HW8: Polynomial Fits

Austin Gardiner

October 10, 2025

1 Fitting a Second Order Polynomial

For each dataset, the equations shown in Fig. 1 are used to fit a 2nd order polynomial to the data for each direction. In these equations, x corresponds to the time data, t, while y corresponds to the direction that is being fit, x, y, or z, respectively. All fits follow the pattern shown in eq. 1.

$$y = A_1 + A_2 x + A_3 x^2 \tag{1}$$

$$A_1 = \frac{1}{\Delta}\left|\begin{bmatrix} \sum y_i \frac{1}{u_i^2} & \sum \frac{x_i}{u_i^2} & \sum \frac{x_i^2}{u_i^2} \\ \sum y_i \frac{x_i}{u_i^2} & \sum \frac{x_i^2}{u_i^2} & \sum \frac{x_i^3}{u_i^2} \\ \sum y_i \frac{x_i^2}{u_i^2} & \sum \frac{x_i^3}{u_i^2} & \sum \frac{x_i^4}{u_i^2} \end{bmatrix}\right|$$

$$A_2 = \frac{1}{\Delta}\left|\begin{bmatrix} \sum \frac{1}{u_i^2} & \sum y_i \frac{1}{u_i^2} & \sum \frac{x_i^2}{u_i^2} \\ \sum \frac{x_i}{u_i^2} & \sum y_i \frac{x_i}{u_i^2} & \sum \frac{x_i^3}{u_i^2} \\ \sum \frac{x_i^2}{u_i^2} & \sum y_i \frac{x_i^2}{u_i^2} & \sum \frac{x_i^4}{u_i^2} \end{bmatrix}\right|$$

$$A_3 = \frac{1}{\Delta}\left|\begin{bmatrix} \sum \frac{1}{u_i^2} & \sum \frac{x_i}{u_i^2} & \sum y_i \frac{1}{u_i^2} \\ \sum \frac{x_i}{u_i^2} & \sum \frac{x_i^2}{u_i^2} & \sum y_i \frac{x_i}{u_i^2} \\ \sum \frac{x_i^2}{u_i^2} & \sum \frac{x_i^3}{u_i^2} & \sum y_i \frac{x_i^2}{u_i^2} \end{bmatrix}\right|,$$

$$\text{with}\Delta = \left|\begin{bmatrix} \sum \frac{1}{u_i^2} & \sum \frac{x_i}{u_i^2} & \sum \frac{x_i^2}{u_i^2} \\ \sum \frac{x_i}{u_i^2} & \sum \frac{x_i^2}{u_i^2} & \sum \frac{x_i^3}{u_i^2} \\ \sum \frac{x_i^2}{u_i^2} & \sum \frac{x_i^3}{u_i^2} & \sum \frac{x_i^4}{u_i^2} \end{bmatrix}\right|$$

Figure 1: Equations to calculate the fit coefficients

These plots show the original data with a 0.5 in. uncertainty band as well as the polynomial fit computed using the coefficients. The position data is given in ft. so the uncertainty bands are difficult to discern due to their small size.

1.1 N20

For the N=20 dataset, the coefficients were calculated and rounded to four decimal places. The following equations have the coefficients placed in the fit equation as described in eq. 1. To check my work, I compared the fits calculated using Fig. 1 to the fit coefficients determined using numpy polyfit and they matched to at least four decimal places.

$$x = 0.5228 + -1.1974t + 2.8787t^2 \tag{2}$$

$$y = 54.1385 + -107.3560t + 7.1517t^2 \tag{3}$$

$$z = 6.2056 + -0.2005t + -20.1755t^2 \tag{4}$$

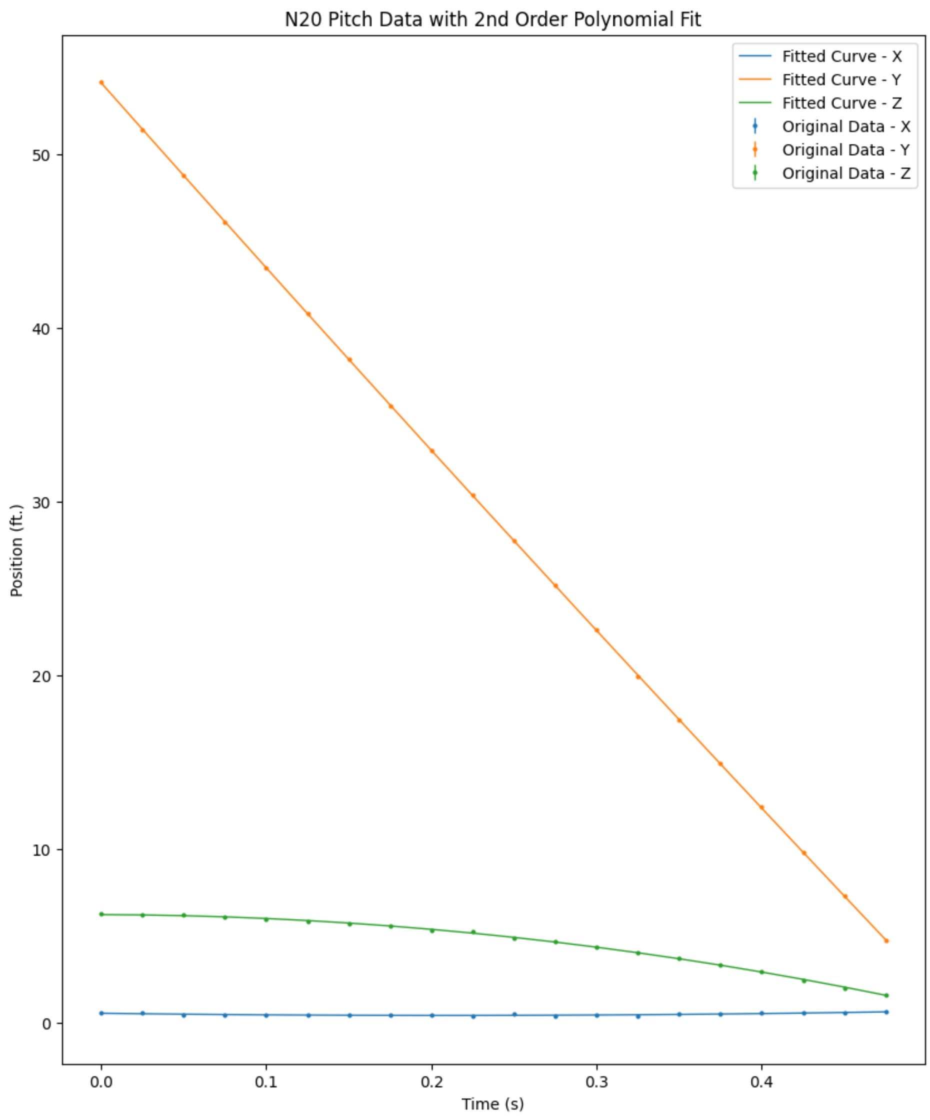

Figure 2: 2nd order polynomial fits for the 20 point dataset

1.2 N100

These are the fit equations for the N100 dataset.

$$x = 0.4788 + -0.8376t + 2.1905t^2 \tag{5}$$

$$y = 54.1259 + -107.3051t + 6.9758t^2 \tag{6}$$

$$z = 6.2077 + -0.4104t + -19.7456t^2 \tag{7}$$

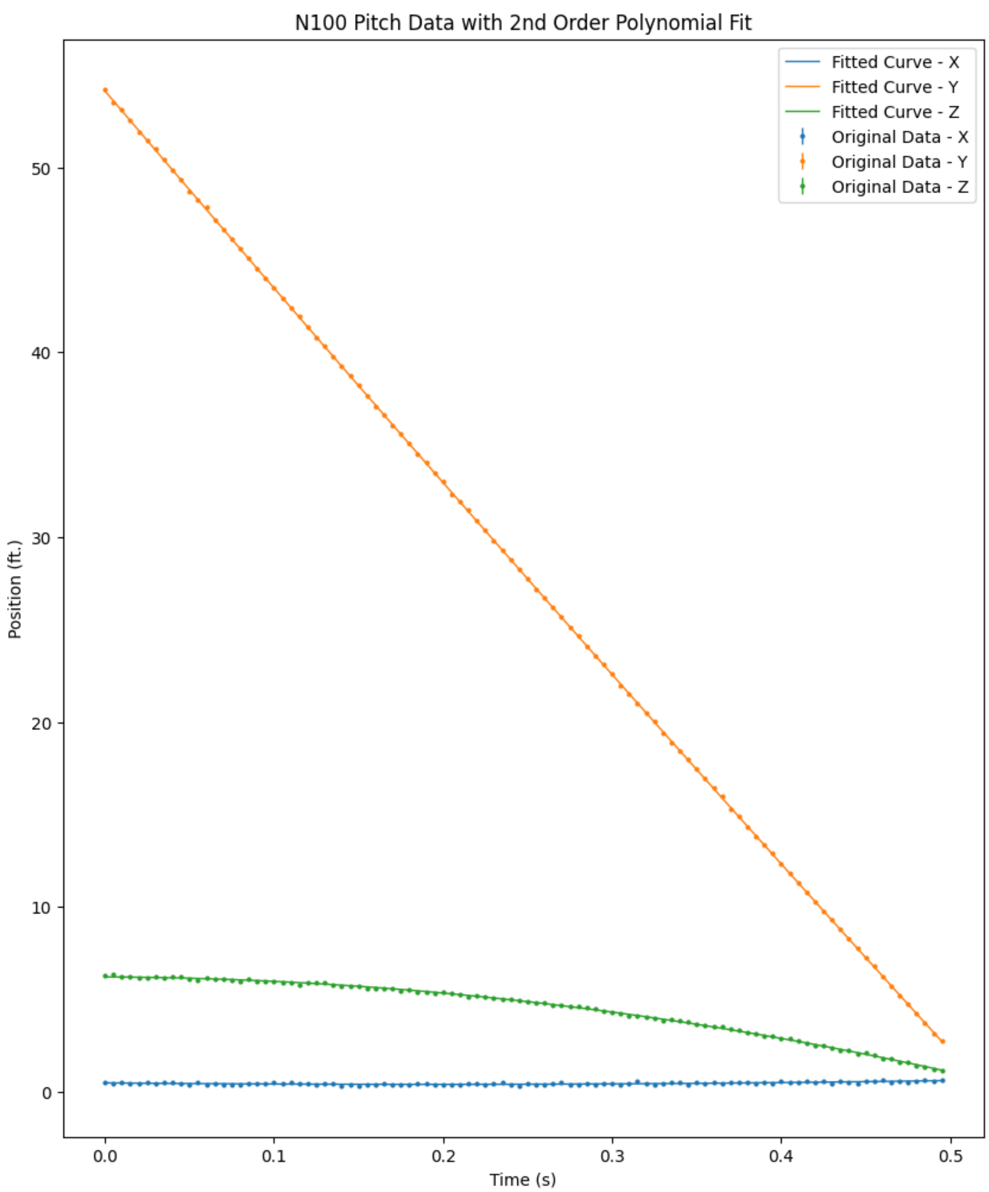

Figure 3: 2nd order polynomial fits for the 100 point dataset

1.3 N300

These are the fit equations for the N300 dataset.

$$x = 0.4918 + -0.9208t + 2.3560t^2 \tag{8}$$

$$y = 54.1461 + -107.3712t + 6.9547t^2 \tag{9}$$

$$z = 6.2033 + -0.2718t + -20.0696t^2 \tag{10}$$

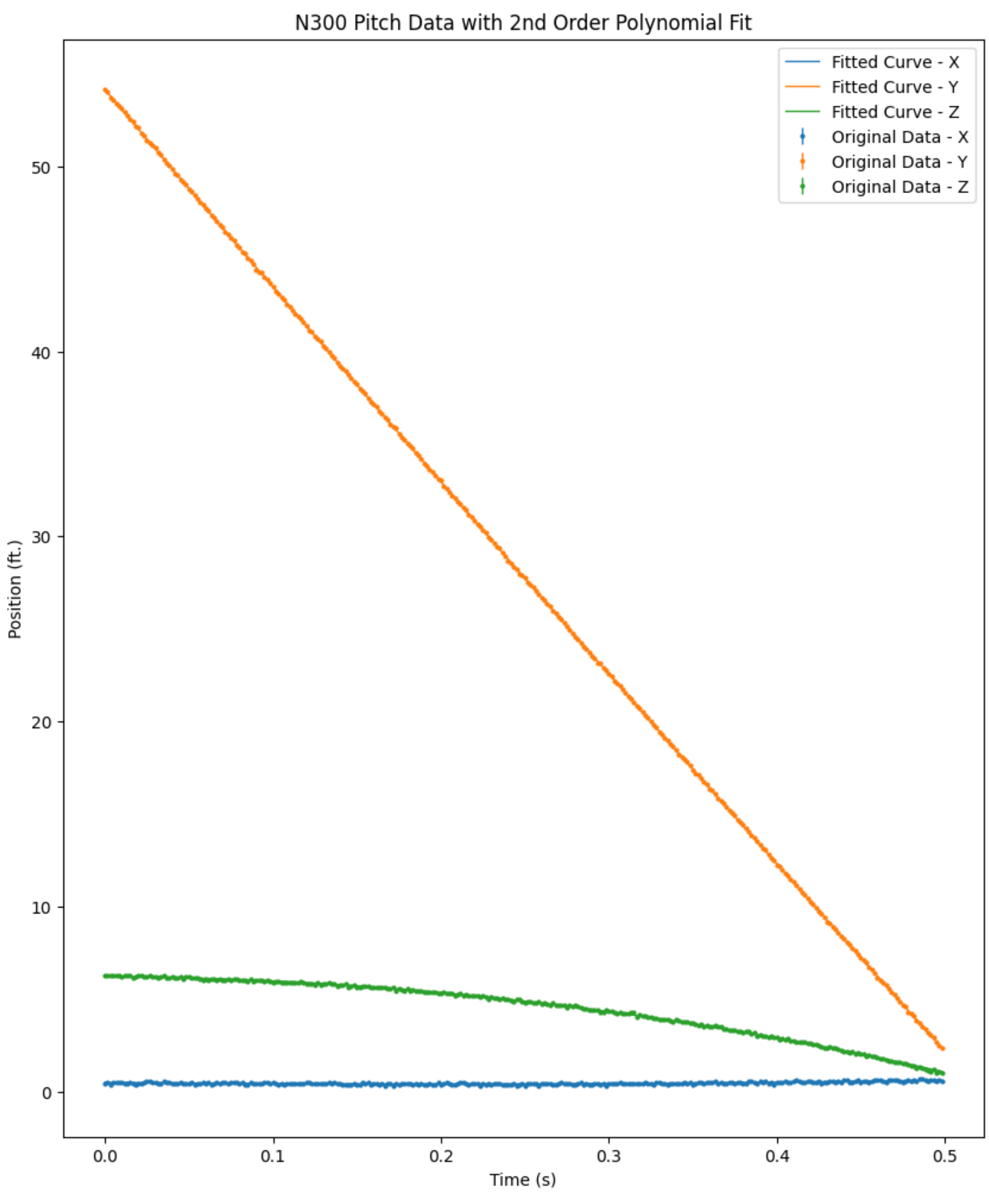

Figure 4: 2nd order polynomial fits for the 300 point dataset

1.4 N1000

These are the fit equations for the N1000 dataset.

$$x = 0.4937 + -0.9674t + 2.4283t^2 \tag{11}$$

$$y = 54.1324 + -107.2821t + 6.8334t^2 \tag{12}$$

$$z = 6.1994 + -0.2281t + -20.1971t^2 \tag{13}$$

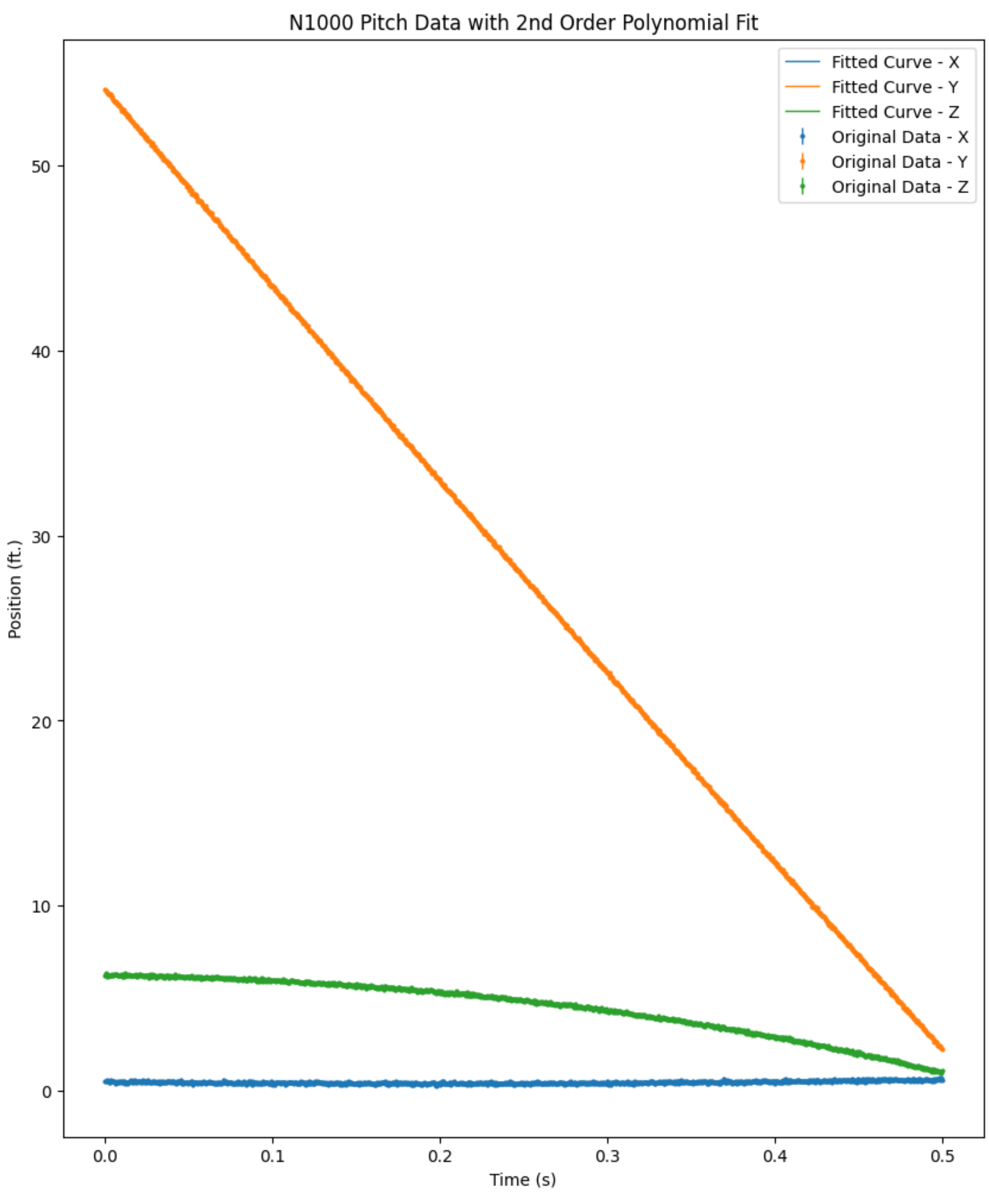

Figure 5: 2nd order polynomial fits for the 1000 point dataset

2 Uncertainty of the Fit Parameters

The uncertainties of the fit parameters are determined by first building a 3x3 α matrix.

$$\alpha_{l,k} = \sum_{i=1}^{N} [\frac{1}{u_{y_i}^2} x_i^{l-1} x_i^{k-1}] \tag{14}$$

The ϵ matrix is then calculated by doing the inverse of α.

$$\epsilon = \alpha^{-1} \tag{15}$$

The elements on the diagonal of ϵ are the squares of the uncertainties of the coefficients. The relative uncertainties are then calculated for each dataset by dividing the uncertainties by the coefficients for each fit.

Last, the residuals are calculated using eq. 16 and compared to the actual total uncertainty, $u^2 = 0.001736$.

$$r_y^2 = s_y^2 \approx \frac{1}{N-3} \sum_{i=1}^{N} (y_i - A_1 - A_2 x_i - A_3 x_i^2)^2 \tag{16}$$

```
N=20
Uncertainties: u_A1 = 0.0254, u_A2 = 0.2476, u_A3 = 0.5031

Relative Uncertainties:
x: u_A1/A1 = 4.8529%, u_A2/A2 = 20.6773%, u_A3/A3 = 17.4786%
y: u_A1/A1 = 0.0469%, u_A2/A2 = 0.2306%, u_A3/A3 = 7.0352%
z: u_A1/A1 = 0.4089%, u_A2/A2 = 123.4974%, u_A3/A3 = 2.4939%

s^2 values:
x: s^2 = 0.0014
y: s^2 = 0.0012
z: s^2 = 0.0022

N=100
Uncertainties: u_A1 = 0.0123, u_A2 = 0.1144, u_A3 = 0.2237

Relative Uncertainties:
x: u_A1/A1 = 2.5593%, u_A2/A2 = 13.6599%, u_A3/A3 = 10.2105%
y: u_A1/A1 = 0.0226%, u_A2/A2 = 0.1066%, u_A3/A3 = 3.2063%
z: u_A1/A1 = 0.1974%, u_A2/A2 = 27.8788%, u_A3/A3 = 1.1327%

s^2 values:
x: s^2 = 0.0019
y: s^2 = 0.0017
z: s^2 = 0.0021

N=300
Uncertainties: u_A1 = 0.0072, u_A2 = 0.0665, u_A3 = 0.1291

Relative Uncertainties:
x: u_A1/A1 = 1.4578%, u_A2/A2 = 7.2173%, u_A3/A3 = 5.4797%
y: u_A1/A1 = 0.0132%, u_A2/A2 = 0.0619%, u_A3/A3 = 1.8563%
z: u_A1/A1 = 0.1156%, u_A2/A2 = 24.4487%, u_A3/A3 = 0.6433%

s^2 values:
```

```
x: s^2 = 0.0019
y: s^2 = 0.0014
z: s^2 = 0.0024

N=1000
Uncertainties: u_A1 = 0.0039, u_A2 = 0.0365, u_A3 = 0.0707

Relative Uncertainties:
x: u_A1/A1 = 0.7991%, u_A2/A2 = 3.7711%, u_A3/A3 = 2.9119%
y: u_A1/A1 = 0.0073%, u_A2/A2 = 0.0340%, u_A3/A3 = 1.0348%
z: u_A1/A1 = 0.0636%, u_A2/A2 = 15.9936%, u_A3/A3 = 0.3501%

s^2 values:
x: s^2 = 0.0017
y: s^2 = 0.0018
z: s^2 = 0.0023
```

As seen from the numbers above, the s^2 calculated values for x and y are very close to the value of 0.001736 that is expected. For z, s^2 is slightly higher than expected, but still within a reasonable range. The relative uncertainty is relatively good for all coefficients except some values of A_2 which is caused by a small value of A_2 rather than a high value of the uncertainty.

The uncertainty of the coefficients is plotted on a log log plot vs the number of data points in the dataset. As seen in the plot, the uncertainty of the coefficients decreases following a linear path on the log-log plot as the size of the dataset increases.

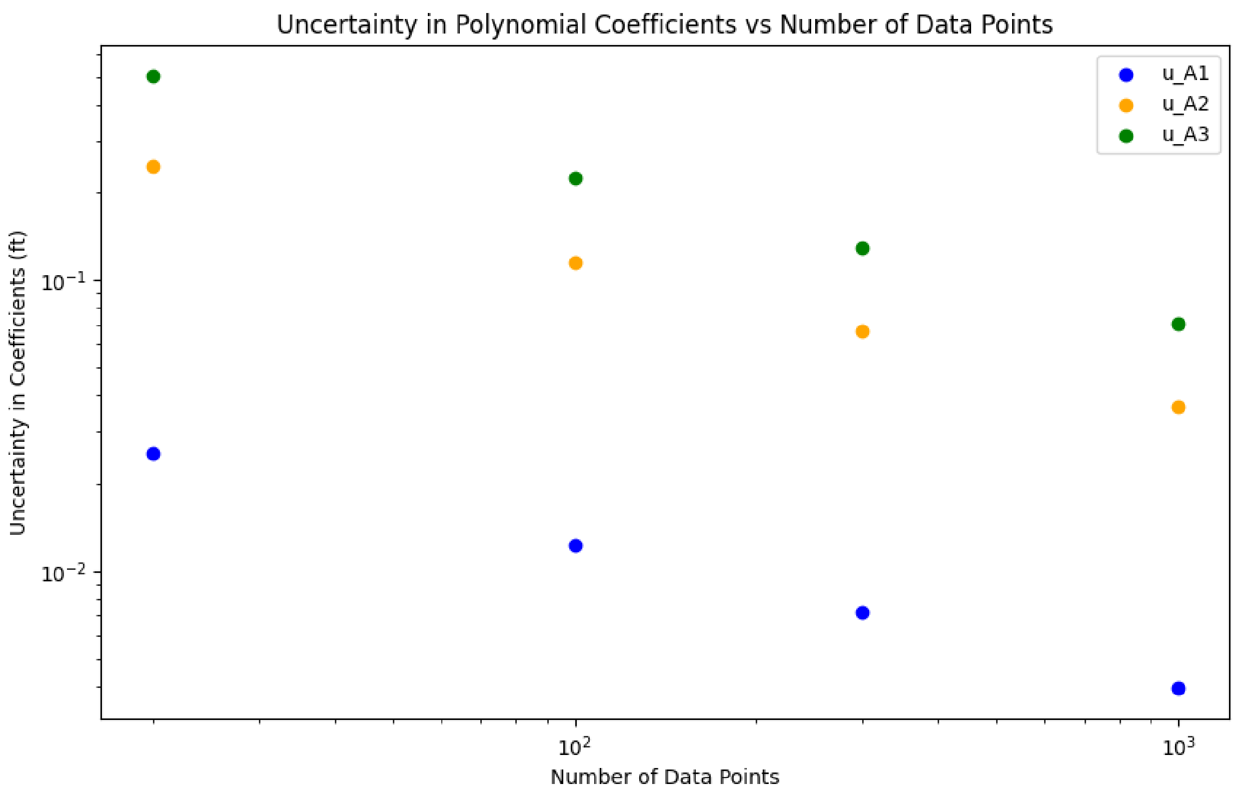

Figure 6: Log-Log plot of the uncertainty of the polynomial fit coefficients vs the number of points in the dataset

3 Acceleration in Each Direction

The acceleration in each direction is the A_3 coefficient for each fit. For a freefalling object with no spin and no wind resistance, the acceleration in the z direction would be $g = -32.2\frac{ft}{s^2}$ and all other accelerations, A_3 coefficients, would be 0. The accelerations for each direction for each case are shown in table 1 below.

N	20	100	300	1,000
a_x	2.8787	2.1905	2.3560	2.4283
a_y	7.1518	6.9758	6.9547	6.8334
a_z	-20.1755	-19.7456	-20.0796	-20.1971

Table 1: Accelerations in each direction for each case

I also plotted the movement of the ball towards home plate using the fit from N1000. The green dashed line is the difference between the fitted curve and a curve that follows the acceleration due to gravity, which is shown in red. This shows that the ball is experiencing lift due to the backspin of the ball.

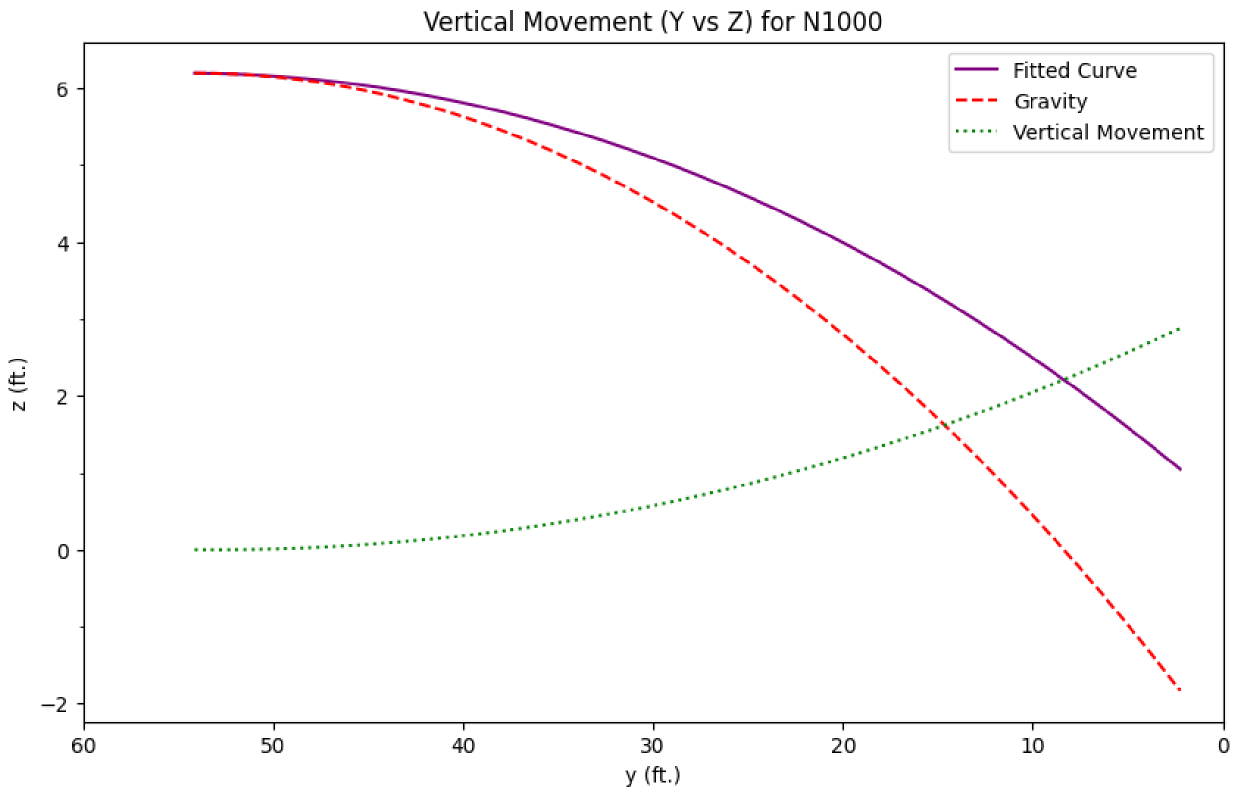

Figure 7: Plot of the z and y fits compared to the ball drop expected due to gravity

4 What Type of Pitch is it?

To determine the type of pitch, I looked at three parameters: vertical movement, horizontal movement, and velocity. Looking at the vertical movement chart in Fig. 7, the pitch rises, which means it is thrown with backspin. Looking at the horizontal movement in Fig 8, the pitch breaks away from a right handed batter. Additionally, I calculated the velocity at time $t = 0$ using eq. 17. The initial velocity of the pitch is 73.15 mph (107.29 ft/s).

These three observations suggest that the pitch is a cutter.

$$v_{mph} = \sqrt{A_{2,x}^2 + A_{2,y}^2 + A_{2,z}^2} * \frac{3600}{5280} \tag{17}$$

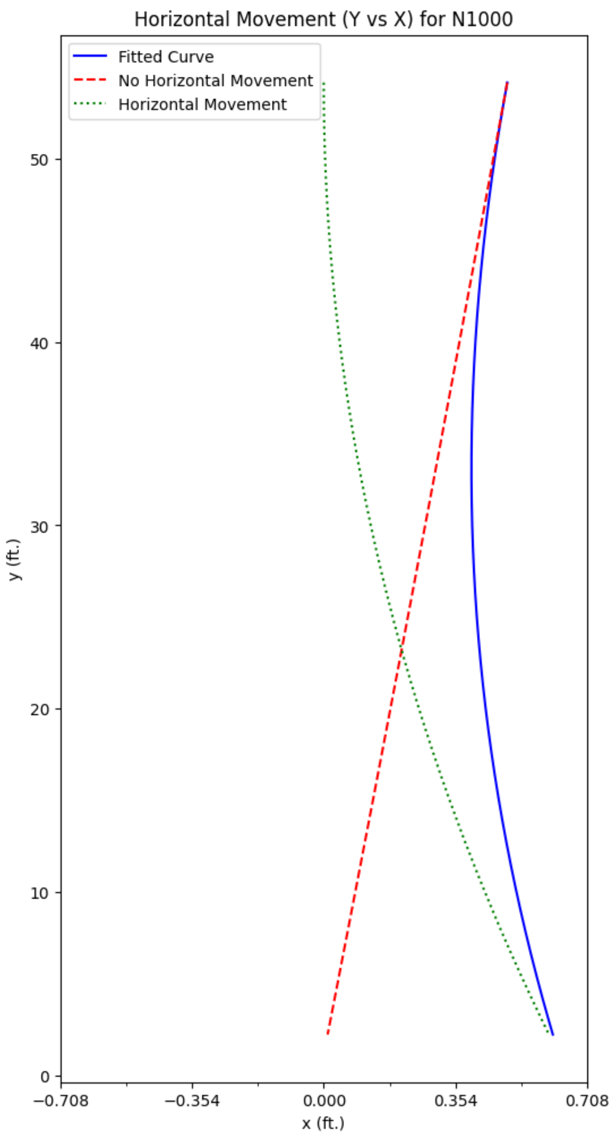

Figure 8: Horizontal movement of the ball from a top down perspective