# Baseball Pitch Trajectory Project

Olivia Bernstein

November 2, 2016

# 1 Introduction

Since 2007, all major league baseball pitches have been analyzed using the Sportvision technology PITCHf/x. Digital cameras track each pitch, and the data is analyzed to calculate the speed, position, and trajectory. Sportvision claims that "the PITCHf/x®service tracks and digitally records the full trajectory of live baseball pitches to within an inch of accuracy" (Citation 1). However, because this technology assumes that acceleration is constant over the entire flight of the pitch, it does not account for forces caused by lift and drag on the ball, which affect the ball's trajectory.

Another method of computing the trajectory of the baseballs with reduced error has been proposed by physicist Alan Nathan. He demonstrated how to extract the coefficients of drag and lift, values that will assist in a more accurate position calculation, from the PITCHf/x data (Citation 2). Using these coefficients, we will compute the trajectory of a pitch by solving a system of differential equations. By calculating the basic trajectory proposed by Sportvision, calculating the more accurate trajectory using a fourth-order Runge-Kutta procedure, and then comparing the results, we will be able to estimate the error accrued by the Sportvision calculation. We will do this with two different pitches, one made by Barry Bonds in 2007, and the other made by Blake Trienen in 2016.

# 2 Methods

## 2.1 Sportvision Calculation

The Sportvision paper provides a model (equation 1) that they claim reproduces the trajectory of the pitch. We used their equations to in an attempt to replicate the path of the ball for two different pitches, Bonds' pitch (Pitch 1) and Trienen's pitch (Pitch 2). We analyzed these two pitches and used the initial conditions provided by PITCHf/x for position, velocity, and acceleration (Table 1).

$$\mathbf{x}_m(t) = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix} + (t - t_0) \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} + \frac{1}{2}(t - t_0)^2 \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix} \tag{1}$$

The coordinate system describing the ball's position and motion assigns the y-direction from the batter to the pitcher, the z-direction as vertical motion and points upward, and the x-direction as horizontal motion, with the positive direction to the catcher's right. To find the trajectory we wanted to calculate the position as time progressed until the ball reaches the front of home plate, where y = 1.417 feet. The equation uses initial values for position, velocity, acceleration, and time. A time vector ($t$) begins at $t_0 = 0$ and is incremented from $t_0$ until the time where y = 1.417 to model the pitch. We used an initial value for time a step size of $10^{-7}$ for our calculations.

Table 1: Initial Conditions (ft)

| Pitch | $X_0$ | $Y_0$ | $Z_0$ | $V_x$ | $V_y$ | $V_z$ | $A_x$ | $A_y$ | $A_z$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | -2.509 | 50 | 5.928 | 9.182 | -132.785 | -10.967 | -19.268 | 30.713 | -16.580 |
| 2 | -2.43 | 50 | 6.46 | 9.46 | -143.17 | -9.15 | -23.08 | 34.2 | -26.09 |

To find the time where the ball was at the front of the home plate, we started by using a large time span with a final time of 2 seconds, but this allowed the ball to go beyond home plate and the y-values progressed into negatives values. We used this calculation to find the point in time where the y-position of the ball was at 1.417 feet. Next, to allow the final position to be at the front of home plate, we ran the same calculation and ended it at the final time we had determined.

The final positions of both pitches were calculated with this method and can be seen in Table 2 and Figure 1. The path is plotted in XZ space which is the ball's path when observed from the batter's perspective, and "X" marks the final position at the front of the plate. The whole pitch has been plotted in 3-space in Figure 2.

Table 2: Final Position Calculated by Sportvision (ft)

| Pitch | $X_0$ | $Y_0$ | $Z_0$ |
|---|---|---|---|
| 1 | -0.406 | 1.417 | 0.515 |
| 2 | -0.527 | 1.417 | 1.580 |

## 2.2 Runge-Kutta Calculation

A Runge-Kutta of the fourth order (RK4) is a very powerful calculation is utilized in solving systems of differential equations. In Alan Nathan's paper, three differential equations for the position of the ball are given (Equations 2-4). In these equations, $K$, $C_D$, $C_L$, and $\phi$ are considered to be constants, where $C_D$and $C_L$ are the coefficients of drift and lag, respectively. The $v_x$,$v_y$, and $v_z$ describe the velocity of the ball along its trajectory, while $x$, $y$, and $z$ describe its position. These values change as the ball moves on its path due to the

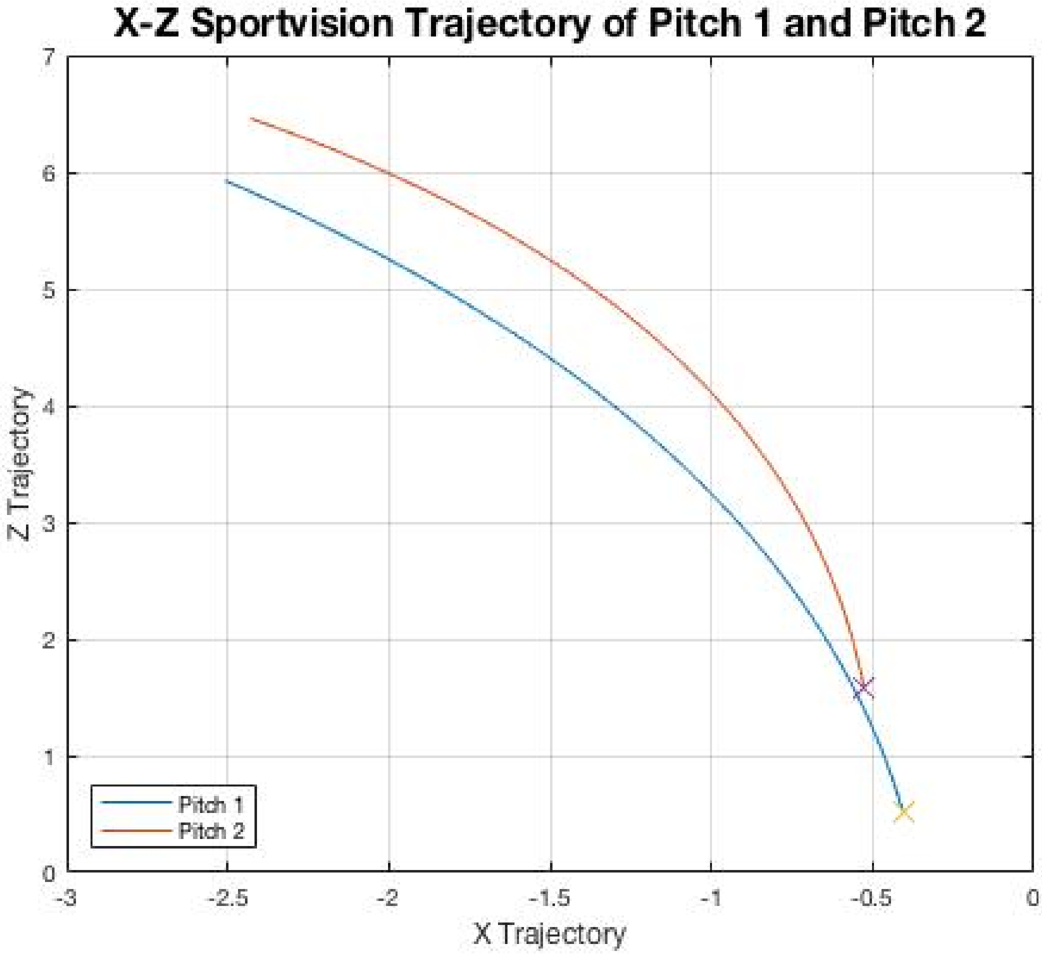


Figure 1: This plot shows the x- and z-trajectory of the pitch 1 (blue line) and pitch 2 (orange). This would be the view the batter would see of the path that the ball traveled. The orange mark at the end of each curve shows the final position.

different forces acting on the ball.

$$x = -KC_D v v_x - KC_L v v_y \text{sin`}\phi \tag{2}$$
$$y = -KC_D v v_y + KC_L v(v_x \text{sin}\phi - v_x \text{cos}\phi) \tag{3}$$
$$z = -KC_D v v_z + KC_L v v_y \text{cos}\phi - g \tag{4}$$

In order to use the RK4 method, the equations must first be put into vectorized notation. This form places all the variables in one vector, and each time the value for time is updated, a new value is calculated for each of these variables. The differential equations are solved simultaneously, and the calculation stops when a certain value is reached. In our case, the return condition is when the ball position on the y-axis is at or past the front of home plate. The final position of the ball is the $x$, $y$, and $z$ coordinates at this point (see Table 3).

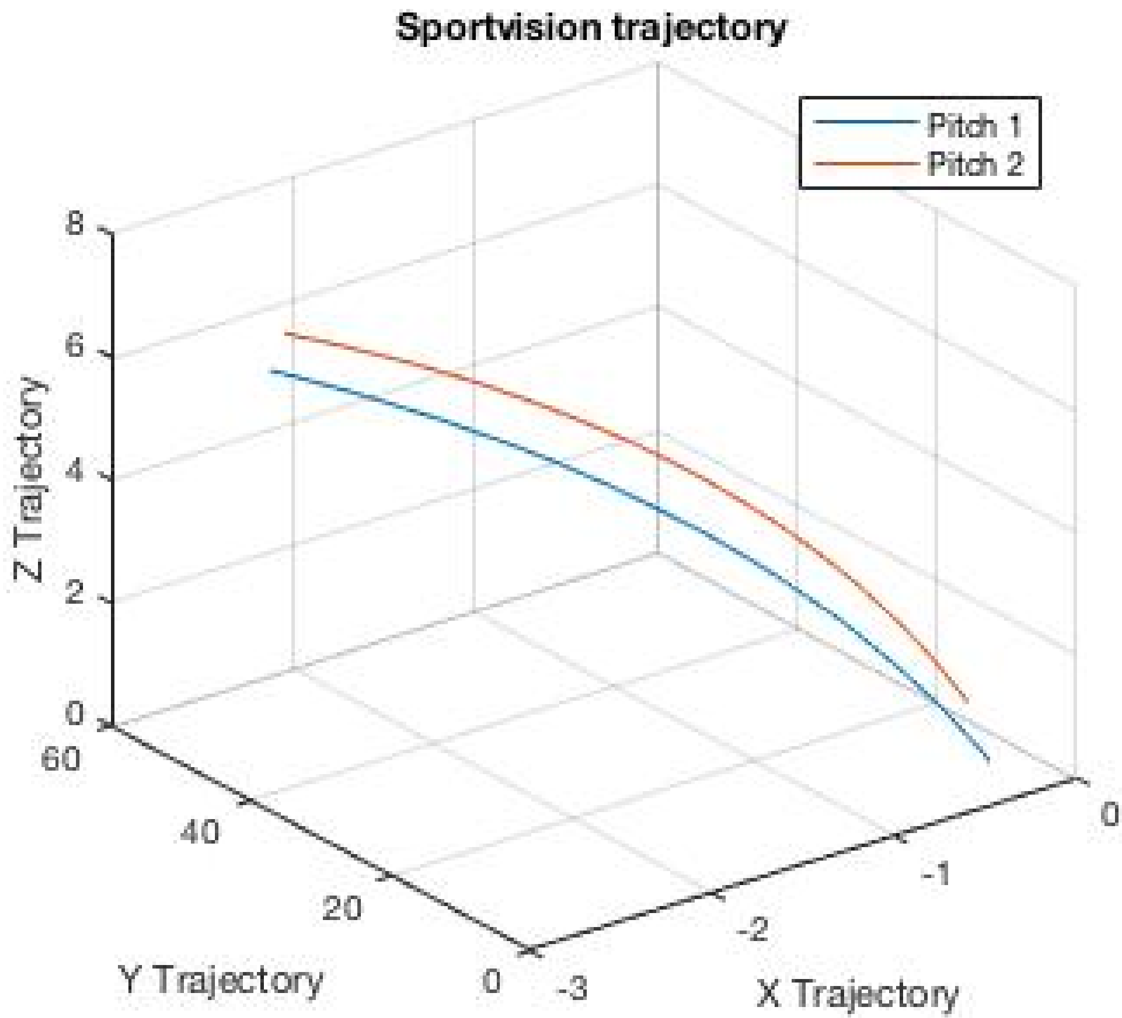


Figure 2: This plot shows the 3D trajectory of the pitch 1 (blue line) and pitch 2 (orange).

# 3 Results

## 3.1 Barry Bonds Pitch

After calculating the pitch trajectory using both the Sportvision equations and Alan Nathan?s aerodynamic equations, we achieved a different position vector for each pitch. The results for Barry Bonds' pitch can be seen in the table below (Table !!!!!). The largest percent difference between the Sportvision results and the ((other)) results is ADD IN. This corresponds to ADD IN inches, which means that the [Sportvision accuracy claims were incorrect OR that this pitch was within Sportvision?s margin of error it depends on the result we get.]

Table 3: Final Position Calculated by RK4 (ft)

| Pitch | $X_0$ | $Y_0$ | $Z_0$ |
|---|---|---|---|
| 1 | -0.456 | 1.417 | 0.514 |
| 2 | -0.578 | 1.417 | 1.561 |

Table 4: Pitch 1: Final Positions

| Sportvision | $RK_4$ | % Difference |
|---|---|---|
| x | x | |
| y | y | |
| z | z | |
| $\sqrt{x^2+y^2+z^2}$ | $\sqrt{x^2+y^2+z^2}$ | |

### 3.2 Blake Trienen Pitch

Similarly, we calculated two different trajectory paths for Blake Trienen's pitch. As can be seen in the table below (!!!Table !!!), this pitch had, on average, [less OR more OR about the same amount of] percent difference between the Sportvision calculation and the RK4 calculation when compared to Barry Bond?s pitch. (We can say something about the very high Quality of Pitch that Sportvision gave it, I think Dr. Wilson said 9.99, and remark on this. If this one has less error, or more, etc

Table 5: Pitch 2: Method Comparison

| Sportvision | $RK_4$ | % Difference |
|---|---|---|
| x | x | |
| y | y | |
| z | z | |
| $\sqrt{x^2+y^2+z^2}$ | $\sqrt{x^2+y^2+z^2}$ | |

# 4 Conclusion

# 5 Matlab Scripts

### 5.1 Sportvision Calculations

```
%%pitch 1
pos1 = [-2.509; 50; 5.928];
```

```
vel2 = [9.182; -132.785; -10.967];
acc2 = [-19.268; 30.713; -16.580];
t0 = 0;

f1 = @(t) pos1 + (t-t0).*vel2 + .5.*(t-t0).^2.*acc2;
t1 = [0:.0000001:0.3803727];
posn_f1 = f1(t1);
display = [posn_f1;t1]; %used to find t value corresponding to y = 1.417
plot3(posn_f1(1,:),posn_f1(2,:),posn_f1(3,:));
xlabel('X'); ylabel('Y'); zlabel('Z')
title('Sportvision trajectory')
grid on
n = length(posn_f1(1,:));
final_posn = posn_f1(:,3803728);
final_posn

%%2016 pitch
pos2 = [-2.43;50;6.46];
vel2 = [9.46;-143.17;-9.15];
acc2 = [-23.08;34.2;-26.09];

f2 = @(t) pos2 + (t-t0).*vel2 + .5.*(t-t0).^2.*acc2;
t2 = [0:.0000001:0.3520677];
posn_f2 = f2(t2);
display = [posn_f2;t2]; %used to find t value
plot3(posn_f2(1,:),posn_f2(2,:),posn_f2(3,:));
xlabel('X'); ylabel('Y'); zlabel('Z')
title('Sportvision trajectory')
grid on

%%plot both pitches
hold on
plot3(posn_f1(1,:),posn_f1(2,:),posn_f1(3,:));
hold off

%%plot xy side to make sure they were both starting at y = 50
plot(posn_f1(1,:),posn_f1(2,:))
hold on
plot(posn_f2(1,:),posn_f2(2,:))
grid on
xlabel('X Trajectory'); ylabel('Y Trajectory');
title('X-Y Sportvision Trajectory of Pitch 1 and Pitch 2')

%YZ trajectory
hold off
```

```
plot(posn_f1(2,:),posn_f1(3,:))
hold on
plot(posn_f2(2,:),posn_f2(3,:))
grid on
xlabel('Y Trajectory'); ylabel('Z Trajectory');
title('Y-Z Sportvision Trajectory of Pitch 1 and Pitch 2')
```