Bouncing Ball Error Analysis

M. Klimuszka
4223034

H14POD Advanced Engineering Research Project Organisation and Design
15 April 2015

Introduction

The interaction between a bouncing ball and the ground can be characterized by the coefficient of restitution, C_R. This describes the relationship between the height a ball is dropped from and the height it bounces to and is defined in equation 1. This report measures C_R for a golf ball using a variety of methods using video and audio capture. Particular attention is paid to the experimental error for each method. From each of three data sets, C_R is calculated using two different mathematical methods. Corrections for parallax are also introduced in an attempt to increase the measurement accuracy. The bouncing ball is also simulated in Microsoft Excel and MATLAB in order to compare the experimental data to simulation. In these comparisons, the long distance visual technique proved to be the most similar to the simulation once it had been corrected for parallax. This method resulted in the C_R measurement of 0.902 ± 0.003.

$$C_R=\sqrt{\frac{H_{i+1}}{H_i}}$$

Equation 1: Coefficient of Restitution definition. i ***is bounce number (or peak number).***

The general motion of a bouncing ball is depicted in Figure 1. The time of each bounce and the heights of the drop and each peak were measured. Three techniques were used. The first method involved recording a bouncing ball using a phone camera that was placed 250 cm away, this method is referred to as "far shot." For the second method, the camera was moved as close to the experiment as possible, referred to as "close shot." This was done to increase the detail of the recording and reduce parallax errors by placing the camera at a vertical position close to what was expected for the peak height. The last method was based on the audio portion of the recordings. The audio was opened in Audacity and the times of the bounces were measured. Having measured the time intervals of each bounce, the heights of the peaks could be calculated.

Experimental Procedure

Equipment List

- Measuring Tape
- Golf Ball
- Packaging Tape
- Marker
- Camera with Audio

The experimental setup was simple. A tape measure was attached to the wall to measure from the ground up to a height of 200 cm. Parallel to the tape measure, dark marks were made on packaging tape every 10 cm to make measurements easier to read on the video recording. The ball was dropped by hand from a height of 200 cm while recording using the methods described above. Ten trials were recorded at the far distance, but when the videos were analyzed, measurements for every bounce were not readable due to the ball occasionally bouncing out of frame. Then, the close shots were taken, but some of them also turned out to be unusable.

In order to take measurements from the video files, they were opened in VLC media player and the correct frames were chosen and copied into a suitable image editor to take detailed measurements. An example of this process is shown in Figure 2. Retrieving time data from the audio portion of the files was done by importing the videos into Audacity. After removing background noise, a profile similar to Figure 3 remains. From there, the time stamp of each bounce can be read, and the peak heights calculated using basic mechanics. The measurements for each height were averaged for each of the three methods. All measurement errors were propagated using standard techniques (Equation 2).

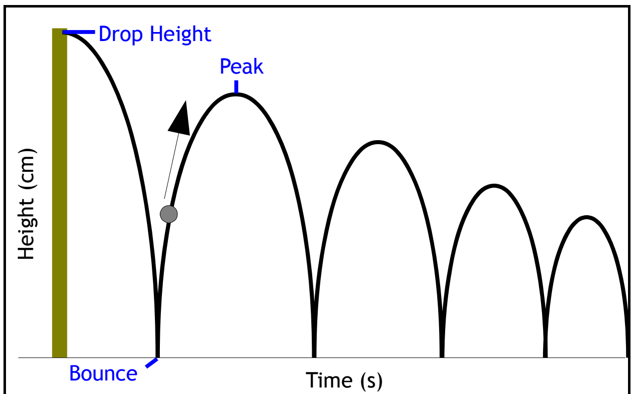


Figure 1: The general movement of a bouncing ball, along with the labeled values that were measured.

$$\delta f(x,y,\ldots)=\sqrt{\left(\frac{\partial f}{\partial x}\right)^2(\delta x)^2+\left(\frac{\partial f}{\partial y}\right)^2(\delta y)^2+\ldots}$$

Equation 2: Where δf is the uncertainty in the value of the function f.

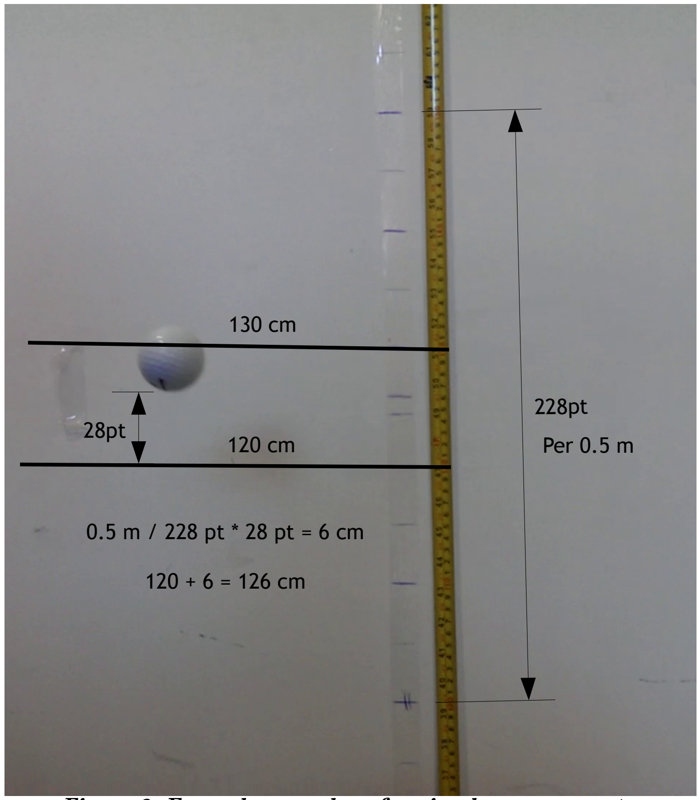


Figure 2: Example procedure for visual measurements.

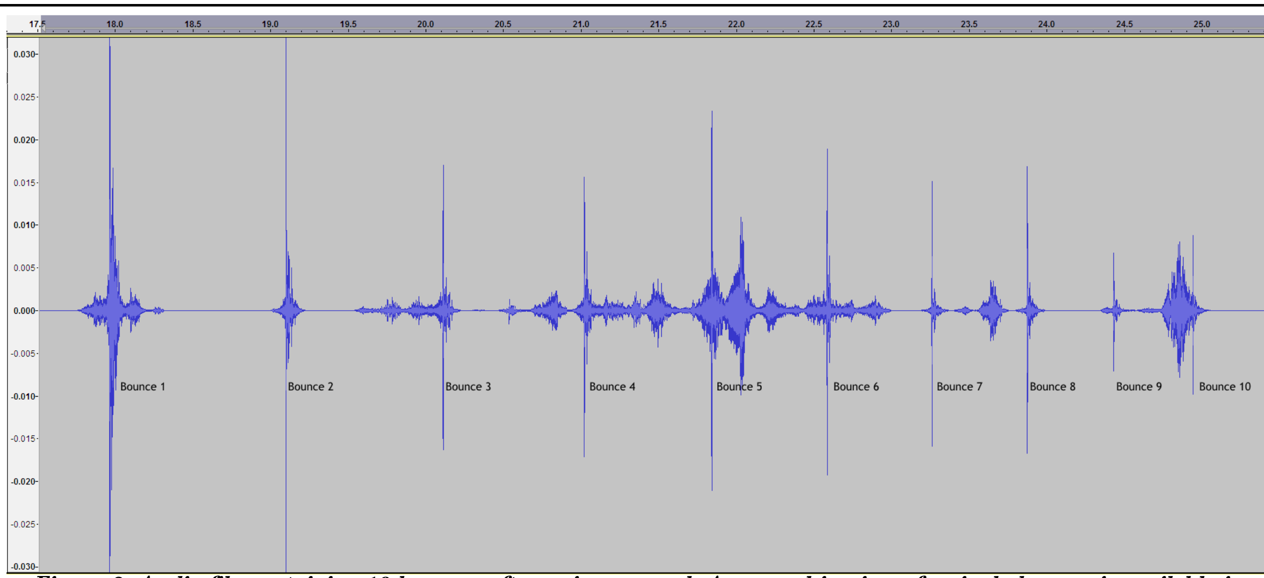


Figure 3: Audio file containing 10 bounces after noise removal. A zoomed in view of a single bounce is available in Appendix 1.

Calculating C_R was done using two different methods. The most straightforward method is to use definition of C_R (Equation 1) and calculate it using the height of two adjacent peaks. The other method is to perform regression analysis on the data, which requires it to be transformed as follows since Microsoft Excel has limited data fitting capabilities.

$$C_R = \sqrt{\frac{H_{i+1}}{H_i}}$$
$$H_i C_R^2 = H_{i+1}$$

$$H_1 = H_0 C_R^2$$
$$H_2 = H_1 C_R^2 = H_0 C_R^{(2\cdot 2)}$$
$$\vdots$$
$$H_i = C_R^{2i} H_0$$
$$\frac{H_i}{H_0} = C_R^{2i}$$

$$\ln\left(\frac{H_i}{H_0}\right) = 2i \ln(C_R)$$

Plotting $\ln\left(\frac{H_i}{H_0}\right)$ vs. bounce number, i will result in a linear plot with a slope of $2\ln(C_R)$.

$$C_R = e^{slope/2}$$
$$\delta C_R = \frac{C_R}{2} \delta slope$$

From here, Excel's LINEST function can be used to obtain the slope, with uncertainty, and then calculate C_R. Unfortunately, LINEST calculates uncertainty based solely on data scatter and cannot take uncertainty in the data points into consideration. The results of every C_R calculation are listed in Table 1.

	Close Camera				Far Camera				Sound	
	Raw		Parallax Corrected		Raw		Parallax Corrected			
	2 Point	Linear Fit	2 Point	Linear Fit	2 Point	Linear Fit	2 Point	Linear Fit	2 Point	Linear Fit
C_R	0.90	0.900	0.904	0.906	0.89	0.890	0.902	0.901	0.904	0.904
δC_R	0.01	0.002	0.002	0.002	0.02	0.003	0.003	0.001	0.001	0.001

Table 1: C_R calculations from various methods with uncertainty.

Error Analysis

Drop Height Error

The ball was manually dropped from a height of 200cm. Being done by a human, there is random error estimated to be ± 3 cm. This could be reduced by using a mechanical method to drop the ball from a consistent height. A properly designed mechanism would essentially reduce this error to zero.

Height Uncertainty Due to Frame Rate.

It is very clear which frame is closest to the peak of the bounce, since it slows to a stop. This gives a $\pm$ half frame random uncertainty in the visual measurement. Since the ball is at rest at the peak height, and the frame rate of the camera is constant (27 fps), the possible error in height is 0.17 cm. This could be decreased by using a camera with a higher frame rate.

$$y = y_0 + v_0 t + \frac{1}{2} a t^2$$
$$y = (0)\left(\frac{1}{27}\frac{1}{2}\right) + \frac{1}{2}(980)\left(\frac{1}{27}\frac{1}{2}\right)^2$$
$$y = 0.17\ cm$$

Error in Measuring Height from Screen Shots

In addition to the uncertainty in capturing the correct moment on film as described previously, there is an uncertainty in measuring heights from the screen shots. This is due to the blurriness of the ball and how zoomed out the camera is. The size of this error is 1-3 cm depending on the zoom level and adds to the frame rate error. The allocation of this error can be seen in Appendix 2B and an example of one measuring technique is shown in Figure 2.

Parallax Error

Since the video camera was not perfectly level with the ball at the time it was measured, and the ball does not stay flush to the wall, a parallax error occurs. This causes the measurements to be high when the ball is above the camera, and the measurements to read low when it is below the camera. . Figure 4 illustrates how this error (H_Δ) occurs. This error can be reduced by placing the camera at the height of the expected measurement, which was attempted in the close shots. Other ways to reduce parallax error would be to increase the camera distance from the wall, which would require a higher quality camera, or to stop the ball from bouncing away from the wall, which would require a flatter floor.

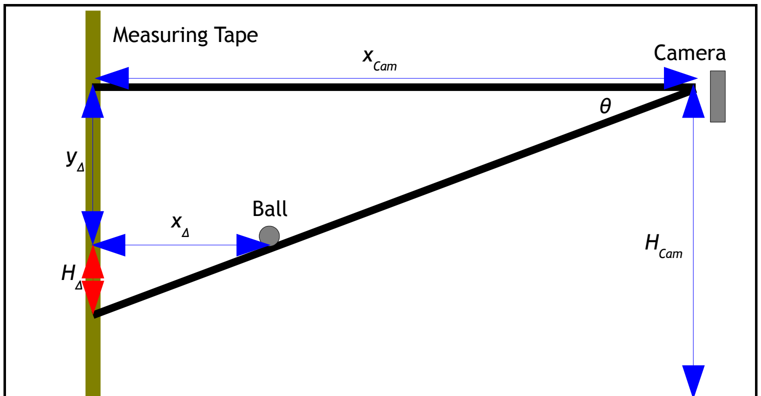


Figure 4: An illustration of parallax error, not drawn to scale, with the required measurements labeled.

Parallax error can also be mathematically corrected afterwards, which was done and is the difference between far/close raw and corrected data. The procedure for applying these corrections requires knowledge of the positions of the ball and camera for each measurement, as seen in Figure 4. The process for calculating these corrections is described here. First, y_Δ was determined using the simulated ball height and the camera height. Then, the angle can be calculated.

$$\theta = \arctan\left(\frac{y_\Delta + H_\Delta}{x_{Cam}}\right) \approx \arctan\left(\frac{y_\Delta}{x_{Cam}}\right)$$

This approximation assumes $H_\Delta << y_\Delta$,which generally holds true for when the camera is in the far position. This assumption is less valid for the close camera position, and it is evident that these corrections are less accurate when comparing the corrected close data with the simulation as seen in Figure 7. Being able to calculate the angle, the parallax error, H_Δ, can be calculated.

$$H_\Delta = \tan(\theta)\, x_\Delta = \frac{y_\Delta}{x_{Cam}} x_\Delta$$

x_Δ was determined by an estimated rate of movement, $x_\Delta = 3 + \frac{37}{5} t$

For the close shots, the camera distance, x_{Cam} was measured for one distance and other distances were calculated based on the amount of measuring tape that could be seen in the frame, assuming a linear relationship between that and distance. The far shot value for x_{Cam} was approximately 250 cm.

The size of this parallax error ranged between 0 and 8 cm depending on the set-up, calculations can be found in Appendix 2E.

Errors in Audio Measurements

Initially, the audio based method was expected to be the most accurate measuring technique. Bounce waveforms can be easily spotted in Audacity and timed with a precision of ± 0.003 seconds, including human judgment errors. But it is only possible to measure the times of bounces in this way, not the exact time of dropping the ball. Two methods were used to determine the drop time. One includes the person dropping the ball saying "drop" which takes 0.3 seconds to say. This gives an uncertainty of at least 0.15 seconds, and 0.3 seconds was used for a conservative estimate. The other method uses the video to watch the experimenter drop the ball. This can be done with a maximum accuracy of ± 1 frame, or 0.04 seconds. Even with these methods, the time for the first bounce was measured much lower than what is reasonable, so initial heights/times as measured by audio recordings are left out of the analysis. The time intervals between bounces, however, can be measured to a very high precision.

The accuracy of these measurements, unfortunately, comes into question. When determining C_R from these measurements using the two point method, C_R is calculated to steadily increase after each bounce (Appendix 2C), as opposed to the random fluctuations as seen in the camera measurements (Appendix 2B). This indicates that there is a systematic error with an unidentified source. A technological systematic error is also indicated in the defined curve that this data's simulation comparison makes in Figure 7. The most likely cause for this error is a mismatch between recording rate and playback speed. Some preliminary experiments show that the camera utilizes a variable frame rate, which can introduce timing errors. Even small fractions of a second can make a large difference when converting time intervals to peak heights. The C_R values found using this method should not be trusted. Their measurements were also used to find the total time, but the human error, which includes the initial drop time seems to be larger than this systematic error for that purpose.

Simulations and Results

Microsoft Excel

To perform the spreadsheet simulation, all of the peak heights were found using the definition of C_R with H_0 = 200 cm and C_R = 0.900. This value of C_R is the mean of the ten measurements of this value in this report. For each height, the time to fall from the peak to the ground was calculated.

$$y = y_0 + v_0 t + \frac{1}{2} a t^2$$

$$0 = H_i + (0) t + \frac{1}{2}(-980) t^2$$

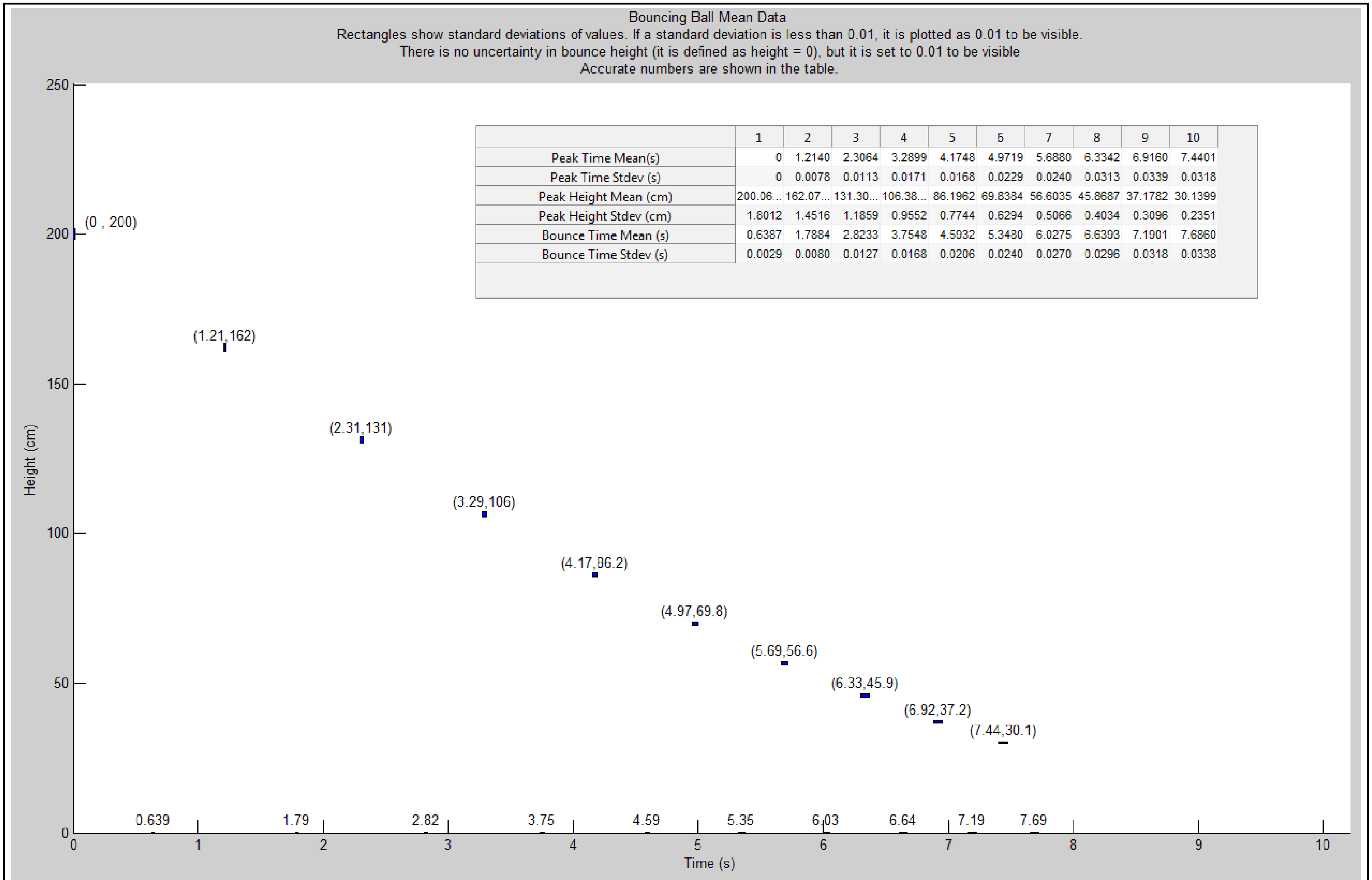
$$t = \sqrt{\frac{2 \cdot H_i}{980}}$$

From there, it is trivial to add these half bounce times together to find the total time elapsed at each bounce, which turned out to be 7.68 seconds for C_R = 0.900. This is outside the bounds defined by the audio measurement of 7.48 ± 0.07 cm. But, the uncertainty in that measurement assumed the drop time error was random. If it was not random and timed consistently late, then the real values agree much better than is indicated here.

MATLAB

The MATLAB simulation used basic mechanics equations to simulate the incremental motion of a bouncing ball. When the ball moved below ground level, the ball was reset to y = 0 cm, and the appropriate time for this position was calculated. From there, the velocity was reversed and adjusted according to C_R. This method gives accurate values since the bounce position and velocities are corrected, not just reversed; it plots nicely due to the small time increments; and it produces a smooth and real time .avi animation because nearly every frame is the exact same length. The user may specify the number of times the simulation runs, and allows random error to be added to the drop height. The result of 15 trials with ± 3 cm random error is plotted in Figure 5.

The simulations produced in MATLAB and Microsoft Excel agree with each other perfectly, which is expected since they are both based on the same equations, although the process of calculation is different for each. The results of the simulation are plotted with the various height measurements in Figure 6. Figure 7 shows the difference between the measurements and the simulation. A more detailed comparison between the various height and time measurements and the simulation can be found in Appendix 2D.



	1	2	3	4	5	6	7	8	9	10
Peak Time Mean(s)	0	1.2140	2.3064	3.2899	4.1748	4.9719	5.6880	6.3342	6.9160	7.4401
Peak Time Stdev (s)	0	0.0078	0.0113	0.0171	0.0168	0.0229	0.0240	0.0313	0.0339	0.0318
Peak Height Mean (cm)	200.06...	162.07...	131.30...	106.38...	86.1962	69.8384	56.6035	45.8687	37.1782	30.1399
Peak Height Stdev (cm)	1.8012	1.4516	1.1859	0.9552	0.7744	0.6294	0.5066	0.4034	0.3096	0.2351
Bounce Time Mean (s)	0.6387	1.7884	2.8233	3.7548	4.5932	5.3480	6.0275	6.6393	7.1901	7.6860
Bounce Time Stdev (s)	0.0029	0.0080	0.0127	0.0168	0.0206	0.0240	0.0270	0.0296	0.0318	0.0338

Figure 5: The mean and standard deviation of simulated data from 15 MATLAB trials with random error of + 3 cm. See subtitle of plot for more information.

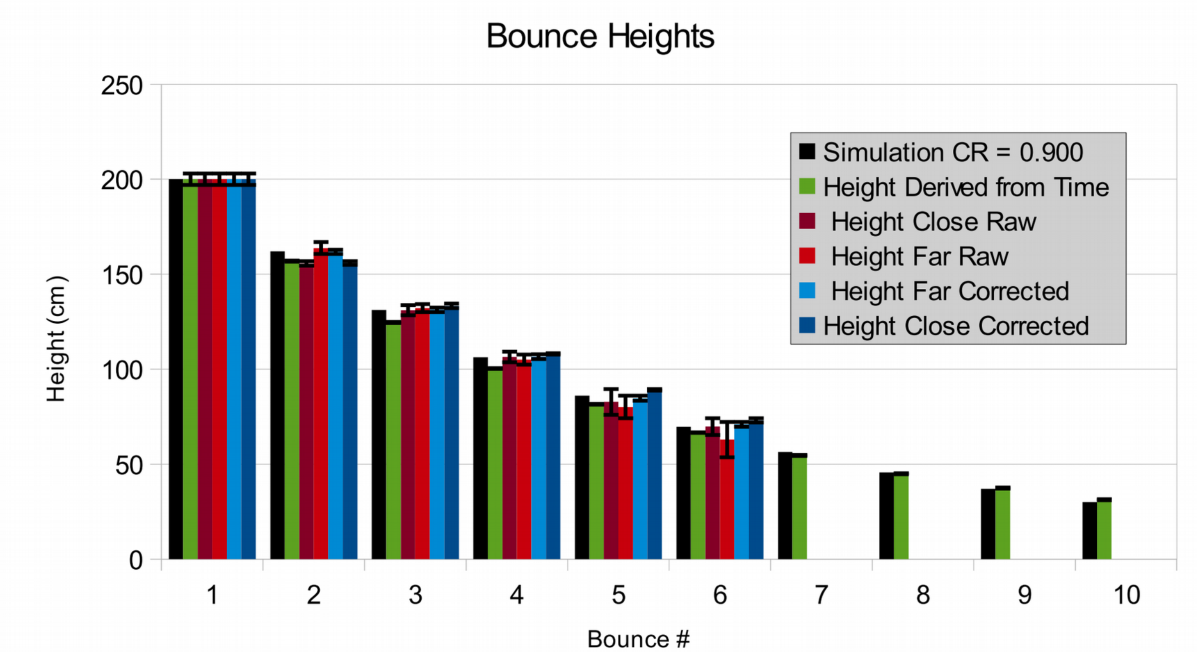


Figure 6: Bounce heights from every measurement method. "Corrected" data was corrected for parallax effects, details can be found in Appendix 2E. The first height is the same for all methods. It is assumed to be 200 ± 3 cm. "Height Derived from Time" refers to audio measurements.

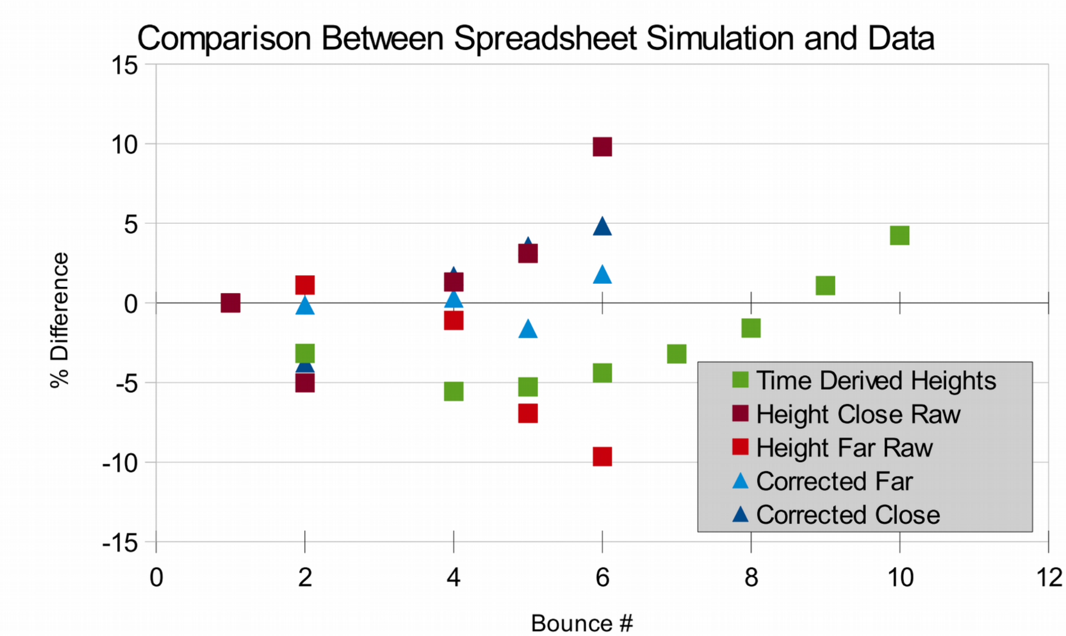


Figure 7: Percent difference between the measured heights and the simulation heights with C_R = 0.900. "Corrected" data was corrected for parallax effects, details can be found in Appendix 2E. "Time Derived Heights" refer to audio measurements.

Conclusion

The motion of a bouncing ball was both measured and simulated. Various methods were used to determine the coefficient of restitution, C_R,and the most accurate is the far shot method with corrections for parallax error. Both the far and close visual methods suffered from parallax error as their largest uncertainty, but the stationary far camera and small relative size of H_Δ to y_Δ allowed the parallax error to be corrected. This resulted in superior agreement with the simulation, remaining under a 2% difference for all 6 visually measured bounces(Figure 7). The audio technique was expected to be the most accurate measurement because the high precision that is possible when measuring bounce times. In the end, the initial drop time could be measured no more accurately than ± 0.15 seconds in a purely sound based process, and an unknown systematic error plagued the results. It did, however, disagree with the simulation by no more than 6% and was able to measure more bounces than the visual methods since it does not require the ball to stay in frame.

The coefficient of restitution was found to be 0.902 ± 0.003 from the simple two point method with the far shot parallax corrected data. The two point method gives a more accurate error than linear regression, since it takes into consideration measurement error. The linear regression method gives $C_R = 0.901 \pm 0.001$. This might give a more accurate value, but it overlaps with the range from the two point method, so choosing this over the other method has very little advantage.

Appendix 1

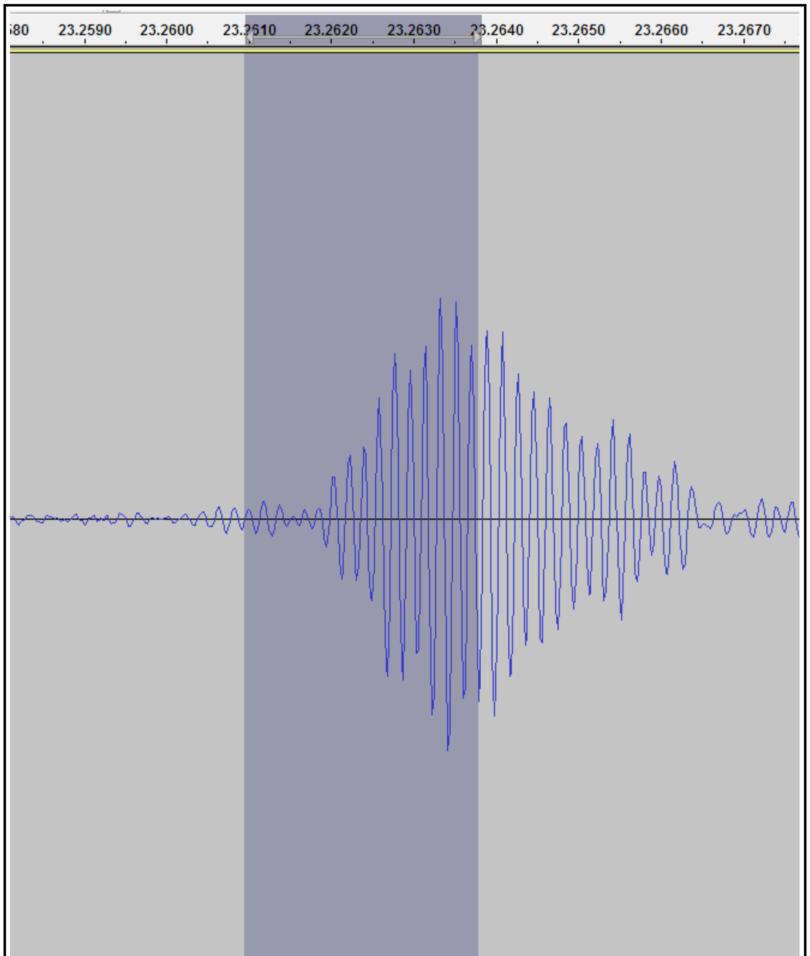


Figure 8: The sound of one bounce.

Appendix 2

See spreadsheet for detailed calculations. The different sheets in the file are referred to in this document as lettered sub appendices.

Appendix 2A - "Summary" Sheet

Appendix 2B - "Heights" Sheet

Appendix 2C - "Audio" Sheet

Appendix 2D - "Simulation" Sheet

Appendix 2E - "Parallax Error" Sheet

Appendix 3

See submitted video file from the MATLAB simulation. A screen shot is here.

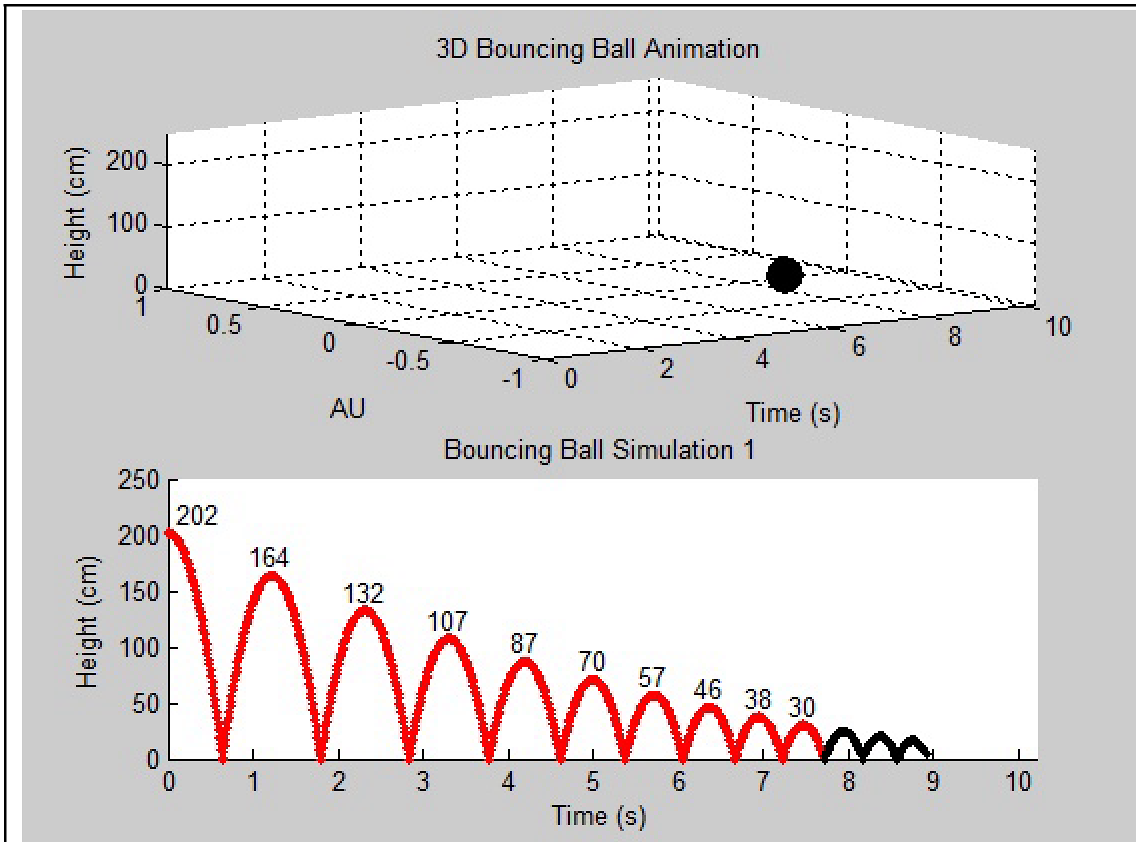


Figure 9: A screenshot from the submitted .avi file. Peak heights are labeled, and a random error of 2cm was added to the specified drop height (200 cm).

Works Referenced

The only works referenced were pages from the MATLAB documentation files for assistance in writing the simulation and the original assignment sheet for the definition of coefficient of restitution.