

Laboratory 2: Coefficient of restitution, linear regression**Needed:**

1. Rubber ball
2. Meter stick
3. Stopwatch (online)
4. Group of 2-3 students

I Introduction

The coefficient of restitution ϵ is a measure of how much energy is lost in a real collision. Specifically, it is the ratio of the speed of separation after a collision to the speed of approach before a collision:

$$\epsilon = \frac{v_{\text{after}}}{v_{\text{before}}} \quad \text{or} \quad v_{\text{after}} = \epsilon v_{\text{before}} \quad (1)$$

For a collision with perfect conservation of mechanical energy (an “elastic collision”), $\epsilon = 1$. The coefficient of restitution is a convenient way of specifying how much energy in a collision is transferred to internal energy. In general it will depend on the specific materials which are colliding, but under experimentally realizable conditions, it is a constant factor representing the energy transfer from kinetic energy to internal energy. A value of $\epsilon = 0$ implies a perfectly inelastic collision – a ball that does not bounce. A value of, say, $\epsilon = 0.5$ means that the ball rebounds from the floor with half the speed at which it struck the floor.

I.1 Height of a bouncing ball after n bounces

In order to illustrate the utility of the coefficient of restitution, consider a ball of mass m dropped from rest at a height h_0 . When it reaches the ground, it will have a speed v_0 . Neglecting air resistance, conservation of mechanical energy dictates that the starting gravitational potential energy mgh_0 must be equal to the kinetic energy just before impact:

$$mgh_0 = \frac{1}{2}mv_0^2 \quad (2)$$

yielding $v_0 = \sqrt{2gh_0}$. Here, $g = 9.81 \text{ m/s}^2$ is the gravitational acceleration. This result tells us that if we know the starting height that the ball is dropped from, we can predict its speed just before it hits the ground. The coefficient of restitution ϵ defined above gives us a way to predict the speed after the ball just *after* rebounding. If we call the velocity after a single bounce v_1 , we determine $v_1 = \epsilon v_0 = \epsilon\sqrt{2gh_0}$. We can now run our conservation of energy in reverse to find the maximum height after one rebound:

$$h_1 = \frac{v_1^2}{2g} = \frac{\epsilon^2(2gh_0)}{2g} = \epsilon^2 h_0 \quad \text{height after 1 bounce}$$
$$\epsilon = \frac{v_{\text{after}}}{v_{\text{before}}} = \sqrt{\frac{h_1}{h_0}}$$

One can repeat the same analysis, and show that the height after the second rebound is $h_2 = \epsilon^4 h_0$, and in general after n bounces the rebound height is

$$h_n = \epsilon^{2n} h_0 \quad \text{height after } n \text{ bounces} \quad (3)$$

Measuring the height after n rebounds should allow us to extract a value for ϵ . What is curious is that h_n does not reach zero for finite n , i.e., in principle the ball bounces an infinite number of times. (This is version of “Zeno’s paradox.”)

1.2 How long does it take to stop bouncing?

We can also calculate the time it takes for each bounce and rebound. For the initial drop from height h_0 , the time is $t_0 = \sqrt{\frac{2h_0}{g}}$, and each subsequent rebound and drop is calculated in the same way (counting twice for the round trip). For example, the time for the first two bounces is:

$$t_1 = 2\sqrt{\frac{2h_1}{g}} = 2\epsilon\sqrt{\frac{2h_0}{g}} = 2\epsilon t_0 \quad t_2 = 2\sqrt{\frac{2h_2}{g}} = 2\epsilon^2\sqrt{\frac{2h_0}{g}} = 2\epsilon^2 t_0 \quad (4)$$

Continuing this way, we can sum the time for each of the infinite number of bounces, noting that the first times above lead us to the pattern $t_n = 2\epsilon^n t_0$:

$$\begin{aligned}t_{\text{total}} &= t_0 + t_1 + t_2 + \dots \\&= t_0 [1 + 2\epsilon + 2\epsilon^2 + 2\epsilon^3 + \dots] \\&= t_0 [2 + 2\epsilon + 2\epsilon^2 + 2\epsilon^3 + \dots - 1] \\&= t_0 \left[\sum_{n=0}^{\infty} 2\epsilon^n \right] - 1 = t_0 \left[\frac{2}{1-\epsilon} - 1 \right] = t_0 \left[\frac{1+\epsilon}{1-\epsilon} \right]\end{aligned}$$

The sum converges, and we find that the time it takes to stop bouncing $t_{\text{total}} = t_0 (1 + \epsilon) / (1 - \epsilon)$. So even though the ball bounces an “infinite” number of times, it does so in a finite amount of time.ⁱ We see that the total time before the ball stops bouncing is proportional to the square root of the original height, *which we can use as another way to determine ϵ : we can simply measure how long it takes the ball to stop bouncing when dropped from a known height.*

2 Hypothesis:

The decrease in rebound height of a bouncing ball after successive bounces can be well-characterized by a single parameter ϵ representing the loss in speed after each bounce. Two independent relationships are proposed above, which we can test to verify the relationships.

3 Procedure

You will need a meter stick, a ball to bounce, and a stopwatchⁱⁱ. We will measure the coefficient of restitution ϵ in two different ways, time permitting, and learn how to calculate the best-fit slope to data displaying a linear relationship.

Method 1: *in class*

From the equations above, we note that the ball’s height after n bounces is $h_n = \epsilon^{2n} h_0$, or, the ratio of the height of the n^{th} bounce to the original release height will be ϵ^{2n} . Thus, a plot of $\log \frac{h_n}{h_0}$ (on the y axis) should be linear in n :

ⁱOf course in reality there are not an infinite number of bounces.

ⁱⁱDo a Google search for “stopwatch.” You are feeling lucky.

$$h_n = \epsilon^{2n} h_0$$

$$\frac{h_n}{h_0} = \epsilon^{2n}$$

$$\log \frac{h_n}{h_0} = \log \epsilon^{2n}$$

$$\log \frac{h_n}{h_0} = 2n \log \epsilon$$

$$\implies y = mx + b \quad \text{where} \quad m = 2 \log \epsilon, \quad x = n, \quad y = \log \frac{h_n}{h_0}$$

1. Drop your ball vertically from a given height (~ 1 m) next to an upright meter stick. This is h_0 . Take care not to spin the ball while releasing it.
2. Measure the rebound height after a certain number of bounces.
3. Drop your ball again from *the same height*, and measure the rebound height after a different number of bounces.
4. Do this for several values of n , the number of bounces, and make a data table of your raw data and $\log \frac{h_n}{h_0}$.
5. Plot $\log \frac{h_n}{h_0}$ (as y) versus n (as x) in Excel (scatter plot).
6. The slope of this plot should give you $2 \log \epsilon$, from which you can determine ϵ ; Section 5 instructs you on how to calculate the best-fit slope.
7. Report your value for ϵ with an estimate of its accuracy and the quality of the fit used to determine it.

Repeat this method for a different sort of ball and determine its value of ϵ . Does the ball you consider “bouncier” have a higher or lower ϵ ?

Method 2: *your homework*

From the equations above, plotting the time it takes the ball to stop bouncing should scale as the square root of the initial height it was dropped from. Plotting the total bouncing time (on the y axis) versus the square root of the initial height $\sqrt{h_0}$ (on the x axis) should give a straight line of slope:

$$\text{slope of trendline} = m = \sqrt{\frac{2}{g}} \left(\frac{1 + \epsilon}{1 - \epsilon} \right) \tag{5}$$

where again $g = 9.81 \text{ m/s}^2$. Thus, once you have the slope of a trendline fitting your plot, you can find ϵ :

$$\epsilon = \frac{(\text{slope}) \sqrt{\frac{g}{2}} - 1}{(\text{slope}) \sqrt{\frac{g}{2}} + 1} \quad (6)$$

1. As above, drop the ball next to the meter stick from a known height h_0 .
2. Using a stopwatch, measure how long the ball takes to stop bouncing by some reasonable and reproducible definition (e.g., when the sound of distinct bounces is no longer discernible.)
3. Repeat this for a number of initial heights (*measured in meters*).
4. Plot the total time of bouncing versus $\sqrt{h_0}$ for several initial heights (say, 5). The coefficient of restitution can be found from the slope of this plot, as noted above.

If you finish early enough, repeat method 2 with either a different ball (different material) or a different surface to bounce it off of. The value of ϵ will in general be different for different material combinations.

4 What you should report

- Your plots (including trendline) and data tables
- The equation for your trendline and the r value for the fit
- The values of ϵ you determined, along with an estimate of error

Further Reading:

N. Farkas and R.D. Ramsier, *Physics Education* 41, 73 (2006)

<http://scienceworld.wolfram.com/physics/CoefficientofRestitution.html>

5 Linear Regression: finding a best-fit trendline

Let's say we have a series of data on an $x - y$ plot which seemed to be scattered about a straight line, i.e., the data seem to at least approximately obey a linear relationship. Better yet, perhaps we even have a theoretical model which *predicts* a linear relationship for this particular experimental data.

Linear regression attempts to model the relationship between two variables (say, x and y) by fitting a linear equation (a line) to the observed data. One variable is considered to be an explanatory variable (usually on the x axis), and the other is considered to be a dependent variable (usually on the y axis). The explanatory variable is the thing you are changing or controlling in the experiment, the dependent variable is your primary measurement. For example, in the first experiment above you have acquired and plotted data of the sort $y = \log \frac{h_n}{h_0}$ and $x = n$ which is predicted to obey a linear relationship, $y = mx + b$.

GROUP MEMBERS

If the data *look* linear, and it seems like we could draw a straight line through at least most of the data ... how do we find the best possible line to draw? That is, what is the equation of a straight line that has the minimum total deviation from the actual data set? Further, how well does this line describe our data (i.e., how close is our data to being linear)? The line of best fit is often called a “fit” or “trendline,” and if it is a good enough fit to the data, it allows us to model a large data set approximately by a simple linear relationship. In the present experiment, the slope of this line will yield the coefficient of restitution.

First, some definitions. Let our experimental data set be x_i and y_i , a collection of x and y values. Our first data point is $i = 1$, (x_1, y_1) , and the fifth is (x_5, y_5) . The trend line we seek can be described by

$$y_{\text{fit}} = mx + b \tag{7}$$

Once we determine the proper slope m and intercept b for our trendline, we can plug in a value for our explanatory variable x and the equation predicts a value for our dependent variable y_{fit} . If we choose to plug in the same x values as in our experiment, the corresponding values of y_{fit} will not necessarily match the data exactly for all points. Rather, the best fit trendline is *the one that minimizes the overall differences between the trendline and the data*.

Finding the line of best fit is not terribly different from calculating standard deviation, as we did last time. First, you will need a data table: your x and y values, their squares, and their products.ⁱⁱⁱ Something like the table below; we have included two header columns to give you a better idea of what is going on.

Table 1: Data table

| i | x_i | y_i | x_i^2 | y_i^2 | $x_i y_i$ |
|----------------|-------|-------|---------|---------|-----------|
| 1 | 2 | 4.7 | 4 | 22.09 | 9.4 |
| 2 | 3 | 7.1 | 9 | 50.41 | 21.3 |
| 3 | 5 | 11.2 | 25 | 125.44 | 56.0 |
| 4 | 7 | 14.7 | 49 | 216.09 | 102.9 |
| totals | 17 | 37.7 | 87 | 414.03 | 189.6 |
| average | 4.25 | 9.425 | - | - | - |

Note that for your first experiment, the table header would really be like this:

| measurement | bounces | n | $\log \frac{h_n}{h_0}$ | n^2 | $\left(\log \frac{h_n}{h_0}\right)^2$ | $n \log \frac{h_n}{h_0}$ |
|-------------|---------|-----|------------------------|-------|---------------------------------------|--------------------------|
|-------------|---------|-----|------------------------|-------|---------------------------------------|--------------------------|

Now we have enough data tabulated to determine the line of best fit to our experimental data. The derivation of how and why this works is beyond the scope of our discussion, but the totals above are

ⁱⁱⁱDon't forget that $y = \log \frac{h_n}{h_0}$ and $x = n$ for the first experiment.

enough:

$$\text{trendline slope} = m = \frac{\left(\sum_{i=1}^{i=n} x_i y_i\right) - n\bar{x}\bar{y}}{\left(\sum_{i=1}^{i=n} x_i^2\right) - n\bar{x}^2} \quad (8)$$

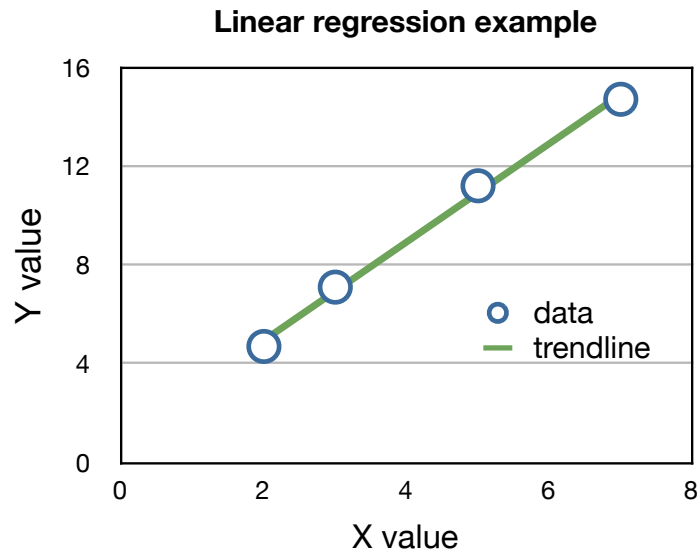
Here the sums are defined as last time; the one in the numerator is the total of the column $x_i y_i$, the one in the denominator is the total of the x_i^2 column. The averages of all x and y values are \bar{x} and \bar{y} , respectively, and n is the total number of data points. For our data, we have

$$m = \frac{189.6 - 4 \times 4.25 \times 9.425}{87 - 4 \times 4.25^2} \approx 1.99 \quad (9)$$

The intercept is found a bit more easily:

$$\text{trendline intercept} = b = \bar{y} - m\bar{x} = 9.425 - 1.99 \times 4.25 \approx 0.968 \quad (10)$$

A neat fact is that trend line must pass through the dataset average point (\bar{x}, \bar{y}) . Here is a plot of our data and the calculated trendline:



It is quite good! But how good? We can also define (but will not derive) a “goodness of fit” parameter r . If $r = 0$, there is no correlation between x and y – total randomness. If $r = -1$, the data are perfectly *negatively* correlated – a line with negative slope. If $r = +1$, the data is perfectly *positively* correlated – a line with positive slope. The closer r is to $+1$, for example, the better the positive correlation. A small

value of r near zero indicates poor correlation. We would guess that our r should be positive, and close to 1 based on the plot above. Here's how we calculate r , using the same data table we already generated:

$$\text{quality of fit} = r = \frac{n \left(\sum_{i=1}^{i=n} x_i y_i \right) - \left(\sum_{i=1}^{i=n} x_i \right) \left(\sum_{i=1}^{i=n} y_i \right)}{\sqrt{n \left(\sum_{i=1}^{i=n} x_i^2 \right) - \left(\sum_{i=1}^{i=n} x_i \right)^2} \sqrt{n \left(\sum_{i=1}^{i=n} y_i^2 \right) - \left(\sum_{i=1}^{i=n} y_i \right)^2}} \quad (11)$$

It is a bit fearsome-looking, but just involves sums we've already done. In our case,

$$r = \frac{4 \times 189.6 - 17 \times 37.7}{\sqrt{4 \times 87 - 17^2} \sqrt{4 \times 414.03 - 37.7^2}} \approx 0.998 \quad (12)$$

Almost perfect positive correlation. Of course, real data is often not quite this good; if you find $r \approx 0.8 - 0.9$ consider it a good day.

How about an analysis of how much the fit line and real data deviate? We can calculate this a lot like standard deviation: at every data point, for a *given* x_i value subtract the real measured y_i value from the value at that x_i calculated from the fit line y_{fit} , and square the result. Add all those squared deviations together, divide by the number of points minus 2, and there is your estimate, called the *root mean square error* of the fit:

$$\text{root mean square error} = \frac{1}{n-2} \sum_{i=1}^{i=n} (y_i - y_{\text{fit}})^2 \quad (13)$$

Like standard deviation, it has the same units as the original data, and tells you that the bulk of the data will fall within \pm this much of the trendline.

Finally: both the trendline generation and quality of fit estimate can be done automatically for a given plot in Excel! Now that you've done it manually, it is no longer just another magic feature: it is based on solid mathematics. One note in passing, Excel will report r^2 rather than r for the quality of the fit. Values of r^2 extend from 0 to 1, with 0 being no correlation, and 1 being perfect correlation.

Further Reading:

http://people.hofstra.edu/Stefan_Waner/realworld/calctopic1/regression.html

<http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm>

http://www.curvefit.com/linear_regression.htm