significant source of error in our comparison, especially for balls that roll much further than where they land such as hard hit ground balls. Thus, we restricted our sample to balls that were hit far into the outfield and were likely caught, giving us a more accurate baseline to compare off of. We then compare the distance of the simulated and true landing coordinates and compute the mean absolute error (MAE) as a metric for how far off our model was.

Figure 6 shows a scatter plot visualization of predicted versus actual landing positions. Our model achieves a mean absolute error of 4.39 m difference in the landing coordinates, a significant improvement from the MAE of 16.87 m on the unrestricted sample containing ground balls. The remaining prediction discrepancy is likely a result of a combination of unaccounted-for physical forces, weather conditions, and remaining error in our dataset ground truth.

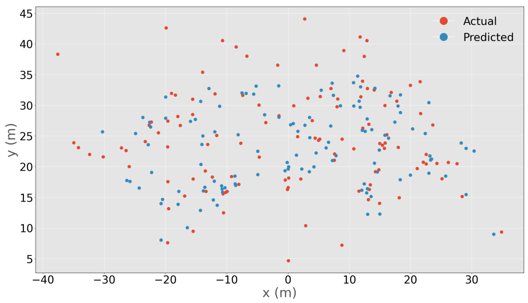

Fig. 6. Comparison of predicted (blue) and actual (red) landing positions for 1,000 randomly sampled batted balls from the 2024 MLB season. The coordinate system origin represents home plate, with the positive y-axis pointing toward center field and the positive x-axis pointing toward left field.

IV. CONCLUSIONS

In this paper, we have tested a computational approach to modeling baseball trajectories using numerical methods. By solving the differential equations that govern baseball flight, we capture the effects of gravity and air resistance on batted balls.

Our findings highlight the importance of launch parameters, particularly exit velocity and launch angle, in determining trajectory outcomes. The optimal combination of these parameters—exit velocity of 105-110 mph and launch angle of 38-42°—maximizes distance and increases the likelihood of home runs.

The model's validation against Statcast data shows reasonable accuracy considering the complexity of real-world baseball flight. While not capturing all factors affecting trajectories, our approach provides a physically interpretable method for understanding batted ball outcomes.

Future work could include incorporating Magnus forces to account for ball spin, adding environmental factors such as wind, altitude, temperature, and humidity, and fitting our model to the data in order to retrieve a more accurate estimate for the baseball's drag coefficient.

REFERENCES

[1] R. K. Adair, The Physics of Baseball, 3rd ed. New York: Harper Perennial, 2002.

[2] A. M. Nathan, "The effect of spin on the flight of a baseball," American Journal of Physics, vol. 76, no. 2, pp. 119-124, 2008.

[3] Sanie, B. The Perfect Pitch: Baseball Meets Computational Physics. Georgia Institute of Technology, 2023.

[4] Liu, X. Ordinary Differential Equations (ODEs) Project (MATLAB). CSC 160, 2011.

[5] Mohammadzadeh, A. Learning from a Golf Ball. Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition, 2004.