EFFECT OF SEAM HEIGHT ON BASEBALL FLIGHT

by

Annaliese Reedholm

Submitted in partial fulfillment of the
requirements for Departmental Honors in
the Department of Engineering
Texas Christian University
Fort Worth, Texas

May 4, 2015

EFFECT OF SEAM HEIGHT ON BASEBALL FLIGHT

Project Approved:

Supervising Professor: Robert Bittle, Ph.D.
Department of Engineering

Walt Williamson, Ph.D.
Department of Engineering

Efton Park, Ph.D.
Department of Math

ABSTRACT

The National Collegiate Athletic Association (NCAA) Division I baseball committee voted to reduce the seam height for baseballs from 0.048 inches to 0.031 inches beginning in the 2015 baseball season. The NCAA claimed that the ball would travel further while maintaining player safety. To test these claims, two balls were donated from the 2014 baseball season and two balls from the 2015 baseball season from Texas Christian University's Division I baseball team. A ball from each season was tested in the two-seam configuration and the four-seam configuration. Using wind tunnel analysis, the forces that act on the baseballs were calculated to model flight path. Different initial conditions were simulated by the model to verify these claims. First, the model simulated a home run returning from the bat at 95 miles per hour, 1400 revolutions per minute at 25 degrees from the horizontal. On average, the new ball traveled 18 feet farther than the old one did. Next, the model simulated a well-hit line drive at 115 miles per hour at 1 degree above the horizontal. On average, the new ball travels half of a millisecond faster. Thus, the model proved the NCAA claims hold: the new ball travels further while maintaining pitcher safety. Furthermore, the new ball travels half of a millisecond faster for a well-hit line drive. Since half a millisecond is considered negligible for reaction time, the model proved the NCAA claims hold: the new ball travels further while maintaining pitcher safety.

ACKNOWLEDGEMENTS

It is with extreme gratitude that I acknowledge the immense support and help of my professor, Dr. Robert Bittle. His passion for and understanding of baseball allowed me to understand and create a successful model. He continually pushed me to improve my model and carry on through the project. I would not have been able to complete this project without his continuous support, patience, and understanding. I would also like to extend thanks to the rest of my honors committee: Dr. Efton Park and Dr. Walt Williamson. I am extremely grateful for their willingness to serve on my committee and the work that it involves. I have had each one of my committee members as a professor in multiple classes and have enjoyed each one.

I would also like to thank the Department of Engineering at Texas Christian University. I have had such an amazing experience throughout my four years due to the continuous support and kindness of the staff. In particular, I would like to thank Dr. Becky Bittle for giving academic and life advice and Ms. Teresa Berry for providing great kindness to every student, including treats whenever I needed it. Additionally, I would like to thank David Yale and Mark Roegels of the machine shop for always managing to be available when I needed help, particularly with mounting the baseballs.

I would also like to thank Kevin Case, Jon Langston, and Blake Williams for their willingness to help with the dropped ball test. They helped in their spare time without a single complaint.

Finally, and most importantly, I would like to thank my personal support system. I thank Sean Epke. He was always there through the stressful times of the project and offered endless support. I also thank my siblings, Colleen, Caitlin, and Paul. They always know when to provide laughter or motivation . Lastly and most importantly, I thank my parents, Joe and Katherine, for always supporting me and encouraging me to be my best. I would not be the person I am today without their continuous love or the drive they instilled in me.

TABLE OF CONTENTS

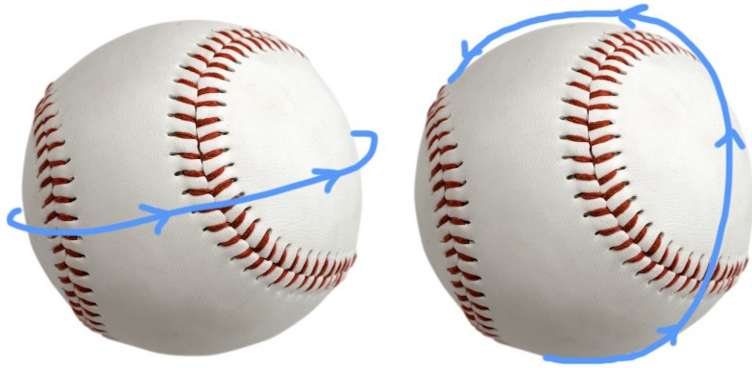
LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

On November 11, 2013, the National Collegiate Athletic Association (NCAA) Division I baseball committee came to a unanimous vote to allow conferences to adopt a new baseball for regular-season play in 2015. Prior to this vote, the NCAA used a raised seam baseball with a seam height of 0.048 inches. The new ball has a flatter seam ball with a seam height of 0.031 inches. The Washington State University Sports Science Laboratory conducted research on the NCAA's behalf and found that flat-seamed baseballs traveled approximately 20 feet farther than the raised-seamed baseballs when launched at an initial condition of a 25-degree angle, at 95 miles per hour, and with a 1,400 revolutions per minute back-spin rate. Additionally, it was claimed that the safety of the players would not be compromised (Johnson 1).

In order to validate the previous research, four baseballs were obtained from the Texas Christian University (TCU) NCAA Division I baseball team. Two baseballs were from the 2014 season with raised seams (high seams) and two were from the 2015 season with the lower seams. Two common pitching configurations, a two seam and a four seam, were tested in the TCU Department of Engineering Wind Tunnel from the 2014 (high seam) and 2015 (low seam) season as seen in Figure 1. For the two seam configuration, the thrown baseball rotates about its spin axis with only two seams interacting perpendicularly with the passing air. In the four seam configuration, the ball rotates about its spin axis with four seams interacting perpendicularly with the passing air.



2 Seam Configuration 4 Seam Configuration

Figure 1 - 2 Seam and 4 Seam Configuration

Each baseball was tested in the TCU Wind Tunnel in a static (non-rotating) position. Analysis of the wind tunnel tests provided the drag coefficient at velocities ranging from 65 to 180 feet per second (45 to 123 miles per hour).

Figure 2 - TCU Wind Tunnel

In conjunction with the wind tunnel results, engineering and physics principles were used to model the flight path of each baseball. The model calculates instantaneous velocity along the flight path including horizontal and vertical distances, as well as the projection angle through the flight.

Various iterations of the model can be run by adjusting the initial conditions of velocity, angle, and spin. In this analysis, several conditions were tested. First, a drop ball case was used to confirm the static results from the wind tunnel and model. Next, the conditions tested by Washington State University Sports Science Laboratory, which is certified by the NCAA, were used to validate the model. Finally, a well-hit line drive was simulated to test the time for a ball to return to the pitching mound, thus assessing changes to the pitcher's safety. Examining the flight path, as well as flight duration, allowed direct analysis of the claims.

THEORY

The flight path of a baseball can be modeled based on key engineering and physics principles.

Scalars and Vectors

Before delving into specific concepts, it is important to understand scalars and vectors. A scalar is a constant number specified only by magnitude, such as mass and length. A vector is quantity that has a direction as well as a magnitude, such as velocity and acceleration. Graphically, a vector is shown by an arrow with a tip (pointed) and tail. Commonly, two-dimensional vectors are broken into Cartesian coordinates x- and y-components using trigonometry as seen in Figure 3.

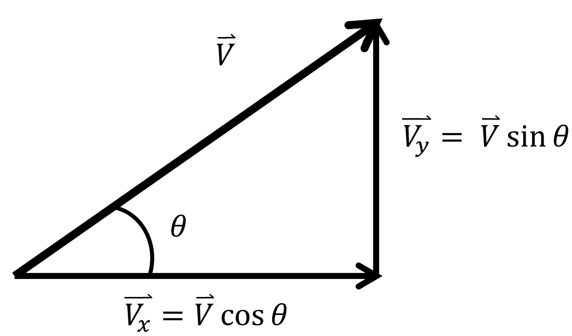


Figure 3 - Vector Components

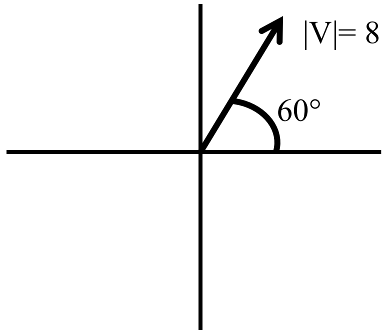


Figure 4 - Example of a Vector

For example, the above vector has a length of 8 at 60° North of East.

$$\vec{V} = 8 \angle 60° \ N \ of \ E$$

$$\vec{V} = (x, y) = \ (8 \cos 60°, 8 \sin 60°)$$

Vectors can be manipulated by scalar multiplication or vector addition. In scalar multiplication, a vector's magnitude is changed by a factor. If the scalar is negative, the vector will change to the opposite direction. Vector addition can be accomplished through a few ways. One way is by adding tail-to-tip: adding one vector's tail to the other vector's tip. The new vector goes from one vector's tail to the second vector's tip. A second way is by adding the x- and y- components together to form a new vector. The second method is the approach used in modeling the flight path.

Free Body Diagrams

Forces are represented by vectors. All forces acting on an object act according to Newton's Second Law which states that the acceleration of an object as produced by a net force is directly proportional to the magnitude of the net force, in the same direction as the net force, and inversely proportional to the mass of the object ("Newton's Second Law" 1).

$$\Sigma\vec{F} = m\vec{a} \tag{1}$$

The net force is found by summing all the forces acting upon that object. A common approach is to draw the object isolated from its surroundings. Then, all the forces acting upon that object are drawn with direction and magnitude. This process is known as creating a free-body diagram (FBD).

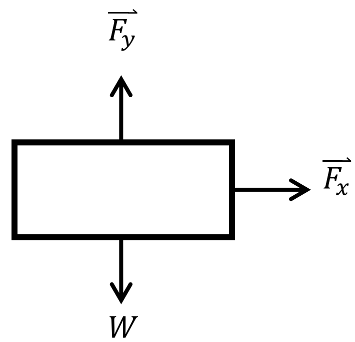


Figure 5 – Free-Body Diagram (FBD)

Work and Energy

In addition to vectors and forces, it is pertinent to understand work and energy. Work done on an object by a constant applied force is the product of the force and displacement (Wolfson 86).

$$W_x = F_x \Delta x \tag{2}$$

It is important to note that work is only done when there is a change in distance due to an applied force in the same direction. Negative work occurs when a force is applied in the opposite direction of motion. If an object is stationary, despite an applied force, there is no work done on the object. For example, if a person applies a force to a box but the box does not move, then no work is done. On the other hand, if a person is carrying a box from one side of a room to another without vertical displacement, no work is done because the displacement is not in the same direction as the applied force to hold the box.

When work is done on an object, that object gains mechanical energy. Mechanical energy is an object's energy due to motion or position. It is broken down into kinetic and potential energy. Kinetic energy is the energy associated with motion. The kinetic energy, KE, of an object of mass *m* moving at speed *v* is

$$KE = \frac{1}{2}mv^2. \tag{3}$$

Thus, all moving objects possess kinetic energy. The change in an object's kinetic energy is equal to the net work done on the object. This concept is known as the Work-Energy Theorem (Wolfson 92-93).

$$\Delta KE = W_{net} \tag{4}$$

For baseball flight, the W_{net} is associated with the aerodynamic lift and drag.

Potential energy (*PE*) is the energy associated with vertical position. The change ΔPE_{AB} in "potential energy associated with a conservative force is the negative of the work done by that force as it acts over any path from point A to point B" (Wolfson 103):

$$\Delta PE_{AB} = -\int_A^B \vec{F} \cdot d\vec{r}. \tag{5}$$

More often, potential energy is thought of as gravitational potential energy. The gravitational potential energy of an object is determined by the change in vertical displacement (Δy), mass (*m*) and gravity (*g*) (Wolfson 104).

$$\Delta PE = mg\Delta y \tag{6}$$

In summary, mechanical energy, the sum of kinetic and potential energy, and the work energy, is conserved.

$$\Delta KE + \Delta PE - W_{in,net} = 0 \tag{7}$$

The conservation of mechanical energy is a key concept in modeling the path of any trajectory.

Fluid Mechanics

For any airborne object, the fluid forces of lift and drag must be considered since these are critical to the transfer of work energy to the object. Before delving into lift and drag forces, it is important to understand the basics of fluid mechanics. The nature of fluid flow field is described as laminar, transitional, or turbulent.

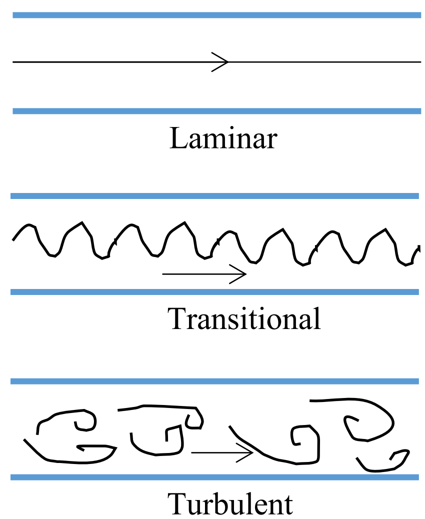


Figure 6 - Types of Fluid Flow

Laminar flow occurs at relatively low velocities where a fluid particle moves in straight lines. Transitional flow occurs at higher velocities. A fluid particle in transitional flow begins to break from the straight path to a waved shape. The transitional period occurs quickly between laminar and turbulent flow. Turbulent flow occurs at high velocities where the fluid particle motion is unpredictable. A real world example of laminar and turbulent flow can be seen while doing dishes. At lower flow rates, the water comes out of the faucet as laminar flow. When the water hits a spoon, the water's motion becomes random, thus turbulent.

As an object moves through a fluid, the fluid is displaced and is forced to flow around the object. At low velocities, the fluid moves around the object in a well-behaved manner, continues in streamlines, and settle undisturbed behind the object, as seen below in Figure 6. As the velocity increases, the flow begins to separate behind the object, and a wake region of disturbed fluid is created. The specific characteristics of this wake region are dependent on the nature of the flow (either laminar or turbulent) and are an important factor in determining the aerodynamic drag.

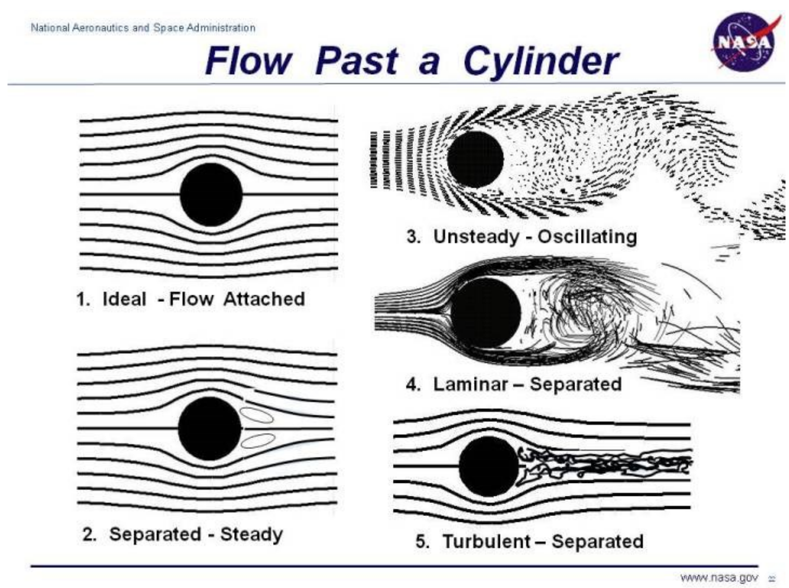


Figure 7 - Flow Past a Cylinder from NASA

Lift and Drag

Lift is the aerodynamic force that acts to hold an object in the air, and by definition is perpendicular to the direction of motion. Lift is dependent on the density of the fluid (ρ), the geometry of the object (A), the relative velocity (V), and the lift coefficient (C_L).

$$F_L = \frac{1}{2}\rho A v^2 C_L \quad (8)$$

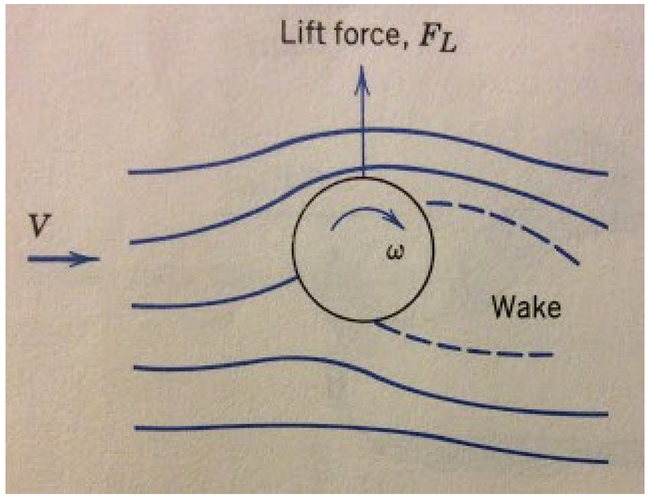


Figure 8 - Flow Pattern (Fox 464)

Lift is created by different pressures on opposite sides of an object, and for a baseball this is created by the spin. As seen in Figure 7, if a baseball is rotating clockwise (backspin) as it travels right to left through the air, then the upper surface of the ball moves in the same direction of the ball. Conversely, the lower surface travels in the opposing direction. This creates an off-center wake region causing the air past the ball to be deflected downward. The downward force is offset by an equal and opposite upward force known as lift force (Watts 57). More formally, the Kutta-Joukowski Lift theorem can be used to calculate the lift of a spinning ball **(**"Ideal Lift of a Spinning Ball" 1). In this equation, r is the ball radius and ω is the speed of rotation.

$$F_L = \frac{4}{3}(4\pi^2 r^3 \omega \rho v) \quad (9)$$

Drag is the other key aerodynamic force. Aerodynamic drag is defined as the fluid drag force that acts on any moving solid body through a fluid flow field. Thus, drag acts in the opposite direction of motion. Similar to lift, drag is dependent on the density of the fluid, the cross sectional area of the object, the relative velocity, and the drag coefficient (C_D).

$$F_D = \frac{1}{2}\rho A v^2 C_D \tag{10}$$

As the drag coefficient increases, the drag force increases. Thus, the greater the drag coefficient is, the more the fluid acts to slow the object down. The aerodynamic drag coefficient can be determined by experimentally measuring the drag force on a ball at varying air speeds in a wind tunnel.

MODELING FLIGHT PATH

Forces Acting on a Baseball

To correctly model the flight path of a baseball, it is pertinent to identify all forces acting on the ball. As a spinning baseball moves through the air, it experiences lift, drag, and gravity forces. The direction of the flight path for a non-ground ball is initially above the horizontal (+Θ), and then incrementally decreases to below the horizontal (-Θ). Because of this, two free-body diagrams are required: one for the upward motion and one for the downward motion.

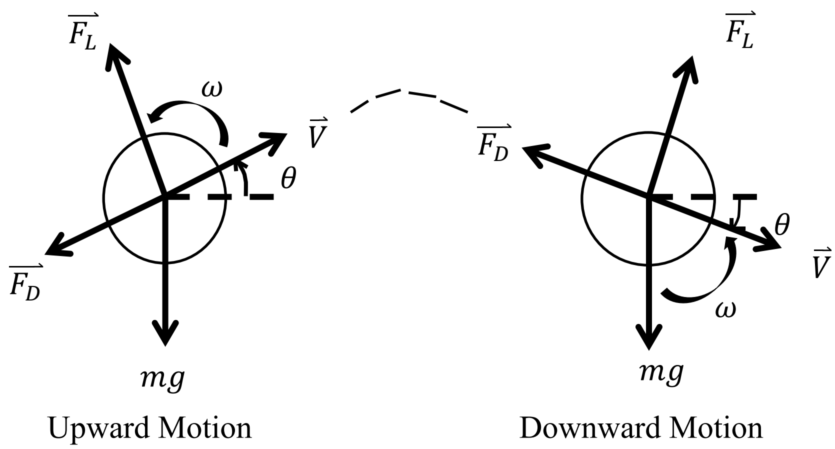


Figure 9 - FBDs for Baseball

As previously described, the drag force acts in the opposite direction of motion and the lift force acts perpendicular to the direction of motion. The force of gravity (*mg*) acts vertically downward at all times. These fundamental forces can be broken into Cartesian coordinates to further help explain how the forces act on the object.

Upward Motion:

$$\Sigma F_x = -F_D \cos\theta - F_L \sin\theta = 0 \qquad (11)$$

$$\Sigma F_y = -F_D \sin\theta + F_L \cos\theta - mg = 0 \qquad (12)$$

Downward Motion:

$$\Sigma F_x = -F_D \cos\theta + F_L \sin\theta = 0 \qquad (13)$$

$$\Sigma F_y = F_D \sin\theta + F_L \cos\theta - mg = 0 \qquad (14)$$

Lift and drag work together to oppose the forward motion as the ball height above the ground level increases. But after the height reaches the maximum and the motion is downward, lift actually assists the forward motion while the drag force continues to oppose the forward motion. On the other hand, the lift force assists the upward motion as the ball's height increases, while the drag opposes the upward motion. But then on the way down, the lift and drag both oppose the downward motion. This results in a non-parabolic, non-symmetric flight path for the spinning baseball.

Drag force is calculated using the drag coefficient at the corresponding velocity. Drag force is also dependent on the ball configuration and seam height, which will be detailed later. The lift force is calculated using an equation derived from an experiment performed by Robert Watts and R. Ferrer. The experiment used three spinning baseballs in various configurations in a subsonic wind tunnel. Their "measurements showed that the force on the baseball is proportional to the product of rotational speed (ω) and velocity (*V*), not rotational speed and velocity squared (ωv^2)" (Watts 75). Specifically, they found that the equation for lift force for a spinning baseball is as follows:

$$F_L = (6.4 \times 10^7)\omega v,\ where\ \omega\ [rpm\ ], v\ \left[\frac{ft}{s}\right], F_L\ \left[lb_f\right]. \tag{15}$$

The aforementioned equations for lift and drag will be used in creating the model.

Flight Path Model

The flight path of a baseball is modeled using the conservation of mechanical energy in the x- and y- directions. As previously discussed, the change in kinetic energy of the baseball is determined by the change in velocity and the change in potential energy is determined by the change of height of the baseball. Work of the baseball is determined by the products of the drag and lift forces and the change in distance. The conservation of mechanical energy requires the following:

$$\Delta KE + \Delta PE - W_{net} = 0. \tag{16}$$

The conservation of energy equations can be split into x- and y-components. Lift and drag act in both directions, thus the resultant work will be included in both equations. Velocity also acts in both directions so the kinetic energy will be included in both equations. The force of gravity only acts in the y-direction.

$x - Direction$:

$$\left(F_{drag,x} + F_{Lift,x}\right)x + \frac{1}{2}mv_x^2 = constant \qquad (17)$$

$y - Direction$:

$$\left(F_{drag,y} + F_{Lift,y}\right)y + mgy + \frac{1}{2}mv_y^2 = constant \qquad (18)$$

The conservation of energy assumes an initial position and a final position. The initial position is determined by the initial conditions of a batted ball, including an initial x- and y-coordinate, velocity, angle from the horizontal, and spin. The next position is determined by a change in the horizontal position. The idea is to move through each horizontal increment while adjusting the velocity until the conservation of energy holds for both x- and y-directions. Essentially, the velocity will be decreased at each increment until the difference between the energy in and energy out approaches zero.

$$Difference_x = KE_{out,x} + W_{out,x} - KE_{in,x} \qquad (19)$$

$$= \frac{1}{2}mv_{out,x}^2 + \left(F_{L,x} + F_{D,x}\right)(x_o - x_i) - \frac{1}{2}mv_{out,x}^2 \qquad (20)$$

$$= \frac{1}{2}mv_{out,x}^2 + \left(\left((6.4 \times 10^{-7})\omega v_{out,x}\right) + \left(\frac{1}{2}\rho AC_{D,x}V_{avg,x}^2\right)\right)(x_o - x_i) - \frac{1}{2}mv_{out,x}^2 \qquad (21)$$

$$Difference_y = KE_{out,y} + W_{out,y} - KE_{in,y} + PE_{out} - PE_{in} \qquad (22)$$

$$= \frac{1}{2}mv_{out,y}^2 + \left[F_{D,y}|y_o - y_i| - F_{L,y}(y_o - y_i)\right] - \frac{1}{2}mv_{out,y}^2 + mg(y_o - y_i) \qquad (23)$$

$$= \frac{1}{2}mv_{out,y}^2 + \left[\left(\frac{1}{2}\rho AC_{D,y}V_{avg,y}^2\right)|y_o - y_i| - \left((6.4 \times 10^{-7})\omega v_{out,y}\right)(y_o - y_i)\right] - \frac{1}{2}mv_{out,y}^2 \qquad (24)$$

At each horizontal increment, the horizontal velocity is decreased until the difference in energy in and energy out in the x-direction is less than 0.001 $lb_f \cdot ft$. The same approach is then applied in the y-direction. The resultant x and y velocities of each increment is used in the next increment and so on, until the ball has reached a vertical height of zero (ground level).

Model Verification

Before applying these principles to the flight path of a baseball, a check can be performed of the wind tunnel results and a simplified version of the model. The simplified version of the model predicts the time for a ball to drop from the third story of Tucker Technology Center on the TCU campus to the basement, a height of 50 feet 3 inches. With this model, only the work due to drag force, energy due to position, and energy due to motion are considered in the y-direction.

$$\Delta KE + \Delta PE - W_{in,net} = 0 \tag{25}$$

$$\frac{1}{2}m(V_2^2 - V_1^2) + mg(y_2 - y_1) - \left(\frac{1}{2}\rho s C_d\left(\frac{V_2+V_1}{2}\right)\right)(y_2 - y_1) = 0 \tag{26}$$

Again, the goal is to minimize the difference between each side of the equation by changing the velocity. The time for the ball to drop can be calculated at each increment and summed to find the total time to drop.

$$t = \frac{y_2 - y_1}{V_{avg}} = \frac{2(y_2 - y_1)}{V_2 + V_1} \tag{27}$$

The dropped ball test was performed with an orange dimpled ball from the batting cages at a local Putt-Putt activities center. Using the analysis described in Appendix A, the orange dimpled ball displayed a nearly constant drag coefficient of 0.25 (Figure 10).

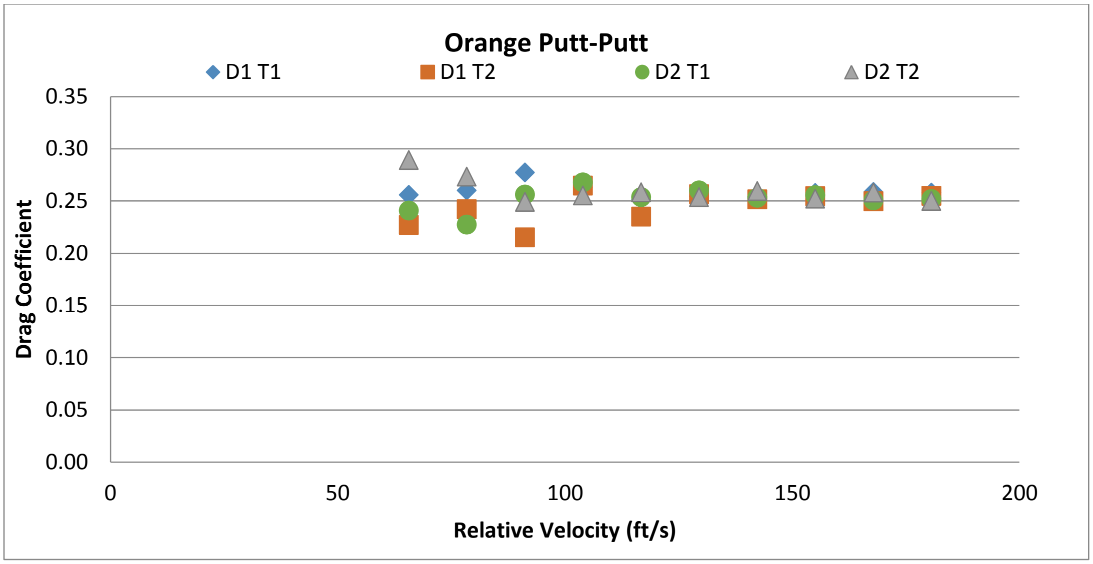


Figure 10 - Results of Wind Tunnel Analysis for Orange Putt-Putt Ball

The model predicted the ball would hit the ground 1.77 seconds after being released. In order to properly assess the experimental error and confirm the assumption, a total of twenty-five drop tests were performed among three people. Each person dropped the ball and used a stop watch to track the time from release to hitting the ground. As seen in Appendix B, the results ranged from 1.69 to 1.94 seconds with a standard deviation of 0.06 seconds. The times to hit the ground averaged at 1.80 seconds. Thus, there was a less than a 2% difference between the model prediction and the test average. The results were the first step in confirming the validity of the modeling approach, and the accuracy of the drag coefficient measured in the TCU Wind Tunnel.

Wind Tunnel Test Results of the Baseballs

In order to test the claims made by the NCAA, four unused baseballs provided by the TCU baseball team were tested in the wind tunnel. Two of the baseballs came from TCU's 2014 Division I season and the other two came from the 2015 season. A baseball from each season was mounted in the two seam configuration, and the other in the four seam configuration. Each ball was tested on two separate days in the TCU Department of Engineering Wind Tunnel. Analysis of the drag coefficients found the drag coefficient to be linearly decreasing at low velocities and nearly constant at high velocities.

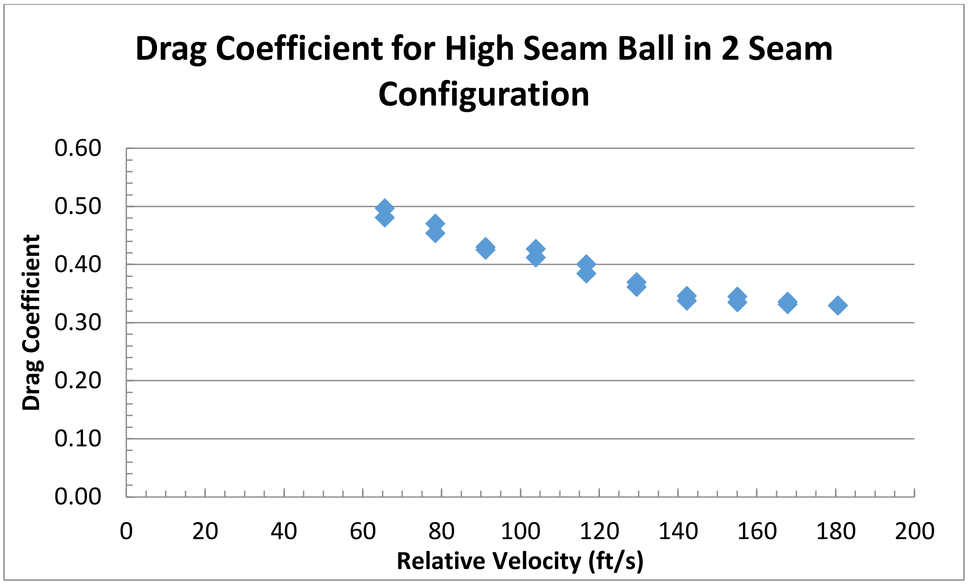


Figure 11 - Drag Coefficient for High Seam Ball – 2 Seam

As a general trend, the drag coefficient is linearly decreasing at low velocities and then constant at high velocities. For the higher seamed (old) baseball, the drag coefficient is at a maximum near 0.5 at 65 feet per second and decreases to 0.33 at 147 feet per second. At velocities above 147 feet per second, the drag coefficient is constant at 0.33.

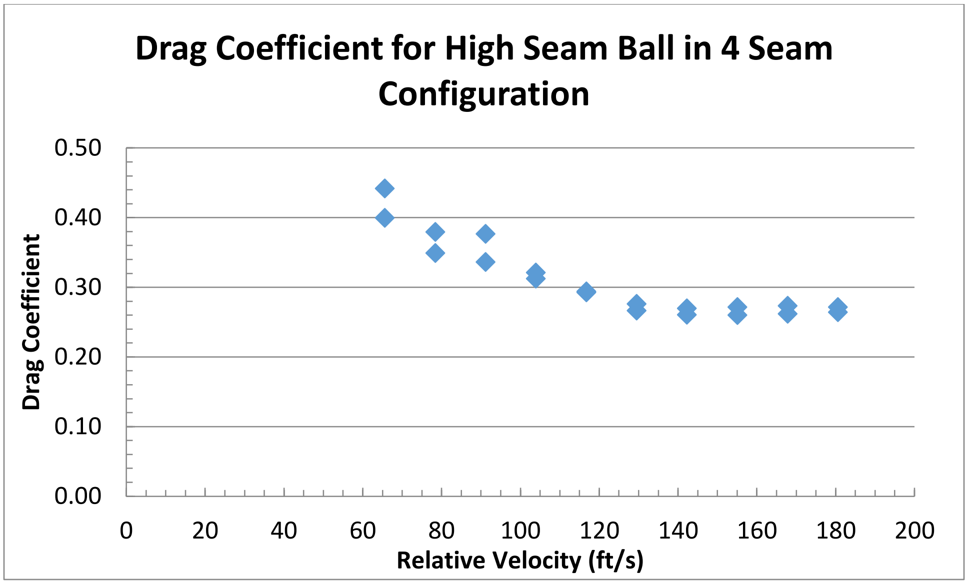


Figure 12 - Drag Coefficient for High Seam Ball - 4 Seam

When the high seam baseball is in the four seam configuration, the baseball experiences less drag. The drag coefficient is at a maximum around 0.45 and decreases to 0.27 around 130 feet per second. At this velocity, the drag coefficient becomes constant at 0.27.

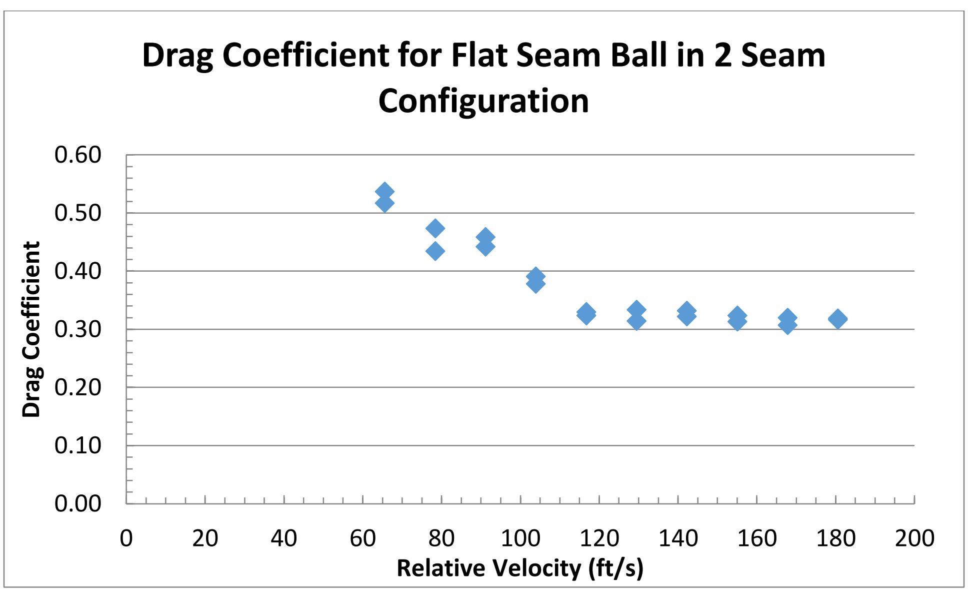


Figure 13 - Drag Coefficient for Flat Seam Ball - 2 Seam

The new, flat seam baseball has steeper slope at the low velocities than the old baseball in the two seam configuration. The drag coefficient peaks around 0.55 and decreases until 0.32 around 117 feet per second. It is important to note that the flat seam baseball's drag coefficient becomes constant at a lower velocity than the high seam baseball.

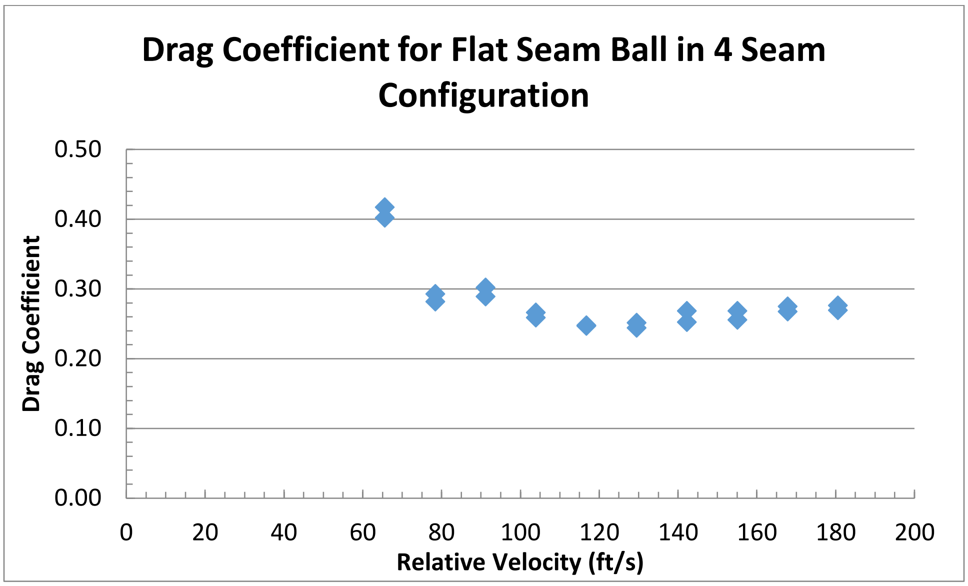


Figure 14 - Drag Coefficient for Flat Seam Ball - 4 Seam

The flat seam baseball in the four seam configuration has a key variation not presented in the other baseball configurations. The drag coefficient decreases linearly at a comparable rate until 117 feet per second and then actually increases slightly at high velocities. The relationships between drag and relative velocity for each baseball and configuration are detailed in Appendix A.

In a paper by Bearman and Harvey, the spin rate was shown to cause a slight increase in drag coefficient for a dimpled golf ball over the range of spin number applicable to this current baseball study. Based on the reported data, the measured static drag coefficient for each baseball will be increased by 5% in the flight model (for a 1400 rpm spin rate) to account for the effect if spin on drag.

The diameter of each ball was measured at three points on the ball: seam-to-seam, leather-to-leather, and leather-to-leather on the opposing side. The average values were used in subsequent modeling calculations.

Table 1 - Summary of Diameter

Baseball	Leather-Leather (in)	Leather-Leather (in)	Seam-Seam (in)	Average (in)
Old - 2 Seam	2.855	2.863	2.952	2.890
Old - 4 Seam	2.856	2.86	2.952	2.889
New - 2 Seam	2.845	2.86	2.93	2.878
New - 4 Seam	2.86	2.87	2.94	2.890

MODELING RESULTS

Comparison to NCAA Claims

Washington State University Sports Science Laboratory used a pitching machine located at the home plate to launch balls spinning at 1400 revolutions per minute (back spin) at a speed of 95 miles per hour and an angle of 25 degrees. For this condition, the reputed data showed that the old baseball (high seams) traveled an average of 367 feet, while the new baseball (flat seams) traveled 20 feet farther.

Each ball's flight path was then modeled using the methodology previously described. Each ball used the following initial conditions and constants: weight, initial vertical displacement, initial velocity of ball coming off the bat, initial angle from horizontal off the bat, speed of rotation, and air density. The diameter, thus cross-sectional area, and the drag coefficient were dependent on the ball and configuration. Matlab®, a software program, ran through a logical convergence algorithm for each ball to minimize the difference between energy into the system and energy out of the system. Concurrently, the program stores each data point of the flight path and produces a trajectory plot. The details of this code are in Appendix D.

Table 2 - Initial Conditions for NCAA Comparison

Weight (lb_f)	0.3125	**Speed of Rotation (rpm)**	1400	**Velocity (mph)**	95
Initial Height (ft)	3	**Air Density (lb_m/ft^3)**	0.075	**Angle (degrees)**	25

The model runs each baseball with unique diameter and drag coefficient. The resultant distance of the two seam and four seam configurations for each season's baseballs were averaged together to find an average distance for the old and new baseball. A summary of horizontal distance traveled is contained in Table 3 while flight paths are shown in Figure 15**.**

Table 3 - Summary of Horizontal Distance with Full Drag Effects

Configuration	Old, Higher-Seamed Ball (ft)	New, Flatter-Seamed Ball (ft)	Difference (ft)
2 Seam	348	356	8
4 Seam	386	414	28
Average	367	385	18

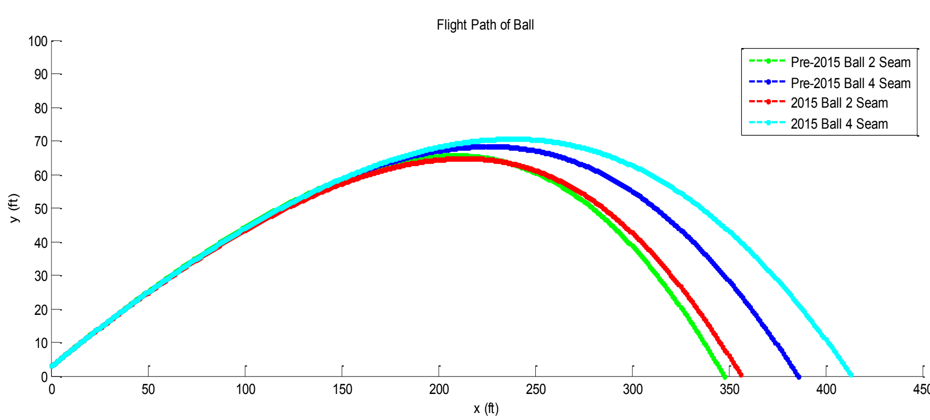


Figure 15 - Flight Path with Full Drag Effects

The results further confirm the modeling approach and the accuracy of the wind tunnel testing.

No Spin Case

To understand the importance of spin, the flight path was modeled without the spin effect. The lift force and drag force due to spin (the 5% enhancement) was neglected in the flight path model of the new (high seam) ball in the four seam configuration. The initial conditions were the same as the conditions for the NCAA distance claim.

Table 4 - Spin Effect for New Baseball in 4 Seam

Set Up	Distance Traveled (ft)
No Spin	342
With Spin	414
Distance	72

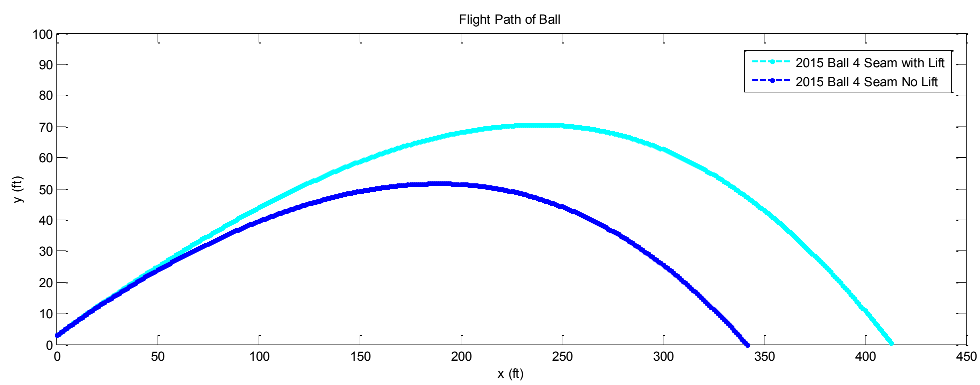


Figure 16 - No Spin Effect for New Baseball in 4 Seam

Line Drive Back Through the Box

The second claim maintained that the safety of the players would not be affected. A study found that college baseball pitchers were capable of catching balls approaching between 70 and 90 miles per hour. Additionally, the players are able to avoid contact with fast approaching (100 mph) balls with quick reactionary movement, at times beginning under 100 milliseconds after the ball was hit (Young 31). One way to test the safety claim is by simulating a high velocity line drive to the pitcher. A line drive is a type of hard hit that travels horizontal, or slightly above the horizontal. Again, the Matlab code was used to model the flight path. Instead of modeling the entire flight path, the time to travel the 60 feet to the pitcher's mound is recorded. The test results are summarized in Table 6.

Table 5 - Initial Conditions for Line Drive Hit

Weight (lb_f)	0.3125	**Speed of Rotation (rpm)**	1400	**Velocity (mph)**	115
Initial Height (ft)	3	**Air Density (lb_m/ft^3)**	0.075	**Angle (degrees)**	1

Table 6 - Summary of Time to Pitcher's Mound

Configuration	Old, Higher-Seamed Ball (s)	New, Flatter-Seamed Ball (s)	Difference (s)
2 Seam	0.3768	0.3760	0.0008
4 Seam	0.3729	0.3728	0.0001
Average	0.37485	0.3744	0.00045

DISCUSSION

One primary goal of the research was to test the claims made by the NCAA and Washington State University (WSU). Experimental data from the pitching machine created by WSU to simulate a batted homerun showed an average distance of 367 feet for the high seam old baseballs. The new flat seam baseballs traveled an average of 387 feet, 20 feet farther. In the first test, only the static drag was used.

The model predicted average distance traveled for the old ball agreed with the 367feet measured by WSU. The model predicted the new ball to travel an average of 18 feet farther at 385 feet, in comparison to the WSU measured at 387 feet. Comparisons of the test results and the model predictions are shown in Table 7.

Table 7- Comparison of Model Predictions and NCAA Data

Ball	Model Average (ft)	NCAA Data (ft)
Old	367	367
New	385	387

The minor difference in distance traveled for the batted ball may be attributed to potential minor errors in configuration, drag and lift forces and the weather conditions. The model assumes air density at 70°F and sea level. Additionally, it does not take any wind into account. These factors combined could provide the minimal difference between the claim and model.

The effect of spin alone was also considered. The spin creates the lift force as well as inducing small amounts of drag. The new (high seam) baseball in the four seam configuration was modeled with and without the effect of spin. When neglecting spin, the baseball's path appears to be symmetric and parabolic (Fig. 16). To understand the effect of lift, it is better to break the graphs into the upward and downward path. In the upward motion, both flight paths start the same. But, the spin case diverges from the path and continues upward at higher angles. Furthermore, the spin case reaches its maximum height farther from the home plate than the no spin case. This is the direct result of the spin creating a lift force that pushes up on the ball during upward motion. In the downward motion, the no spin case remains parabolic. The lift created in the spin case works with the drag to oppose the downward motion. This allows the spin case to travel farther in the downward motion. Overall, the importance of spin is apparent and must be included to model baseball flight path.

Finally, the claim regarding player safety was tested. Since player safety was said to be maintained, it is feasible to directly compare the results of each ball. The time for each ball to travel 60 feet ranged from 0.373 to 0.377 seconds. Furthermore, the two-seam configuration traveled faster than the four-seam configuration by approximately 3 milliseconds. The difference in the average travel time of the higher-seamed old baseball and the flatter-seamed new baseball is less than half a millisecond, as seen in Table 8.

Table 8 - Analysis of Time to Return to Mound

Ball	Average (s)
Old	0.374859
New	0.374397
Difference	0.000462

CONCLUSION

The model based on engineering and physics principles was created to accurately track the flight path of NCAA Division I Big 12 baseballs from the 2014 and 2015 seasons. In addition to tracking flight path, the model also recorded time to travel to the pitcher's mound. These key parameters allowed the claim regarding increased travel distance while maintaining player safety to be vetted.

The modeling results for the batted baseball in flight were confirmed by test data. The average model predicted distance for the old, high seam baseball hit at 95 miles per hour, 1400 revolutions per minute and an angle of 25 degrees was 367 feet, which compared exactly with the measured distance data. The average model predicted distance for the new, flat seam baseball traveled was 18 feet farther, only 2 feet short of the measured data. The difference in two feet can possibly be attributed to the testing conditions, such as initial height, configuration, weather, lift, and drag, for each experiment. But, in general, the claim holds true: the new baseball will travel approximately 20 feet farther for a homerun-like hit.

The model was also used to simulate a powerful line drive hit at 115 miles per hour at an angle of 1 degree. The difference in time to return to the pitcher's mound between the new and old baseballs was less than half a millisecond. Since a college pitcher has quick reactionary movements near 100 milliseconds, it is fair to assume that half a millisecond will not impact the pitcher's ability to catch or dodge the ball (Young 31-32). In other terms, for a pitcher to bring a glove up to his face to protect from a line drive, there is less than 1/8 inch difference in position. Thus, the model proved the NCAA claims to hold true: the new ball travels further while maintaining pitcher's safety.

APPENDIX A

Relevant Equations for Analysis of Wind Tunnel Results

$$\rho_{air}\left[\frac{lb_m}{ft^3}\right] = \frac{70.8\left[\frac{lb_f}{ft^2 \cdot in\ Hg}\right] P_{atm}[in\ Hg]}{53.34\left[\frac{ft \cdot lb_f}{lb_m \cdot {}^\circ R}\right](T_{air} + 460.67)[{}^\circ R]}$$

$$Fan\ Speed, v = 5.1104f - 10.9807, \qquad where\ f\ is\ frequency\ in\ Hz$$

$$Post\ Correction\ F_{drag}\ [lb_f] = |F_{drag}| - (5.0856 \times 10^{-5})f^2$$

$$Drag\ Coefficient, C_D = \frac{Post\ Correction\ F_{drag}\ [lb_f]}{\frac{1}{2}\rho_{air}\left[\frac{lb_m}{ft^3}\right] v^2 \left[\frac{ft^2}{s^2}\right]\left(\frac{\pi}{4}\ d^2\ [in^2]\right)} \times 32.2\left[\frac{lb_m \cdot ft}{lb_f \cdot s^2}\right] \times 144\ \left[\frac{in^2}{ft^2}\right]$$

$$Lift\ Coefficient, C_L = \frac{F_{lift}\ [lb_f]}{\frac{1}{2}\rho_{air}\left[\frac{lb_m}{ft^3}\right] v^2 \left[\frac{ft^2}{s^2}\right]\left(\frac{\pi}{4}\ d^2\ [in^2]\right)} \times 32.2\left[\frac{lb_m \cdot ft}{lb_f \cdot s^2}\right] \times 144\ \left[\frac{in^2}{ft^2}\right]$$

Testing Date: 01/13/2015

Testing Conditions:

P_{atm} = 29.32 in Hg

T_{air} = 70 °F

μ =3.80E-07 (lbf·s)/ft²

ρ_{air} = 0.07333 lb_m/ft^3

Table 9 - Putt Putt Ball, Test 1 – 1/13/15

Frequency (Hz)	Fan Speed (ft/s)	F-drag (lbf)	F-drag (Post Correction) (lbf)	Drag Coefficient	F-lift (lbf)	Lift Coefficient
15	65.6753	-0.06556	0.0541	0.2557	-0.0117	-0.0553
17.5	78.4513	-0.0941	0.0785	0.2600	-0.01789	-0.0592
20	91.2273	-0.13355	0.1132	0.2772	-0.02058	-0.0504
22.5	104.0033	-0.16533	0.1396	0.2630	-0.03975	-0.0749
25	116.7793	-0.2	0.1682	0.2514	-0.06317	-0.0944
27.5	129.5553	-0.24816	0.2097	0.2546	-0.06759	-0.0821
30	142.3313	-0.29809	0.2523	0.2538	-0.07882	-0.0793
32.5	155.1073	-0.3577	0.3040	0.2575	-0.0976	-0.0827
35	167.8833	-0.41974	0.3574	0.2585	-0.10499	-0.0759
37.5	180.6593	-0.4843	0.4128	0.2578	-0.12194	-0.0761

Table 10- Putt Putt Ball, Test 2 – 1/13/15

Frequency (Hz)	Fan Speed (ft/s)	F-drag (lbf)	F-drag (Post Correction) (lbf)	Drag Coefficient	F-lift (lbf)	Lift Coefficient
15	65.6753	-0.05947	0.0480	0.2269	-0.01233	-0.0583
17.5	78.4513	-0.08861	0.0730	0.2419	-0.01845	-0.0611
20	91.2273	-0.10814	0.0878	0.2150	-0.03164	-0.0775
22.5	104.0033	-0.16612	0.1404	0.2645	-0.03316	-0.0625
25	116.7793	-0.18903	0.1572	0.2350	-0.03607	-0.0539
27.5	129.5553	-0.24962	0.2112	0.2564	-0.06591	-0.0800
30	142.3313	-0.29583	0.2501	0.2516	-0.07998	-0.0805
32.5	155.1073	-0.35427	0.3006	0.2546	-0.08242	-0.0698
35	167.8833	-0.40749	0.3452	0.2496	-0.10777	-0.0779
37.5	180.6593	-0.4798	0.4083	0.2550	-0.11631	-0.0726

Testing Date: 01/16/2015

Testing Conditions:

P_{atm} = 29.32 in Hg

T_{air} = 70 °F

μ =3.80E-07 (lbf·s)/ft²

ρ_{air} = 0.07333 lb_m/ft^3

Table 11 - Old Baseball in 4 Seam Configuration - 1/16/15

Frequency (Hz)	Fan Speed (ft/s)	F-drag (lbf)	F-drag (Post Correction) (lbf)	Drag Coefficient	F-lift (lbf)	Lift Coefficient
15	65.6753	-0.09877	0.0873	0.3996	-0.01453	-0.0665
17.5	78.4513	-0.12431	0.1087	0.3487	-0.02668	-0.0856
20	91.2273	-0.16208	0.1417	0.3362	-0.03628	-0.0860
22.5	104.0033	-0.19683	0.1711	0.3122	-0.05361	-0.0978
25	116.7793	-0.23467	0.2029	0.2937	-0.04068	-0.0589
27.5	129.5553	-0.26476	0.2263	0.2661	-0.04577	-0.0538
30	142.3313	-0.3128	0.2670	0.2602	-0.01946	-0.0190
32.5	155.1073	-0.37033	0.3166	0.2598	-0.02691	-0.0221
35	167.8833	-0.43613	0.3738	0.2618	-0.00644	-0.0045
37.5	180.6593	-0.50813	0.4366	0.2641	-0.04237	-0.0256

Table 12 - Old Baseball in 2 Seam Configuration - 1/16/15

Frequency (Hz)	Fan Speed (ft/s)	F-drag (lbf)	F-drag (Post Correction) (lbf)	Drag Coefficient	F-lift (lbf)	Lift Coefficient
15	65.6753	-0.11638	0.1049	0.4806	0.01258	0.0576
17.5	78.4513	-0.15682	0.1412	0.4533	0.0291	0.0934
20	91.2273	-0.19938	0.1790	0.4249	0.0262	0.0622
22.5	104.0033	-0.25121	0.2255	0.4117	0.00966	0.0176
25	116.7793	-0.29702	0.2652	0.3842	0.06684	0.0968
27.5	129.5553	-0.34521	0.3068	0.3610	0.08686	0.1022
30	142.3313	-0.39145	0.3457	0.3371	0.01095	0.0107
32.5	155.1073	-0.47309	0.4194	0.3443	-0.01378	-0.0113
35	167.8833	-0.54094	0.4786	0.3354	0.04544	0.0318
37.5	180.6593	-0.61495	0.5434	0.3289	0.07481	0.0453

Table 13 - New Baseball in 4 Seam Configuration - 1/16/15

Frequency (Hz)	Fan Speed (ft/s)	F-drag (lbf)	F-drag (Post Correction) (lbf)	Drag Coefficient	F-lift (lbf)	Lift Coefficient
15	65.6753	-0.09952	0.0881	0.4019	0.07172	0.3273
17.5	78.4513	-0.10702	0.0914	0.2925	0.03585	0.1147
20	91.2273	-0.14783	0.1275	0.3015	0.04639	0.1097
22.5	104.0033	-0.17177	0.1460	0.2657	0.01707	0.0311
25	116.7793	-0.20317	0.1714	0.2474	-0.01478	-0.0213
27.5	129.5553	-0.25277	0.2143	0.2513	-0.0113	-0.0133
30	142.3313	-0.30537	0.2596	0.2522	-0.00927	-0.0090
32.5	155.1073	-0.36585	0.3121	0.2554	0.02768	0.0226
35	167.8833	-0.45519	0.3929	0.2744	0.11403	0.0796
37.5	180.6593	-0.52895	0.4574	0.2759	0.14969	0.0903

Table 14 - New Baseball in 2 Seam Configuration - 1/16/15

Frequency	**Fan Speed**	**F-drag**	**F-drag** (Post Correction)	**Drag Coefficient**	**F-lift**	**Lift Coefficient**
(Hz)	(ft/s)	(lbf)	(lbf)		(lbf)	
15	65.6753	-0.12348	0.1120	0.5167	-0.01027	-0.0474
17.5	78.4513	-0.14996	0.1344	0.4343	0.11822	0.3821
20	91.2273	-0.20532	0.1850	0.4421	0.17155	0.4100
22.5	104.0033	-0.23803	0.2123	0.3904	0.14242	0.2619
25	116.7793	-0.25367	0.2219	0.3236	0.03321	0.0484
27.5	129.5553	-0.30323	0.2648	0.3138	0.03766	0.0446
30	142.3313	-0.37358	0.3278	0.3219	0.04817	0.0473
32.5	155.1073	-0.44457	0.3909	0.3232	0.11179	0.0924
35	167.8833	-0.51468	0.4524	0.3193	0.21623	0.1526
37.5	180.6593	-0.59429	0.5228	0.3186	0.29004	0.1768

Testing Date: 01/26/2015

Testing Conditions:

P_{atm} = 29.23 in Hg

T_{air} = 70 °F

μ =3.80E-07 (lbf·s)/ft²

ρ_{air} = 0.07311 lb_m/ft^3

Table 15 - Putt Putt Ball, Test 1 – 1/26/15

Frequency (Hz)	Fan Speed (ft/s)	F-drag (lbf)	F-drag (Post Correction) (lbf)	Drag Coefficient	F-lift (lbf)	Lift Coefficient
15	65.6753	-0.06238	0.0509	0.2407	-0.16781	-0.7929
17.5	78.4513	-0.08423	0.0687	0.2273	-0.03132	-0.1037
20	91.2273	-0.12493	0.1046	0.2561	-0.35521	-0.8699
22.5	104.0033	-0.16781	0.1421	0.2677	-0.03723	-0.0701
25	116.7793	-0.20155	0.1698	0.2537	-0.05286	-0.0790
27.5	129.5553	-0.25257	0.2141	0.2600	-0.05315	-0.0645
30	142.3313	-0.29732	0.2515	0.2531	-0.07008	-0.0705
32.5	155.1073	-0.35521	0.3015	0.2554	-0.08676	-0.0735
35	167.8833	-0.40885	0.3466	0.2506	-0.09554	-0.0691
37.5	180.6593	-0.47476	0.4032	0.2518	-0.11115	-0.0694

Table 16 - Putt Putt Ball, Test 2 – 1/26/15

Frequency (Hz)	Fan Speed (ft/s)	F-drag (lbf)	F-drag (Post Correction) (lbf)	Drag Coefficient	F-lift (lbf)	Lift Coefficient
15	65.6753	-0.07264	0.0612	0.2892	-0.01712	-0.0809
17.5	78.4513	-0.09805	0.0825	0.2731	-0.01638	-0.0542
20	91.2273	-0.12201	0.1017	0.2490	-0.03301	-0.0808
22.5	104.0033	-0.16121	0.1355	0.2552	-0.04086	-0.0770
25	116.7793	-0.20453	0.1727	0.2582	-0.04672	-0.0698
27.5	129.5553	-0.24743	0.2090	0.2537	-0.05993	-0.0728
30	142.3313	-0.30372	0.2579	0.2595	-0.0646	-0.0650
32.5	155.1073	-0.35119	0.2975	0.2520	-0.08483	-0.0719
35	167.8833	-0.41877	0.3565	0.2578	-0.09074	-0.0656
37.5	180.6593	-0.47161	0.4001	0.2498	-0.1148	-0.0717

Testing Date: 01/30/2015

Testing Conditions:

P_{atm} = 29.53 in Hg

T_{air} = 70 °F

μ =3.80E-07 (lbf·s)/ft²

ρ_{air} = 0.07386 lb_m/ft^3

Table 17 - Old Baseball in 4 Seam Configuration - 1/30/15

Frequency (Hz)	Fan Speed (ft/s)	F-drag (lbf)	F-drag (Post Correction) (lbf)	Drag Coefficient	F-lift (lbf)	Lift Coefficient
15	65.6753	-0.10863	0.0972	0.4416	-0.02764	-0.1256
17.5	78.4513	-0.1347	0.1191	0.3793	-0.04303	-0.1370
20	91.2273	-0.18017	0.1598	0.3764	-0.05684	-0.1339
22.5	104.0033	-0.20291	0.1772	0.3210	-0.05943	-0.1077
25	116.7793	-0.23517	0.2034	0.2923	-0.04799	-0.0690
27.5	129.5553	-0.27477	0.2363	0.2759	-0.04851	-0.0566
30	142.3313	-0.32452	0.2787	0.2697	-0.04225	-0.0409
32.5	155.1073	-0.38686	0.3331	0.2714	-0.1347	-0.1097
35	167.8833	-0.45511	0.3928	0.2731	-0.101	-0.0702
37.5	180.6593	-0.52343	0.4519	0.2714	-0.11593	-0.0696

Table 18 - Old Baseball in 2 Seam Configuration - 1/30/15

Frequency (Hz)	Fan Speed (ft/s)	F-drag (lbf)	F-drag (Post Correction) (lbf)	Drag Coefficient	F-lift (lbf)	Lift Coefficient
15	65.6753	-0.12066	0.1092	0.4966	-0.05085	-0.2312
17.5	78.4513	-0.16313	0.1476	0.4702	-0.06563	-0.2091
20	91.2273	-0.20279	0.1824	0.4299	-0.0216	-0.0509
22.5	104.0033	-0.26097	0.2352	0.4265	-0.02034	-0.0369
25	116.7793	-0.31	0.2782	0.4001	-0.02419	-0.0348
27.5	129.5553	-0.35449	0.3160	0.3693	0.00057	0.0007
30	142.3313	-0.40255	0.3568	0.3454	-0.01863	-0.0180
32.5	155.1073	-0.46381	0.4101	0.3343	-0.0511	-0.0417
35	167.8833	-0.53832	0.4760	0.3312	-0.03492	-0.0243
37.5	180.6593	-0.61895	0.5474	0.3290	-0.07303	-0.0439

Table 19 - New Baseball in 4 Seam Configuration - 1/30/15

Frequency (Hz)	Fan Speed (ft/s)	F-drag (lbf)	F-drag (Post Correction) (lbf)	Drag Coefficient	F-lift (lbf)	Lift Coefficient
15	65.6753	-0.10342	0.0920	0.4168	-0.06499	-0.2945
17.5	78.4513	-0.10427	0.0887	0.2816	-0.0462	-0.1467
20	91.2273	-0.14335	0.1230	0.2889	-0.03352	-0.0787
22.5	104.0033	-0.16888	0.1431	0.2586	-0.0321	-0.0580
25	116.7793	-0.20378	0.1720	0.2465	0.02099	0.0301
27.5	129.5553	-0.24765	0.2092	0.2436	0.04511	0.0525
30	142.3313	-0.32391	0.2781	0.2683	-0.0555	-0.0535
32.5	155.1073	-0.38373	0.3300	0.2681	-0.1227	-0.0997
35	167.8833	-0.44786	0.3856	0.2674	-0.14098	-0.0978
37.5	180.6593	-0.52066	0.4491	0.2690	-0.15488	-0.0927

Table 20 - New Baseball in 2 Seam Configuration - 1/30/15

Frequency	Fan Speed	F-drag	F-drag (Post Correction)	Drag Coefficient	F-lift	Lift Coefficient
(Hz)	(ft/s)	(lbf)	(lbf)		(lbf)	
15	65.6753	-0.12858	0.1171	0.5364	-0.03506	-0.1605
17.5	78.4513	-0.16308	0.1475	0.4733	0.1283	0.4117
20	91.2273	-0.21338	0.1930	0.4581	-0.02297	-0.0545
22.5	104.0033	-0.23263	0.2069	0.3778	-0.03175	-0.0580
25	116.7793	-0.25945	0.2277	0.3297	-0.03814	-0.0552
27.5	129.5553	-0.32178	0.2833	0.3334	-0.0183	-0.0215
30	142.3313	-0.38598	0.3402	0.3317	-0.00793	-0.0077
32.5	155.1073	-0.43484	0.3811	0.3129	-0.0082	-0.0067
35	167.8833	-0.50019	0.4379	0.3069	-0.05448	-0.0382
37.5	180.6593	-0.59433	0.5228	0.3164	-0.15403	-0.0932

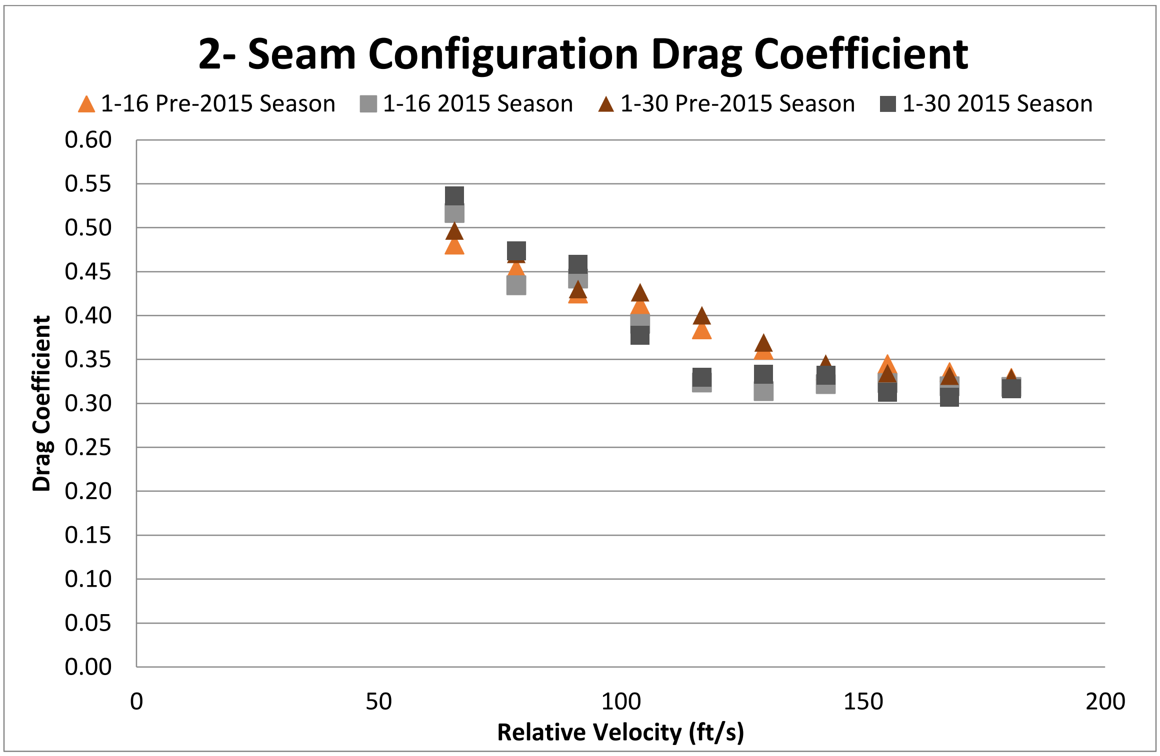


Figure 17 - 2 Seam Configuration for All Data Points

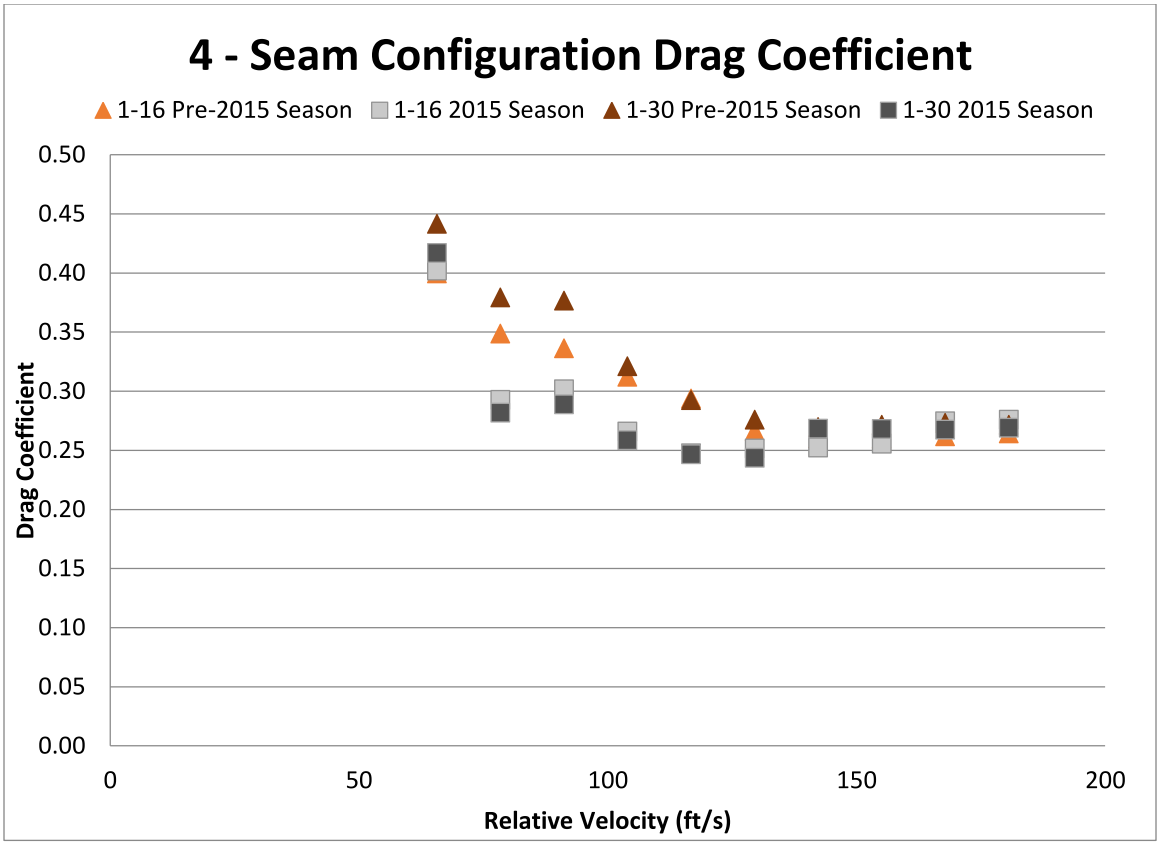


Figure 18 - 4 Seam Configuration for All Data Points

Baseball and Configuration	Lower Range (ft/s)	Relationship	Upper Range (ft/s)	Relationship
Old – 2 Seam	$V < 147$	$C_D = 0.615 - 0.00193v$	$V > 147$	$C_D = 0.33$
Old – 4 Seam	$V < 129.2$	$C_D = 0.53 - 0.00201v$	$V > 129.2$	$C_D = 0.27$
New – 2 Seam	$V < 117$	$C_D = 0.780 – 0.00575v$	$V > 117$	$C_D = 0.32$
New – 4 Seam	$V < 117.3$	$C_D = 0.435 – 0.001619v$	$V > 117.3$	$C_D = 0.185 + 0.000511v$

APPENDIX B

Dropped Ball Test

Matlab Code for Dropped Ball

```
% Time for ball to drop 50.25" (height from TTC 3rd floor to basement)

format long
%Constants
d = 2.815/12; % ft
m = 4.347/16; %lbs
g = 32.2; % ft^2/s
p = 0.073862; %lbm/ft^3

z = transpose([0:.25:50.25]);
v = zeros(length(z),1);
vave = zeros(length(z),1);
time = zeros(length(z),1);
PE = zeros(length(z),1);
KE = zeros(length(z),1);
Fd = zeros(length(z),1);
Wd = zeros(length(z),1);
diff = zeros(length(z),1);
n = length(v);
for i = 1:n

    if i == 1
        vave(i) = v(i);
        time(i) = 0;
        PE(i) = 0;
        KE(i) = 0;
        Fd(i) = 0;
        Wd(i) = 0;
        diff(i) = 0;
    else
    vave(i) = (v(i)+v(i-1))/2;
    time(i) = ((z(i)-z(i-1))/vave(i))+time(i-1);
    PE(i) = m*g*(z(i)-z(i-1))/32.2;
    KE(i) = 0.5*m*((v(i)^2)-(v(i-1)^2))/32.2;
    Fd(i) = 0.5*p*(vave(i)^2)*(pi/4)*(d^2)*(0.27)/32.2;
    Wd(i) = Fd(i)*(z(i)-z(i-1));
    diff(i) = (KE(i)+Wd(i))-PE(i);
    while diff(i) <= 0.001
        v(i) = v(i)+0.01;
        vave(i) = (v(i)+v(i-1))/2;
        time(i) = ((z(i)-z(i-1))/vave(i))+time(i-1);
        PE(i) = m*g*(z(i)-z(i-1))/32.2;
        KE(i) = 0.5*m*((v(i)^2)-(v(i-1)^2))/32.2;
        Fd(i) = 0.5*p*(vave(i)^2)*(pi()/4)*(d^2)*(0.27)/32.2;
        Wd(i) = Fd(i)*(z(i)-z(i-1));
        diff(i) = (KE(i)+Wd(i))-PE(i);
    end
    end
end
```

Results

From Matlab Code, time to hit ground: 1.766 s

Table 21 - Results from Dropped Ball Test

Person	Drop	Time	Residual
Jon	1	1.8	-0.034
	2	1.82	-0.054
	3	1.82	-0.054
	4	1.87	-0.104
	5	1.83	-0.064
	6	1.81	-0.044
	7	1.69	0.076
	8	1.74	0.026
	9	1.81	-0.044
	10	1.82	-0.054
Kevin	1	1.76	0.006
	2	1.74	0.026
	3	1.76	0.006
	4	1.7	0.066
	5	1.8	-0.034
	6	1.87	-0.104
	7	1.75	0.016
	8	1.74	0.026
	9	1.81	-0.044
	10	1.94	-0.174
Annaliese	1	1.87	-0.104
	2	1.83	-0.064
	3	1.81	-0.044
	4	1.75	0.016
	5	1.88	-0.114

Average Time: 1.8008 s

Standard Deviation: 0.0595 s

Comparison

Percent Difference: 1.951%

Percent Error: 1.971%

APPENDIX C

Matlab Code for No Spin Case

```
% Analysis of the 2015 Season Baseball in 4 Seam configuration without
lift
clear

% Constants
tic
weight = 0.3125; % weight (lbf)
yi = 3; % initial height (ft)
fprintf('For the 2015 Season Baseball in 4 seam configuration,\n')
vimph = input('Input the initial velocity in mph:');
theta_degrees = input('Input the initial angle in degrees:');
vi = vimph*(88/60); % initial velocity (ft/s)
theta_initial = theta_degrees*pi/180; % inital angle (radians)

if vi > 117.3
    Cd = 1.05*(.185+.000511*vi);
else
    Cd = 1.05*(0.435-0.001619*vi);
end
w = 1400; % speed of rotation (rpm)
p = 0.075; % denisty of air at 70degF (lbm/ft3)
d = 2.89; % baseball diameter (inches)
s = (pi/4)*(d/12)^2; % cross-sectional area of baseball (ft2)

% Changing Factors. Setting up for preallocation

x = transpose(0:1:450); %Change in horizontal direction (ft)
V = zeros(length(x),1); %Velocity (ft/s)
Vx = zeros(length(x),1); %Velocity in the x direction (ft/s)
Vxavg = zeros(length(x),1); %Avg velocity in the x direction (ft/s)
Vy = zeros(length(x),1); %Velocity in the y direction (ft/s)
Vyavg = zeros(length(x),1); %Avg velocity in the y direction (ft/s)
Cdx = zeros(length(x),1); %Drag Coefficient
Cdy = zeros(length(x),1); %Drag Coefficient
dt = zeros(length(x),1); %Change in time (s)
t = zeros(length(x),1); % Total time (s)
Fdx = zeros(length(x),1); % Drag in X Direction (lbf)
Fdy = zeros(length(x),1); % Drag in y Direction (lbf)
FL = zeros(length(x),1);  % Lift(lbf)
FLx = zeros(length(x),1); % Lift in X Direction (lbf)
FLy = zeros(length(x),1);  % Lift in Y Direction (lbf)
KEx = zeros(length(x),1); % Kinetic Energy (ft-lbf)
KEy = zeros(length(x),1); % Kinetic Energy (ft-lbf)
Wx = zeros(length(x),1); %Work in the x direction (drag and lift)*(dx)
(ft-lbf)
diffx = zeros(length(x),1); %Difference between (KEx-out + Work) and
(KEx-in) (ft-lbf)
dy = zeros(length(x),1); %Change in Y (ft)
y = zeros(length(x),1); %Total distance from ground (ft)
Wy = zeros(length(x),1); %Work in the y direction (drag - lift)*(dy)
(ft-lbf)
```

```
diffy = zeros(length(x),1); %Difference between (KEy-out - KEy-in +
Work) and (PEy) (ft-lbf)
PEy = zeros(length(x),1); %Potential Energy, mgy (ft-lbf)
dPE = zeros(length(x),1);%Change in Potential Energy (ft-lbf)
theta = zeros(length(x),1); %Angle (radians)

% Running through to minimize difference
n = length(x);
for i = 1:n
%The initial conditions are determined by guessing initial velocity and
angle. The first loop (i == 1) sets this up.
    if i == 1
        V(i) = vi;
        Vx(i) = V(i)*cos(theta_initial);
        FL(i) = 0; % (6.4E-07)*w*vi;
        KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
        Vy(i) = V(i)*sin(theta_initial);
        KEy(i) = 0.5*weight*(Vy(i)^2)/32.2;
        y(i) = yi;
        PEy(i) = weight*y(i);
        theta(i) = theta_initial;
    else
%For i >= 2, we run through the rest of the equations. In the next
section, we will guess values of
%Vx and Vy until the difference (diffx, diffy) are minimized).
        Vx(i) = Vx(i-1);
        if Vx(i) > 117.3
            Cdx(i) = 1.05*(0.185+0.000511*Vx(i));
        else
            Cdx(i) = 1.05*(0.435-0.001619*Vx(i));
        end
        Vxavg(i) = (Vx(i)+Vx(i-1))/2;
        dt(i) = (x(i)-x(i-1))/Vxavg(i);
        t(i) = dt(i)+t(i-1);
        Fdx(i) = 0.5*p*(Vxavg(i)^2)*s*Cdx(i)/32.2; %Fd = 1/2p(V^2)sCd
        FL(i) = 0; %No lift case
        FLx(i) = 0;
        Wx(i) = (Fdx(i)+FLx(i))*(x(i)-x(i-1));
        KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
        diffx(i) = KEx(i)+Wx(i)-KEx(i-1);

        Vy(i) = Vy(i-1);
        if Vy(i-1) < .1
            Vy(i) = -1*abs(Vy(i-1));
        end
        Vyavg(i) = (Vy(i)+Vy(i-1))/2;
        if Vy(i) > 117.3
            Cdy(i) = 1.05*(0.185+0.000511*Vy(i));
        else
            Cdy(i) = 1.05*(0.435-0.001619*Vy(i));
        end
        dy(i) = Vy(i)*dt(i);
        y(i) = dy(i) + y(i-1);
        Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd = 1/2p(V^2)sCd
        FLy(i) = 0; %no lift case
        Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
```

```
        KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
        PEy(i) = weight*y(i);
        dPE(i) = PEy(i-1)-PEy(i);
        diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
        %To minimize diffx, we use a while function to decrease Vx
until
        %diffx is less than a certain value
            while diffx(i) > 0.0001
                Vx(i) = Vx(i)-.0001;
                if Vx(i) > 117.3
                    Cdx(i) = 1.05*(0.185+0.000511*Vx(i));
                else
                    Cdx(i) = 1.05*(0.435-0.001619*Vx(i));
                end
                Vxavg(i) = (Vx(i)+Vx(i-1))/2;
                dt(i) = (x(i)-x(i-1))/Vxavg(i);
                t(i) = dt(i)+t(i-1);
                Fdx(i) = 0.5*p*(Vxavg(i)^2)*s*Cdx(i)/32.2; %Fd =
1/2p(V^2)sCd
                FL(i) = 0; %no lift case
                FLx(i) = 0;
                Wx(i) = (Fdx(i)+FLx(i))*(x(i)-x(i-1));
                KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
                diffx(i) = KEx(i)+Wx(i)-KEx(i-1);
            end

        %To minimize diffx, we use a while function to decrease Vx
until
        %diffx is less than a certain value
            while abs(diffy(i)) > 0.0001
                 Vy(i) = Vy(i)-.00001;
                if Vy(i) < 0.1
                    while abs(diffy(i)) > 0.0001
                    Vy(i) = -1*abs(Vy(i));
                    Vy(i) = Vy(i)-0.0001;
                    if Vy(i) > 117.3
                        Cdy(i) = 1.05*(0.185+0.000511*Vy(i));
                    else
                        Cdy(i) = 1.05*(0.435-0.001619*Vy(i));
                    end
                    Vyavg(i) = (Vy(i)+Vy(i-1))/2;
                    dy(i) = Vy(i)*dt(i);
                    y(i) = dy(i) + y(i-1);
                    Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2;
                    FLy(i) = 0; %no lift case
                    Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
                    KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
                    PEy(i) = weight*y(i);
                    dPE(i) = PEy(i-1)-PEy(i);
                    diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
                    end
                end
                Vyavg(i) = (Vy(i)+Vy(i-1))/2;
                if Vy(i) > 117.3
                    Cdy(i) = 1.05*(0.185+0.000511*Vy(i));
                else
                    Cdy(i) = 1.05*(0.435-0.001619*Vy(i));
```

```
                end
                dy(i) = Vy(i)*dt(i);
                y(i) = dy(i) + y(i-1);
                Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd =
1/2p(V^2)sCd
                FLy(i) = 0; %no lift case
                Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
                KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
                PEy(i) = weight*y(i);
                dPE(i) = PEy(i-1)-PEy(i);
                diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
            end

        theta(i) = atan(Vy(i)/Vx(i));
        V(i) = sqrt((Vy(i)^2)+(Vx(i)^2));
        if x(i) == 60
            tot_time = t(i,1);
        end
      end
end

for i = 2:n
    if abs(y(i)) < abs(y(i-1))
        ground_time = t(i);
        total_x = x(i);
    end
end

run_time = toc;
fprintf('\n');
fprintf('For the 2015 season baseball ball in the 4 seam
configuration,\nreturning from the bat at %f mph and at an angle of %f
degrees,\n', vimph, theta_degrees);
fprintf(1, 'Program Run Time: %f seconds \n', run_time);
fprintf(1, 'Final time to travel to pitchers mound: %f seconds.\n',
tot_time);
fprintf(1, 'Final time to hit ground: %f seconds.\n', ground_time);
fprintf(1, 'Total horizontal distance traveled: %f feet.\n', total_x);
fprintf('\n');
```

APPENDIX D

Matlab Codes for Batted Balls

Old High Seam, 2 Seam Configuration

```
% Analysis of the Pre-2015 Season Baseball in 2 Seam configuration

% Constants
tic
weight = 0.3125; % weight (lbf)
yi = 3; % initial height (ft)
fprintf('For the Pre-2015 Season Baseball in 2 seam configuration,\n')
vimph = input('Input the initial velocity in mph:');
theta_degrees = input('Input the initial angle in degrees:');
vi = vimph*(88/60); % initial velocity (ft/s)
theta_initial = theta_degrees*pi/180; % inital angle (radians)
%Adding in a drag correction factor of 5%
if vi > 147
    Cd = 1.05*(0.33);
else
    Cd = 1.05*(0.615-0.00193*vi);
end

w = 1400; % speed of rotation (rpm)
p = 0.075; % density of air at 70degF (lbm/ft3)
d = 2.89; % baseball diameter (inches)
s = (pi/4)*(d/12)^2; % cross-sectional area of baseball (ft2)

% Changing Factors. Setting up for preallocation

x = transpose(0:1:450); %Change in horizontal direction (ft)
V = zeros(length(x),1); %Velocity (ft/s)
Vx = zeros(length(x),1); %Velocity in the x direction (ft/s)
Vxavg = zeros(length(x),1); %Avg velocity in the x direction (ft/s)
Vy = zeros(length(x),1); %Velocity in the y direction (ft/s)
Vyavg = zeros(length(x),1); %Avg velocity in the y direction (ft/s)
Cdx = zeros(length(x),1); %Drag Coefficient
Cdy = zeros(length(x),1); %Drag Coefficient
dt = zeros(length(x),1); %Change in time (s)
t = zeros(length(x),1); % Total time (s)
Fdx = zeros(length(x),1); % Drag in X Direction (lbf)
Fdy = zeros(length(x),1); % Drag in y Direction (lbf)
FL = zeros(length(x),1);  % Lift(lbf)
FLx = zeros(length(x),1); % Lift in X Direction (lbf)
FLy = zeros(length(x),1);  % Lift in Y Direction (lbf)
KEx = zeros(length(x),1); % Kinetic Energy (ft-lbf)
KEy = zeros(length(x),1); % Kinetic Energy (ft-lbf)
Wx = zeros(length(x),1); %Work in the x direction (drag and lift)*(dx)
(ft-lbf)
diffx = zeros(length(x),1); %Difference between (KEx-out + Work) and
(KEx-in) (ft-lbf)
dy = zeros(length(x),1); %Change in Y (ft)
y = zeros(length(x),1); %Total distance from ground (ft)
```

```
Wy = zeros(length(x),1); %Work in the y direction (drag - lift)*(dy)
(ft-lbf)
diffy = zeros(length(x),1); %Difference between (KEy-out - KEy-in +
Work) and (PEy) (ft-lbf)
PEy = zeros(length(x),1); %Potential Energy, mgy (ft-lbf)
dPE = zeros(length(x),1);%Change in Potential Energy (ft-lbf)
theta = zeros(length(x),1); %Angle (radians)

% Running through to minimize difference
n = length(x);

for i = 1:n

%The initial conditions are determined by guessing initial velocity and
angle. The first loop (i == 1) sets this up.
    if i == 1
        V(i) = vi;
        Vx(i) = V(i)*cos(theta_initial);
        if Vx(i) > 147
            Cdx(i) = 1.05*(0.33);
        else
            Cdx(i) = 1.05*(0.615-.00193*Vx(i));
        end
        FL(i) = (6.4E-07)*w*vi;
        KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
        Vy(i) = V(i)*sin(theta_initial);
        KEy(i) = 0.5*weight*(Vy(i)^2)/32.2;
        y(i) = yi;
        PEy(i) = weight*y(i);
        theta(i) = theta_initial;
    else
%For i >= 2, we run through the rest of the equations. In the next
section, we will guess values of
%Vx and Vy until the difference (diffx, diffy) are minimized).
        Vx(i) = Vx(i-1);
        if Vx(i) > 147
            Cdx(i) = 1.05*(0.33);
        else
            Cdx(i) = 1.05*(0.615-.00193*Vx(i));
        end
        Vxavg(i) = (Vx(i)+Vx(i-1))/2;
        dt(i) = (x(i)-x(i-1))/Vxavg(i);
        t(i) = dt(i)+t(i-1);
        Fdx(i) = 0.5*p*(Vxavg(i)^2)*s*Cdx(i)/32.2; %Fd = 1/2p(V^2)sCd
        FL(i) = (6.4E-07)*w*V(i-1);
        FLx(i) = FL(i)*sin(theta(i-1));
        Wx(i) = (Fdx(i)+FLx(i))*(x(i)-x(i-1));
        KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
        diffx(i) = KEx(i)+Wx(i)-KEx(i-1);

        Vy(i) = Vy(i-1);
        if Vy(i-1) < .1
            Vy(i) = -1*abs(Vy(i-1));
        end
        Vyavg(i) = (Vy(i)+Vy(i-1))/2;
        if Vy(i) > 147
```

```
            Cdy(i) = 1.05*(0.33);
        else
            Cdy(i) = 1.05*(0.615-.00193*Vy(i));
        end
        dy(i) = Vy(i)*dt(i);
        y(i) = dy(i) + y(i-1);
        Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd = 1/2p(V^2)sCd
        FLy(i) = FL(i)*cos(theta(i-1));
        Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
        KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
        PEy(i) = weight*y(i);
        dPE(i) = PEy(i-1)-PEy(i);
        diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
        %To minimize diffx, we use a while function to decrease Vx
until
        %diffx is less than a certain value
            while diffx(i) > 0.0001
                Vx(i) = Vx(i)-.0001;
                if Vx(i) > 147
                    Cdx(i) = 1.05*(0.33);
                else
                    Cdx(i) = 1.05*(0.615-.00193*Vx(i));
                end
                Vxavg(i) = (Vx(i)+Vx(i-1))/2;
                dt(i) = (x(i)-x(i-1))/Vxavg(i);
                t(i) = dt(i)+t(i-1);
                Fdx(i) = 0.5*p*(Vxavg(i)^2)*s*Cdx(i)/32.2; %Fd =
1/2p(V^2)sCd
                FL(i) = (6.4E-07)*w*V(i-1);
                FLx(i) = FL(i)*sin(theta(i-1));
                Wx(i) = (Fdx(i)+FLx(i))*(x(i)-x(i-1));
                KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
                diffx(i) = KEx(i)+Wx(i)-KEx(i-1);
            end

        %To minimize diffx, we use a while function to decrease Vx
until
        %diffx is less than a certain value
            while abs(diffy(i)) > 0.0001
                 Vy(i) = Vy(i)-.00001;
                if Vy(i) < 0.1
                    while abs(diffy(i)) > 0.0001
                    Vy(i) = -1*abs(Vy(i));
                    Vy(i) = Vy(i)-0.0001;
                    if Vy(i) > 147
                        Cdy(i) = 1.05*(0.33);
                    else
                        Cdy(i) = 1.05*(0.615-.00193*Vy(i));
                    end
                    Vyavg(i) = (Vy(i)+Vy(i-1))/2;
                    dy(i) = Vy(i)*dt(i);
                    y(i) = dy(i) + y(i-1);
                    Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd =
1/2p(V^2)sCd
                    FLy(i) = FL(i)*cos(theta(i-1));
                    %Wy(i) = -((Fdy(i)*(dy(i)))+(FLy(i)*(dy(i))));
                    Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
```

```
                    KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
                    PEy(i) = weight*y(i);
                    dPE(i) = PEy(i-1)-PEy(i);
                    diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
                    end
                end
                Vyavg(i) = (Vy(i)+Vy(i-1))/2;
                     if Vy(i) > 147
                        Cdy(i) = 1.05*(0.33);
                    else
                        Cdy(i) = 1.05*(0.615-.00193*Vy(i));
                     end
                dy(i) = Vy(i)*dt(i);
                y(i) = dy(i) + y(i-1);
                Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd =
1/2p(V^2)sCd
                FLy(i) = FL(i)*cos(theta(i-1));
                Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
                % Wy(i) = (Fdy(i)-FLy(i))*(dy(i));
                KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
                PEy(i) = weight*y(i);
                dPE(i) = PEy(i-1)-PEy(i);
                diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
            end

        theta(i) = atan(Vy(i)/Vx(i));
        V(i) = sqrt((Vy(i)^2)+(Vx(i)^2));
        if x(i) == 60
            tot_time = t(i,1);
        end


    end

end

for i = 2:n
    if abs(y(i)) < abs(y(i-1))
        ground_time = t(i);
        total_x = x(i);
    end
end


run_time = toc;
fprintf('\n');
fprintf('For the Pre-2015 season baseball ball in the 2 seam
configuration,\nreturning from the bat at %f mph and \nat an angle of
%f degrees,\n', vimph, theta_degrees);
fprintf(1, 'Program Run Time: %f seconds \n', run_time);
fprintf(1, 'Final time to travel to pitchers mound: %f seconds.\n',
tot_time);
fprintf(1, 'Final time to hit ground: %f seconds.\n', ground_time);
fprintf(1, 'Total horizontal distance traveled: %f feet.\n', total_x);
fprintf('\n');
```

Old High Seam, 4 Seam Configuration

```
% Analysis of the Pre-2015 Season Baseball in 4 Seam configuration

% Constants
tic
weight = 0.3125; % weight (lbf)
yi = 3; % initial height (ft)
fprintf('For the Pre-2015 Season Baseball in 4 seam configuration,\n')
vimph = input('Input the initial velocity in mph:');
theta_degrees = input('Input the initial angle in degrees:');
%vimph = 115; % initial velocity (mph)
vi = vimph*(88/60); % initial velocity (ft/s)
%theta_degrees = 20; %initial angle (degrees)
theta_initial = theta_degrees*pi/180; % inital angle (radians)

% Adding in a drag correction factor of 5%
if vi > 129.2
    Cd = 1.05*(0.27);
else
    Cd = 1.05*(0.53-0.00201*vi);
end
w = 1400; % speed of rotation (rpm)
p = 0.075; % denisty of air at 70degF (lbm/ft3)
d = 2.889; % baseball diameter (inches)
s = (pi/4)*(d/12)^2; % cross-sectional area of baseball (ft2)

% Changing Factors. Setting up for preallocation

x = transpose(0:1:450); %Change in horizontal direction (ft)
V = zeros(length(x),1); %Velocity (ft/s)
Vx = zeros(length(x),1); %Velocity in the x direction (ft/s)
Vxavg = zeros(length(x),1); %Avg velocity in the x direction (ft/s)
Vy = zeros(length(x),1); %Velocity in the y direction (ft/s)
Vyavg = zeros(length(x),1); %Avg velocity in the y direction (ft/s)
Cdx = zeros(length(x),1); %Drag Coefficient
Cdy = zeros(length(x),1); %Drag Coefficient
dt = zeros(length(x),1); %Change in time (s)
t = zeros(length(x),1); % Total time (s)
Fdx = zeros(length(x),1); % Drag in X Direction (lbf)
Fdy = zeros(length(x),1); % Drag in y Direction (lbf)
FL = zeros(length(x),1);  % Lift(lbf)
FLx = zeros(length(x),1); % Lift in X Direction (lbf)
FLy = zeros(length(x),1);  % Lift in Y Direction (lbf)
KEx = zeros(length(x),1); % Kinetic Energy (ft-lbf)
KEy = zeros(length(x),1); % Kinetic Energy (ft-lbf)
Wx = zeros(length(x),1); %Work in the x direction (drag and lift)*(dx)
(ft-lbf)
diffx = zeros(length(x),1); %Difference between (KEx-out + Work) and
(KEx-in) (ft-lbf)
dy = zeros(length(x),1); %Change in Y (ft)
y = zeros(length(x),1); %Total distance from ground (ft)
Wy = zeros(length(x),1); %Work in the y direction (drag - lift)*(dy)
(ft-lbf)
```

```
diffy = zeros(length(x),1); %Difference between (KEy-out - KEy-in +
Work) and (PEy) (ft-lbf)
PEy = zeros(length(x),1); %Potential Energy, mgy (ft-lbf)
dPE = zeros(length(x),1);%Change in Potential Energy (ft-lbf)
theta = zeros(length(x),1); %Angle (radians)

% Running through to minimize difference
n = length(x);

for i = 1:n

%The initial conditions are determined by guessing initial velocity and
angle. The first loop (i == 1) sets this up.
    if i == 1
        V(i) = vi;
        Vx(i) = V(i)*cos(theta_initial);
        if Vx(i) > 130
            Cdx(i) = 1.05*(0.2664);
        else
            Cdx(i) = 1.05*(0.5554-.0022*Vx(i));
        end
        FL(i) = (6.4E-07)*w*vi;
        KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
        Vy(i) = V(i)*sin(theta_initial);
        KEy(i) = 0.5*weight*(Vy(i)^2)/32.2;
        y(i) = yi;
        PEy(i) = weight*y(i);
        theta(i) = theta_initial;
    else
%For i >= 2, we run through the rest of the equations. In the next
section, we will guess values of
%Vx and Vy until the difference (diffx, diffy) are minimized).
        Vx(i) = Vx(i-1);
        if Vx(i) > 129.2
            Cdx(i) = 1.05*(0.27);
        else
            Cdx(i) = 1.05*(0.53-0.00201*Vx(i));
        end
        Vxavg(i) = (Vx(i)+Vx(i-1))/2;
        dt(i) = (x(i)-x(i-1))/Vxavg(i);
        t(i) = dt(i)+t(i-1);
        Fdx(i) = 0.5*p*(Vxavg(i)^2)*s*Cdx(i)/32.2; %Fd = 1/2p(V^2)sCd
        FL(i) = (6.4E-07)*w*V(i-1);
        FLx(i) = FL(i)*sin(theta(i-1));
        Wx(i) = (Fdx(i)+FLx(i))*(x(i)-x(i-1));
        KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
        diffx(i) = KEx(i)+Wx(i)-KEx(i-1);

        Vy(i) = Vy(i-1);
        if Vy(i-1) < .1
            Vy(i) = -1*abs(Vy(i-1));
        end
        Vyavg(i) = (Vy(i)+Vy(i-1))/2;
        if Vy(i) > 129.2
            Cdy(i) = 1.05*0.27;
        else
```

```
            Cdy(i) = 1.05*(0.53-0.00201*Vy(i));
        end
        dy(i) = Vy(i)*dt(i);
        y(i) = dy(i) + y(i-1);
        Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd = 1/2p(V^2)sCd
        FLy(i) = FL(i)*cos(theta(i-1));
        Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
        KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
        PEy(i) = weight*y(i);
        dPE(i) = PEy(i-1)-PEy(i);
        diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
        %To minimize diffx, we use a while function to decrease Vx
until
        %diffx is less than a certain value
            while diffx(i) > 0.0001
                Vx(i) = Vx(i)-.0001;
                if Vx(i) > 129.2
                    Cdx(i) = 1.05*0.27;
                else
                    Cdx(i) = 1.05*(0.53-0.00201*Vx(i));
                end
                Vxavg(i) = (Vx(i)+Vx(i-1))/2;
                dt(i) = (x(i)-x(i-1))/Vxavg(i);
                t(i) = dt(i)+t(i-1);
                Fdx(i) = 0.5*p*(Vxavg(i)^2)*s*Cdx(i)/32.2; %Fd =
1/2p(V^2)sCd
                FL(i) = (6.4E-07)*w*V(i-1);
                FLx(i) = FL(i)*sin(theta(i-1));
                Wx(i) = (Fdx(i)+FLx(i))*(x(i)-x(i-1));
                KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
                diffx(i) = KEx(i)+Wx(i)-KEx(i-1);
            end

        %To minimize diffx, we use a while function to decrease Vx
until
        %diffx is less than a certain value
            while abs(diffy(i)) > 0.0001
                 Vy(i) = Vy(i)-.00001;
                if Vy(i) < 0.1
                    while abs(diffy(i)) > 0.0001
                    Vy(i) = -1*abs(Vy(i));
                    Vy(i) = Vy(i)-0.0001;
                    if Vy(i) > 129.2
                        Cdy(i) = 1.05*(0.27);
                    else
                        Cdy(i) = 1.05*(0.53-0.00201*Vy(i));
                    end
                    Vyavg(i) = (Vy(i)+Vy(i-1))/2;
                    dy(i) = Vy(i)*dt(i);
                    y(i) = dy(i) + y(i-1);
                    Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd =
1/2p(V^2)sCd
                    FLy(i) = FL(i)*cos(theta(i-1));
                    Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
                    KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
                    PEy(i) = weight*y(i);
                    dPE(i) = PEy(i-1)-PEy(i);
```

```
                diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
                end
            end
            Vyavg(i) = (Vy(i)+Vy(i-1))/2;
            if Vy(i) > 129.2
                Cdy(i) = 1.05*(0.27);
            else
                Cdy(i) = 1.05*(0.53-0.00201*Vy(i));
            end
            dy(i) = Vy(i)*dt(i);
            y(i) = dy(i) + y(i-1);
            Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd =
1/2p(V^2)sCd
            FLy(i) = FL(i)*cos(theta(i-1));
            Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
            KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
            PEy(i) = weight*y(i);
            dPE(i) = PEy(i-1)-PEy(i);
            diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
        end

        theta(i) = atan(Vy(i)/Vx(i));
        V(i) = sqrt((Vy(i)^2)+(Vx(i)^2));
        if x(i) == 60
            tot_time = t(i,1);
        end


    end

end

for i = 2:n
    if abs(y(i)) < abs(y(i-1))
        ground_time = t(i);
        total_x = x(i);
    end
end


run_time = toc;
fprintf('\n');
fprintf('For the Pre-2015 season baseball ball in the 4 seam
configuration,\nreturning from the bat at %f mph and at an angle of %f
degrees,\n', vimph, theta_degrees);
fprintf(1, 'Program Run Time: %f seconds \n', run_time);
fprintf(1, 'Final time to travel to pitchers mound: %f seconds.\n',
tot_time);
fprintf(1, 'Final time to hit ground: %f seconds.\n', ground_time);
fprintf(1, 'Total horizontal distance traveled: %f feet.\n', total_x);
fprintf('\n');
```

New Flat Seam, 2 Seam Configuration

```
% Analysis of the 2015 Season Baseball in 2 Seam configuration
```

```
clear

% Constants
tic
weight = 0.3125; % weight (lbf)
yi = 3; % initial height (ft)
fprintf('For the 2015 Season Baseball in 2 seam configuration,\n')
vimph = input('Input the initial velocity in mph:');
theta_degrees = input('Input the initial angle in degrees:');
%vimph = 115; % initial velocity (mph)
vi = vimph*(88/60); % initial velocity (ft/s)
%theta_degrees = 20; %initial angle (degrees)
theta_initial = theta_degrees*pi/180; % inital angle (radians)
if vi > 117
    Cd = 1.05*0.32;
else
    Cd = 1.05*(0.78-.00392*vi);
end
w = 1400; % speed of rotation (rpm)
p = 0.075; % denisty of air at 70degF (lbm/ft3)
d = 2.878; % baseball diameter (inches)
s = (pi/4)*(d/12)^2; % cross-sectional area of baseball (ft2)

% Changing Factors. Setting up for preallocation

x = transpose(0:1:450); %Change in horizontal direction (ft)
V = zeros(length(x),1); %Velocity (ft/s)
Vx = zeros(length(x),1); %Velocity in the x direction (ft/s)
Vxavg = zeros(length(x),1); %Avg velocity in the x direction (ft/s)
Vy = zeros(length(x),1); %Velocity in the y direction (ft/s)
Vyavg = zeros(length(x),1); %Avg velocity in the y direction (ft/s)
Cdx = zeros(length(x),1); %Drag Coefficient
Cdy = zeros(length(x),1); %Drag Coefficient
dt = zeros(length(x),1); %Change in time (s)
t = zeros(length(x),1); % Total time (s)
Fdx = zeros(length(x),1); % Drag in X Direction (lbf)
Fdy = zeros(length(x),1); % Drag in y Direction (lbf)
FL = zeros(length(x),1);  % Lift(lbf)
FLx = zeros(length(x),1); % Lift in X Direction (lbf)
FLy = zeros(length(x),1);  % Lift in Y Direction (lbf)
KEx = zeros(length(x),1); % Kinetic Energy (ft-lbf)
KEy = zeros(length(x),1); % Kinetic Energy (ft-lbf)
Wx = zeros(length(x),1); %Work in the x direction (drag and lift)*(dx)
(ft-lbf)
diffx = zeros(length(x),1); %Difference between (KEx-out + Work) and
(KEx-in) (ft-lbf)
dy = zeros(length(x),1); %Change in Y (ft)
y = zeros(length(x),1); %Total distance from ground (ft)
Wy = zeros(length(x),1); %Work in the y direction (drag - lift)*(dy)
(ft-lbf)
diffy = zeros(length(x),1); %Difference between (KEy-out - KEy-in +
Work) and (PEy) (ft-lbf)
PEy = zeros(length(x),1); %Potential Energy, mgy (ft-lbf)
dPE = zeros(length(x),1);%Change in Potential Energy (ft-lbf)
theta = zeros(length(x),1); %Angle (radians)
```

```
% Running through to minimize difference
n = length(x);

for i = 1:n

%The initial conditions are determined by guessing initial velocity and
angle. The first loop (i == 1) sets this up.
    if i == 1
        V(i) = vi;
        Vx(i) = V(i)*cos(theta_initial);
        if Vx(i) > 117
            Cdx(i) = 1.05*(0.32);
        else
            Cdx(i) = 1.05*(0.78-.00392*Vx(i));
        end
        FL(i) = (6.4E-07)*w*vi;
        KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
        Vy(i) = V(i)*sin(theta_initial);
        KEy(i) = 0.5*weight*(Vy(i)^2)/32.2;
        y(i) = yi;
        PEy(i) = weight*y(i);
        theta(i) = theta_initial;
    else
%For i >= 2, we run through the rest of the equations. In the next
section, we will guess values of
%Vx and Vy until the difference (diffx, diffy) are minimized).
        Vx(i) = Vx(i-1);
        if Vx(i) > 117
            Cdx(i) = 1.05*(0.32);
        else
            Cdx(i) = 1.05*(0.78-.00392*Vx(i));
        end
        Vxavg(i) = (Vx(i)+Vx(i-1))/2;
        dt(i) = (x(i)-x(i-1))/Vxavg(i);
        t(i) = dt(i)+t(i-1);
        Fdx(i) = 0.5*p*(Vxavg(i)^2)*s*Cdx(i)/32.2; %Fd = 1/2p(V^2)sCd
        FL(i) = (6.4E-07)*w*V(i-1);
        FLx(i) = FL(i)*sin(theta(i-1));
        Wx(i) = (Fdx(i)+FLx(i))*(x(i)-x(i-1));
        KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
        diffx(i) = KEx(i)+Wx(i)-KEx(i-1);

        Vy(i) = Vy(i-1);
        if Vy(i-1) < .1
            Vy(i) = -1*abs(Vy(i-1));
        end
        Vyavg(i) = (Vy(i)+Vy(i-1))/2;
        if Vy(i) > 117
            Cdy(i) = 1.05*(0.32);
        else
            Cdy(i) = 1.05*(0.78-.00392*Vy(i));
        end
        dy(i) = Vy(i)*dt(i);
        y(i) = dy(i) + y(i-1);
        Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd = 1/2p(V^2)sCd
        FLy(i) = FL(i)*cos(theta(i-1));
```

```
        Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
        KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
        PEy(i) = weight*y(i);
        dPE(i) = PEy(i-1)-PEy(i);
        diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
        %To minimize diffx, we use a while function to decrease Vx
until
        %diffx is less than a certain value
            while diffx(i) > 0.0001
                Vx(i) = Vx(i)-.0001;
                if Vx(i) > 117
                    Cdx(i) = 1.05*(0.32);
                else
                    Cdx(i) = 1.05*(0.78-.00392*Vx(i));
                end
                Vxavg(i) = (Vx(i)+Vx(i-1))/2;
                dt(i) = (x(i)-x(i-1))/Vxavg(i);
                t(i) = dt(i)+t(i-1);
                Fdx(i) = 0.5*p*(Vxavg(i)^2)*s*Cdx(i)/32.2; %Fd =
1/2p(V^2)sCd
                FL(i) = (6.4E-07)*w*V(i-1);
                FLx(i) = FL(i)*sin(theta(i-1));
                Wx(i) = (Fdx(i)+FLx(i))*(x(i)-x(i-1));
                KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
                diffx(i) = KEx(i)+Wx(i)-KEx(i-1);
            end

        %To minimize diffx, we use a while function to decrease Vx
until
        %diffx is less than a certain value
            while abs(diffy(i)) > 0.0001
                 Vy(i) = Vy(i)-.00001;
                if Vy(i) < 0.1
                    while abs(diffy(i)) > 0.0001
                    Vy(i) = -1*abs(Vy(i));
                    Vy(i) = Vy(i)-0.0001;
                    if Vy(i) > 117
                        Cdy(i) = 1.05*(0.32);
                    else
                        Cdy(i) = 1.05*(0.78-.00392*Vy(i));
                    end
                   Vyavg(i) = (Vy(i)+Vy(i-1))/2;
                    dy(i) = Vy(i)*dt(i);
                    y(i) = dy(i) + y(i-1);
                    Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd =
1/2p(V^2)sCd
                    FLy(i) = FL(i)*cos(theta(i-1));
                    Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
                    KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
                    PEy(i) = weight*y(i);
                    dPE(i) = PEy(i-1)-PEy(i);
                    diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
                    end
                end
                Vyavg(i) = (Vy(i)+Vy(i-1))/2;
                if Vy(i) > 117
                    Cdy(i) = 1.05*(0.32);
```

```
                else
                    Cdy(i) = 1.05*(0.78-.00392*Vy(i));
                end
                dy(i) = Vy(i)*dt(i);
                y(i) = dy(i) + y(i-1);
                Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd =
1/2p(V^2)sCd
                FLy(i) = FL(i)*cos(theta(i-1));
                Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
                KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
                PEy(i) = weight*y(i);
                dPE(i) = PEy(i-1)-PEy(i);
                diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
            end

        theta(i) = atan(Vy(i)/Vx(i));
        V(i) = sqrt((Vy(i)^2)+(Vx(i)^2));
        if x(i) == 60
            tot_time = t(i,1);
        end


    end

end

for i = 2:n
    if abs(y(i)) < abs(y(i-1))
        ground_time = t(i);
        total_x = x(i);
    end
end


run_time = toc;
fprintf('\n');
fprintf('For the 2015 season baseball ball in the 2 seam
configuration,\nreturning from the bat at %f mph and at an angle of %f
degrees,\n', vimph, theta_degrees);
fprintf(1, 'Program Run Time: %f seconds \n', run_time);
fprintf(1, 'Final time to travel to pitchers mound: %f seconds.\n',
tot_time);
fprintf(1, 'Final time to hit ground: %f seconds.\n', ground_time);
fprintf(1, 'Total horizontal distance traveled: %f feet.\n', total_x);
fprintf('\n');
```

New Flat Seam, 4 Seam Configuration

```
% Analysis of the 2015 Season Baseball in 4 Seam configuration
clear

% Constants
tic
weight = 0.3125; % weight (lbf)
```

```
yi = 3; % initial height (ft)
fprintf('For the 2015 Season Baseball in 4 seam configuration,\n')
vimph = input('Input the initial velocity in mph:');
theta_degrees = input('Input the initial angle in degrees:');
%vimph = 115; % initial velocity (mph)
vi = vimph*(88/60); % initial velocity (ft/s)
%theta_degrees = 20; %initial angle (degrees)
theta_initial = theta_degrees*pi/180; % inital angle (radians)
%Cdi = ((-4E-07)*(vi^3))+((0.0002*(vi^2)))-(0.0281*vi)+1.6811; % drag
coefficient, assuming constant
if vi > 117.3
    Cd = 1.05*(.185+.000511*vi);
else
    Cd = 1.05*(0.435-0.001619*vi);
end
w = 1400; % speed of rotation (rpm)
p = 0.075; % denisty of air at 70degF (lbm/ft3)
d = 2.89; % baseball diameter (inches)
s = (pi/4)*(d/12)^2; % cross-sectional area of baseball (ft2)

% Changing Factors. Setting up for preallocation

x = transpose(0:1:450); %Change in horizontal direction (ft)
V = zeros(length(x),1); %Velocity (ft/s)
Vx = zeros(length(x),1); %Velocity in the x direction (ft/s)
Vxavg = zeros(length(x),1); %Avg velocity in the x direction (ft/s)
Vy = zeros(length(x),1); %Velocity in the y direction (ft/s)
Vyavg = zeros(length(x),1); %Avg velocity in the y direction (ft/s)
Cdx = zeros(length(x),1); %Drag Coefficient
Cdy = zeros(length(x),1); %Drag Coefficient
dt = zeros(length(x),1); %Change in time (s)
t = zeros(length(x),1); % Total time (s)
Fdx = zeros(length(x),1); % Drag in X Direction (lbf)
Fdy = zeros(length(x),1); % Drag in y Direction (lbf)
FL = zeros(length(x),1);  % Lift(lbf)
FLx = zeros(length(x),1); % Lift in X Direction (lbf)
FLy = zeros(length(x),1);  % Lift in Y Direction (lbf)
KEx = zeros(length(x),1); % Kinetic Energy (ft-lbf)
KEy = zeros(length(x),1); % Kinetic Energy (ft-lbf)
Wx = zeros(length(x),1); %Work in the x direction (drag and lift)*(dx)
(ft-lbf)
diffx = zeros(length(x),1); %Difference between (KEx-out + Work) and
(KEx-in) (ft-lbf)
dy = zeros(length(x),1); %Change in Y (ft)
y = zeros(length(x),1); %Total distance from ground (ft)
Wy = zeros(length(x),1); %Work in the y direction (drag - lift)*(dy)
(ft-lbf)
diffy = zeros(length(x),1); %Difference between (KEy-out - KEy-in +
Work) and (PEy) (ft-lbf)
PEy = zeros(length(x),1); %Potential Energy, mgy (ft-lbf)
dPE = zeros(length(x),1);%Change in Potential Energy (ft-lbf)
theta = zeros(length(x),1); %Angle (radians)

% Running through to minimize difference
n = length(x);
```

```
for i = 1:n

%The initial conditions are determined by guessing initial velocity and
angle. The first loop (i == 1) sets this up.
    if i == 1
        V(i) = vi;
        Vx(i) = V(i)*cos(theta_initial);
        FL(i) = (6.4E-07)*w*vi;
        KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
        Vy(i) = V(i)*sin(theta_initial);
        KEy(i) = 0.5*weight*(Vy(i)^2)/32.2;
        y(i) = yi;
        PEy(i) = weight*y(i);
        theta(i) = theta_initial;
    else
%For i >= 2, we run through the rest of the equations. In the next
section, we will guess values of
%Vx and Vy until the difference (diffx, diffy) are minimized).
        Vx(i) = Vx(i-1);
        if Vx(i) > 117.3
            Cdx(i) = 1.05*(0.185+0.000511*Vx(i));
        else
            Cdx(i) = 1.05*(0.435-0.001619*Vx(i));
        end
        Vxavg(i) = (Vx(i)+Vx(i-1))/2;
        dt(i) = (x(i)-x(i-1))/Vxavg(i);
        t(i) = dt(i)+t(i-1);
        Fdx(i) = 0.5*p*(Vxavg(i)^2)*s*Cdx(i)/32.2; %Fd = 1/2p(V^2)sCd
        FL(i) = (6.4E-07)*w*V(i-1);
        FLx(i) = FL(i)*sin(theta(i-1));
        Wx(i) = (Fdx(i)+FLx(i))*(x(i)-x(i-1));
        KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
        diffx(i) = KEx(i)+Wx(i)-KEx(i-1);

        Vy(i) = Vy(i-1);
        if Vy(i-1) < .1
            Vy(i) = -1*abs(Vy(i-1));
        end
        Vyavg(i) = (Vy(i)+Vy(i-1))/2;
        if Vy(i) > 117.3
            Cdy(i) = 1.05*(0.185+0.000511*Vy(i));
        else
            Cdy(i) = 1.05*(0.435-0.001619*Vy(i));
        end
        dy(i) = Vy(i)*dt(i);
        y(i) = dy(i) + y(i-1);
        Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd = 1/2p(V^2)sCd
        FLy(i) = FL(i)*cos(theta(i-1));
        Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
        KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
        PEy(i) = weight*y(i);
        dPE(i) = PEy(i-1)-PEy(i);
        diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
        %To minimize diffx, we use a while function to decrease Vx
until
        %diffx is less than a certain value
```

```
            while diffx(i) > 0.0001
                Vx(i) = Vx(i)-.0001;
                if Vx(i) > 117.3
                    Cdx(i) = 1.05*(0.185+0.000511*Vx(i));
                else
                    Cdx(i) = 1.05*(0.435-0.001619*Vx(i));
                end
                Vxavg(i) = (Vx(i)+Vx(i-1))/2;
                dt(i) = (x(i)-x(i-1))/Vxavg(i);
                t(i) = dt(i)+t(i-1);
                Fdx(i) = 0.5*p*(Vxavg(i)^2)*s*Cdx(i)/32.2; %Fd =
1/2p(V^2)sCd
                FL(i) = (6.4E-07)*w*V(i-1);
                FLx(i) = FL(i)*sin(theta(i-1));
                Wx(i) = (Fdx(i)+FLx(i))*(x(i)-x(i-1));
                KEx(i) = 0.5*weight*(Vx(i)^2)/32.2;
                diffx(i) = KEx(i)+Wx(i)-KEx(i-1);
            end

        %To minimize diffx, we use a while function to decrease Vx
until
        %diffx is less than a certain value
            while abs(diffy(i)) > 0.0001
                 Vy(i) = Vy(i)-.00001;
                if Vy(i) < 0.1
                    while abs(diffy(i)) > 0.0001
                    Vy(i) = -1*abs(Vy(i));
                    Vy(i) = Vy(i)-0.0001;
                    if Vy(i) > 117.3
                        Cdy(i) = 1.05*(0.185+0.000511*Vy(i));
                    else
                        Cdy(i) = 1.05*(0.435-0.001619*Vy(i));
                    end
                    Vyavg(i) = (Vy(i)+Vy(i-1))/2;
                    dy(i) = Vy(i)*dt(i);
                    y(i) = dy(i) + y(i-1);
                    Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd =
1/2p(V^2)sCd
                    FLy(i) = FL(i)*cos(theta(i-1));
                    Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
                    KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
                    PEy(i) = weight*y(i);
                    dPE(i) = PEy(i-1)-PEy(i);
                    diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
                    end
                end
                Vyavg(i) = (Vy(i)+Vy(i-1))/2;
                if Vy(i) > 117.3
                    Cdy(i) = 1.05*(0.185+0.000511*Vy(i));
                else
                    Cdy(i) = 1.05*(0.435-0.001619*Vy(i));
                end
                dy(i) = Vy(i)*dt(i);
                y(i) = dy(i) + y(i-1);
                Fdy(i) = 0.5*p*(Vyavg(i)^2)*s*Cdy(i)/32.2; %Fd =
1/2p(V^2)sCd
                FLy(i) = FL(i)*cos(theta(i-1));
```

```
                Wy(i) = (Fdy(i)*abs(dy(i)))-(FLy(i)*(dy(i)));
                KEy(i) = abs(0.5*weight*(Vy(i)^2)/32.2);
                PEy(i) = weight*y(i);
                dPE(i) = PEy(i-1)-PEy(i);
                diffy(i) = KEy(i)-KEy(i-1)+Wy(i)-dPE(i);
            end

        theta(i) = atan(Vy(i)/Vx(i));
        V(i) = sqrt((Vy(i)^2)+(Vx(i)^2));
        if x(i) == 60
            tot_time = t(i,1);
        end


    end

end

for i = 2:n
    if abs(y(i)) < abs(y(i-1))
        ground_time = t(i);
        total_x = x(i);
    end
end


run_time = toc;
fprintf('\n');
fprintf('For the 2015 season baseball ball in the 4 seam
configuration,\nreturning from the bat at %f mph and at an angle of %f
degrees,\n', vimph, theta_degrees);
fprintf(1, 'Program Run Time: %f seconds \n', run_time);
fprintf(1, 'Final time to travel to pitchers mound: %f seconds.\n',
tot_time);
fprintf(1, 'Final time to hit ground: %f seconds.\n', ground_time);
fprintf(1, 'Total horizontal distance traveled: %f feet.\n', total_x);
fprintf('\n');
```

LIST OF REFERENCES

Associated Press. "NCAA Adopts New Baseball." ESPN. ESPN Internet Ventures, 5 Nov. 2013. Web. 12 Feb. 2015.

Bearman, P. W., and J. K. Harvey. "Golf Ball Aerodynamics." *The Aeronautical Quarterly* 27 (1976): 112-22. Print.

"Drag of a Sphere." *Drag of a Sphere*. Ed. Tom Benson. National Aeronautics and Space Administration. Web. 15 Feb. 2015.

Fox, Robert W., and Alan T. McDonald. *Introduction to Fluid Mechanics*. 4th ed. New York: J. Wiley, 1992. 464. Print.

"Ideal Lift of a Spinning Ball." *Ideal Lift of a Spinning Ball*. Ed. Tom Benson. National Aeronautics and Space Administration. Web. 15 Feb. 2015.

Johnson, Greg. "DI Committee Changes to Flat-seamed Baseballs for 2015 Championship." NCAA.com. NCAA, 05 Nov. 2013. Web. 12 Feb. 2015.

"Newton's Second Law." Newton's Second Law. The Physics Classroom, n.d. Web. 15 Feb. 2015.

Watts, Robert G., and Terry Bahill. *Keep Your Eye on the Ball*. 2nd ed. New York: W.H. Freeman, 2000. 19-81. Print.

Wolfson, Richard. *University Physics*. 2nd ed. Vol. 1. San Francisco: Pearson Education, 2012. 86-104. Print.

Young, Douglas, Doris Trachtman, Irving Scher, and Richard Schmidt. "High School and College Baseball Pitchers' Response and Glove Movements to Line Drives." *Journal of Applied Biomechanics* 22.1 (2006): 25-32. Web. 16 Feb. 2015. <http://www.humankinetics.com/acucustom/sitename/Documents/DocumentItem/5380.pdf>