PH 150
LAB
MANUAL

R. Todd Lines and Matthew R. Zachreson

Department of Physics

Brigham Young University–Idaho

Last Revised: June 7, 2023

Contents

Part I

Lab Introductions

Chapter 1

Measurement and Uncertainty I

Python You Should Know for This Chapter

None

Questions that you should be able to answer by the end of this chapter.

1. Given the quantity $8.1 \pm 0.02\,\mathrm{s}$, what are its *value* and *error*?

2. What is the difference between being *precise* and being *accurate*?

3. You measure a cylinder and find that it is $4.2 \pm 0.05\,\mathrm{cm}$ long and $1.021 \pm 0.004\,\mathrm{cm}$ in diameter. What is its measured volume (including uncertainty)? $V_{cyl} = \pi r^2 l$

4. You measure the acceleration due to gravity as $10.0 \pm 0.15\mathrm{m/s^2}$. What is your percent error? Was your measurement technique successful?

1.1 The Philosophy of Experimental Measurement

In experimental measurement, we assume that at a single instant of time that there is a “correct" value for some physical quantity we wish to know. For example, how fast a car is going at a particular time. The correct value might be something like $64.99999\,\mathrm{mi/h}$ (because we don’t speed here at BYU-I).

The problem is that we can never measure this “correct" value. That is because no instrument is perfectly accurate. We have to make do with the goal of getting as close as we can to the “correct" value. Your speedometer, for example, probably would report this speed as $65\,\mathrm{mi/h}$. But that is not exactly correct.

Several factors[1] affect how well we can measure quantities:

- Proper instruments
- Instrument calibration
- Measurement repeatability
- Quantization error

[1] Actually there are more factors, but we will deal with the first three in PH150. PH 250 will deal with computer control and so will likely introduce quantization error. If you are a physics major, you will take PH336 and deal with additional error factors.

1.1.1 Values

In experimental physics we use the word *value* to mean a number and its units. For example, we might measure a metal rod and say that the length of the rod is

$$L = 97.6\,\mathrm{cm}$$

The number is 97.6 and the units are centimeters (cm). Together they are a value.

Numbers without units are not useful in experimental physics. **You should always report both a number and its units.**

1.1.2 Errors

I give you this rod length, all you really know is that the rod is not longer than about 97.7 cm or shorter than 97.5 cm (assuming you trust my ability to measure rods!). Could you be sure that the rod was not really 97.61 cm or maybe 97.62 cm?

After making a measurement we have some uncertainty left over because of the limitation of our instrument. In this case our instrument is a meter stick, and you know that meter sticks have a smallest tick mark spacing, usually 1 mm. I could probably judge to within half a millimeter. But it would be kind of crazy to say that you could be correct to within a 0.000001 m using a meter stick. Additionally, meter sticks expand and contract with changes in temperature. So the meter stick itself[2] is not correct to within a hundredth of a millimeter!

One way to express this uncertainty in a measured value is by using significant figures[3]. As an example:

$$\begin{array}{c}\text{Most Significant Figure}\\ \downarrow \\ 97.6\,\mathrm{cm} \\ \uparrow \\ \text{Least Significant Figure}\end{array}$$

We assume the last figure (in this case the 6) holds the uncertainty. Think of making this measurement. You would not be to uncertain about the 90 cm represented by the first digit. It is a big mark on the meter stick, and you could probably tell that the rod was almost a meter long without actually measuring. You are probably not to uncertain about the 7 cm represented by the second digit. You can easily read this from the meter stick. You will have to count millimeter marks to get the 0.6 cm represented by the last digit. But it is unlikely that the rod will end exactly at on the sixth millimeter mark. So this is where our uncertainty comes in. This is why we call the last digit the least significant digit, because it is the most uncertain.

If I give you a measurement like 97.6 cm you would assume that I could be off by 0.1 cm or by one millimeter. That is what stating 97.6 cm means. It would be better to state this explicitly

$$L = 97.6 \pm 0.1\,\mathrm{cm}$$

[2] The meter stick changes in length by about 3 μm per $^\circ$C

[3] You have probably done this in high school. If not, ask one of your classmates or your instructor.

But you may object! You can do better than $\pm 0.1\,\text{cm}$ with a meter stick. And so can I. This is one reason using significant figures is not such a great idea. It would be better to state our measurement and tell the person with whom we are communicating what we think the uncertainty really is. Say

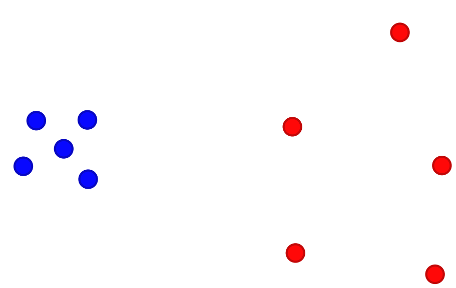
$$L = 97.65 \pm 0.03\,\text{cm}$$

This means that I measured 97 cm, then counted up six millimeter marks, and noticed that the rod end landed at just about half a mark more beyond the sixth mark. I am telling you that I think because of thermal expansion (and my poor eyesight) that I can only be sure of this measurement to within about a third of a millimeter. **We will always use this notation to report values and their errors in this class.** If you report a value without an uncertainty, something is wrong (and the grader will surely notice!). This is because in your actual jobs as scientists not being clear about how well you know a value can be disastrous, even causing loss of life or property. So to keep you safe in future jobs, we will use best practices here.

1.1.3 Precision

So far when we have considered how correct our value is what we have really been talking about is the *precision* of our measurement. If I measure the metal rod 50 times, I will get about the same measurement, but not quite, each time. I might do a poor job of lining up the meter stick and the rod, or the temperature might change and the meter stick might shrink or expand. Whatever the problem, each measurement will be a little different. This small fluctuation about the "correct" value is called the precision of the measurement. It tells us how likely it is to get the same value each time we perform the experiment.

Think of throwing darts. The dots in the next figure represent the location of five darts from two dart players.



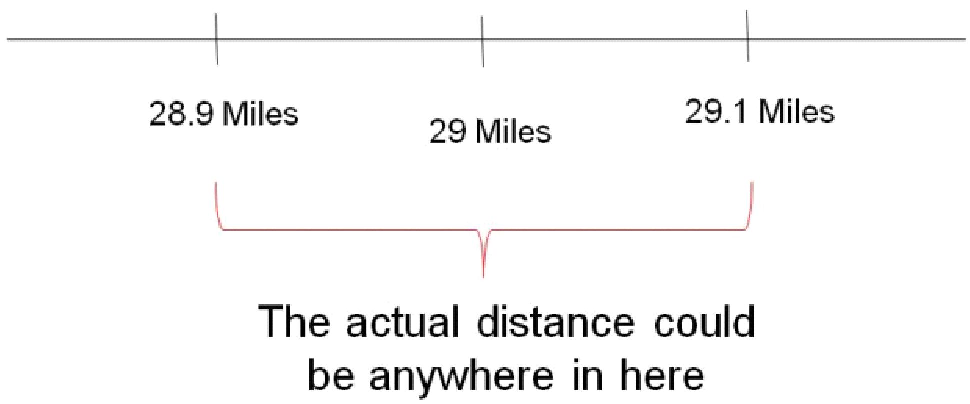
The blue dots show less spread in their locations than the red dots do. We would say that the blue dart player was more precise. Another way to say this is there is less uncertainty in where the blue player's darts will go. In our notation for reporting a value the plus-or-minus-part is this uncertainty[4]:

[4] The traditional symbolic notation for an uncertainty in A is either δA or σ_A. Both are used in this manual, and either is ok to use in this course.

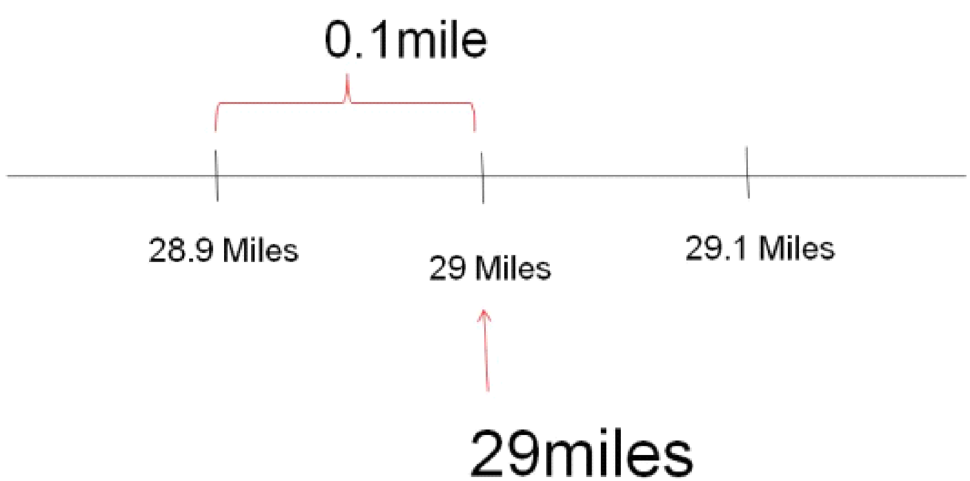
$$A = \overset{\substack{\text{Nominal Value}\\ \downarrow}}{A} \pm \underset{\substack{\uparrow\\ \text{Uncertainty}}}{\delta A}$$

We call the actual measurement the *nominal value*. That would be where the dart thrower was aiming. If you know statistics, you can see that this is a little bit like an average and standard deviation.

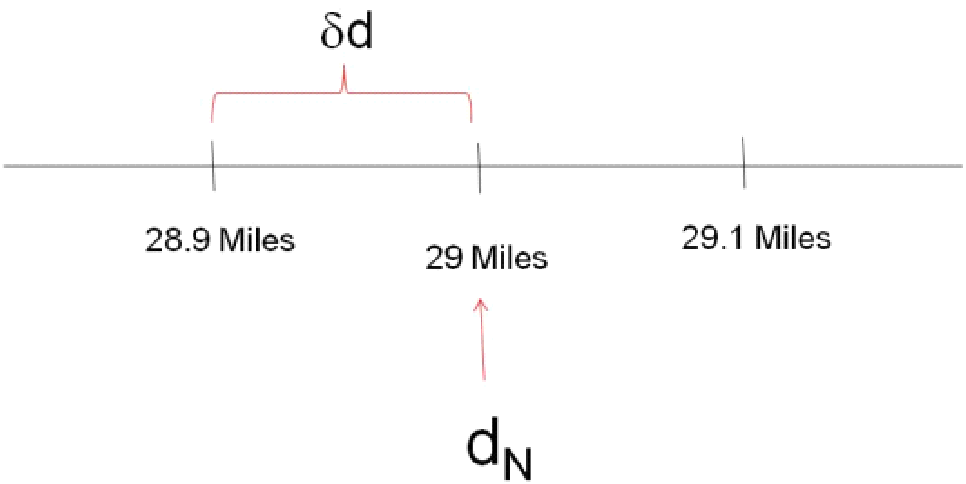
Here is another example. Suppose we drive to Idaho falls. The distance is about 29 miles. If we use our odometer we might find it is marked in 10ths of a mile. So if I report a drive of 29 miles, I might have gone 29.1 miles, or 28.9 miles. Let's place these on a number line.



Notice that my uncertainty is 0.1mile.



If the distance is called d, then we write the uncertainty in the distance by placing the Greek letter δ in front of our name to get δd. This is read "delta d" or the "uncertainty in d."



We name the actual measurement, d_N where the N is for " nominal value" so then

$$d = d_N \pm \delta d = 29 \pm 0.1\,\text{mi}$$

In this case, the 0.1 mi represents what is called the *absolute uncertainty*, and is how we generally report uncertainties. But, sometimes it is more appropriate to use the *relative uncertainty*. Relative uncertaintity is very useful for judging the quality of your measurement. Here's two examples to illustrate:

First, I measure the distance from Rexburg, ID to Rochester, NY as about 2082.90 miles or 3352100m. Further suppose that I tell you I have made this measurement to ± 1 m. Is this measurement good?

Now suppose I measure one of our lab tables. I get that the table is 2 m long and I tell you that my measurement is good to ± 1 m. Is this measurement good?

You can probably see that the first measurement is very good, while the second measurement could probably have been done better by guessing the length of the table. It is a terrible measurement with too much uncertainty.

But what makes the difference? The uncertainty is the same in both cases! Of course the difference is that in one case the uncertainty is a tiny fraction of the whole value, while in the table case the uncertainty is a large fraction of the measured value. If the uncertainty is large compared to the measured value, it is not a good measurement.

But we need a way to communicate this. The error (absolute) is ± 1 m in both cases. Though the absolute uncertainty can't always tell us the quality of a measurement, we can use it to calculate

$$\frac{\delta L}{L} = \text{relative uncertainty}$$

This gives the error as a percentage of the total measurement. For our Rexburg to Rochester measurement

$$\frac{\delta L}{L} = \frac{1\,\text{m}}{352100\,\text{m}} = 2.8401 \times 10^{-6}$$

and for the table

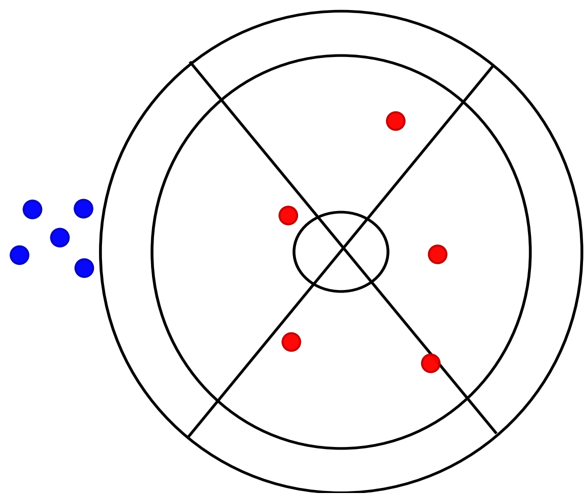
$$\frac{\delta L}{L} = \frac{1\,\mathrm{m}}{2\,\mathrm{m}} = 0.5$$

Since a good measurement has an error that is a small percentage of the total measurement, we can easily identify which measurement is better by calculating the percentage of the total value represented by the uncertainty and observing how small it is. We call this the “relative uncertainty."

The relative uncertainty will be small for a good measurement and large for a bad one. In this case we can easily see that the Rochester measurement is much better by looking at the relative uncertainty and noting that it is much smaller than the table relative uncertainty.

1.1.4 Accuracy

Having a high precision measurement is good, but not enough. Here is a picture of our darts again, but this time I have included a target. Suppose you were trying to get a bull’s-eye. We can see that our blue dart person is very precise, but he missed the bull’s-eye–and the target! We need more than precision.



Notice that if we look at the average location of the red dart dots, the red dot thrower does seem to be aiming right at the target. We would say that he is accurate. Accuracy is whether or not you are aiming at the target. If we drive $29 \pm 0.1\,\mathrm{mi}$ as we discussed earlier, but we end up in Ashton instead of Idaho Falls, we would say that no matter how precise our driving distance is, we did not achieve our goal of getting to Idaho Falls. We mean that we are not accurate.

This might occur in the lab by having a scale that is not zeroed, or a ruler that is too short or has extra material on the end. The measured amount given by an instrument when it should be measuring zero is called the“zero offset." If we know about the problem, we can just adjust the final number by this amount (or re-zero the instrument). But often we don’t know about such problems and they can be hard to detect.

This type of error is called a *systematic error* because it affects the system each time it is used. The device will always be off by this amount until we fix it. We can improve our uncertainty estimate by taking many measurements and using the average as our value. But if we aim at the wrong place, no matter how many darts we throw, the accuracy error won't get better.

Of course what we want is both accuracy and precision in our measurements.

1.2 Combining uncertainty

Suppose I want you to calculate the area of the cover of your text book. That is easy you say! it is the width times the height. And you are right

$$A = w \times h$$

Suppose we measure the book and find it has a height of $h = 28.4\,\text{cm} \pm 0.2\,\text{cm}$ and a width of $22.2\,\text{cm} \pm 0.02\,\text{cm}$. So the area is

$$\begin{aligned} A &= 28.4\,\text{cm} \times 22.2\,\text{cm} \\ &= 630.48\,\text{cm}^2 \end{aligned}$$

but wait, what do we do with the uncertainties? If the initial measurements of the lengths are uncertain, then the area made from them must be more uncertain. We need a way to combine our uncertainties for the area. There are three generally accepted ways to calculate how much our measurement error affects our results[5] They are the algebraic method (which is used only in fields where you aren't expected to know calculus) standard or Gaussian error propagation (the gold standard), and Monte Carlo (used when systems are too complex to do standard error propagation). This chapter will discuss the standard and algebraic methods.

[5] Each of these methods fall under the more general term of *error propagation* or *propagation of error*.

I've also included the "High/Low Method" as an example. You **should never use it in a professional setting**, but it is very good at illustrating how errors in measurement affect your results.

1.2.1 The High/Low Method

Knowing what our uncertainty means now[6], we can estimate the uncertainty in the area calculated above. We can guess that if we were off by a positive +0.2 cm on both measurements, then we would have the biggest area we could possibly get from our measurements. In some way it would be the most off we could get. Let's call this A_{max}. We would find it to be

[6] Reminder: it represents the distance to the poorest measurements from a group of measurements.

$$\begin{aligned} A_{\text{max}} &= (28.4\,\text{cm} + 0.02\,\text{cm}) \times (22.2\,\text{cm} + 0.02\,\text{cm}) \\ &= 28.6\,\text{cm} \times 22.4\,\text{cm} \\ &= 640.64\,\text{cm}^2 \end{aligned}$$

Likewise, if we were off by $-0.2\,\mathrm{cm}$ on both measurements, then we would have the smallest area we could possibly get from our measurements. In another way it would be the most off we could get. Let's call this A_{min}. It would be

$$\begin{aligned} A_{\mathrm{min}} &= (28.4\,\mathrm{cm} - 0.02\,\mathrm{cm}) \times (22.2\,\mathrm{cm} - 0.02\,\mathrm{cm}) \\ &= 28.2\,\mathrm{cm} \times 22.0\,\mathrm{cm} \\ &= 620.4\,\mathrm{cm}^2 \end{aligned}$$

To find the uncertainty, consider our trip to Idaho Falls. We found that the uncertainty was half the distance between our maximum estimate of our distance and the minimum estimate of our distance. We can use the same procedure for our area. We have the maximum and the minimum areas. The uncertainty is half the difference between these two extremes.

$$\delta A = \frac{A_{\mathrm{max}} - A_{\mathrm{min}}}{2}$$

using our numbers

$$\delta A = \frac{640.64\,\mathrm{cm}^2 - 620.4\,\mathrm{cm}^2}{2} = 10.12\,\mathrm{cm}^2$$

so our area should be reported as

$$A = 630\,\mathrm{cm}^2 \pm 10\,\mathrm{cm}^2$$

There are some tricks to this. We have to make sure we have the biggest value we can get when we get the maximum and the smallest value when we get the minimum. Suppose I measure two distances $x = 1.5\,\mathrm{m} \pm 0.3\,\mathrm{m}$ and $y = 3.0\,\mathrm{m} \pm 0.2\,\mathrm{m}$ and I want to calculate

$$z = \frac{y}{x}$$

then

$$\begin{aligned} z &= \frac{3.0\,\mathrm{m}}{1.5\,\mathrm{m}} \\ &= 2.0 \end{aligned}$$

what would the uncertainty in z be? Last time we chose adding the plus uncertainty to both values to get the maximum and we subtracted off both uncertainty values to get the minimum, but this time lets try every combination of plus and minus uncertainties

$$\frac{3.0\,\mathrm{m} + 0.2\,\mathrm{m}}{1.5\,\mathrm{m} + 0.3\,\mathrm{m}} = 1.7778$$

$$\frac{3.0\,\mathrm{m} + 0.2\,\mathrm{m}}{1.5\,\mathrm{m} - 0.3\,\mathrm{m}} = 2.6667$$

$$\frac{3.0\,\mathrm{m} - 0.2\,\mathrm{m}}{1.5\,\mathrm{m} + 0.3\,\mathrm{m}} = 1.5556$$

$$\frac{3.0\,\mathrm{m} - 0.2\,\mathrm{m}}{1.5\,\mathrm{m} - 0.3\,\mathrm{m}} = 2.3333$$

Note that using both + signs did not give the largest value. That is because for division a smaller denominator makes the fraction bigger. For the maximum we want a + in the numerator and a − in the denominator. For the minimum we want a − in the numerator and a + in the denominator. It is a little tricky, but if we think, we can find the very biggest possible value and the very smallest possible value every time. To finish this off, lets find the uncertainty in z

$$\begin{aligned}\delta z &= \frac{z_{\text{max}} - z_{\text{min}}}{2} \\ &= \frac{2.6667 - 1.5556}{2} \\ &= 0.55555\end{aligned}$$

Now imagine checking every single possible combination of + and − on a more complex equation until you found the highest and lowest values. The more formal methods avoid that added complexity, and take some statistics[7] into account to give a more accurate and consistent way to calculate error.

If you already know how to take a derivative, skip the algebraic method and read the section on Standard Error Propagation. If you do not yet know how to take a derivative, skip the Standard Error Propagation section and come back to it when you do know how to take a derivative.

[7] They take advantage of the central limit theorem to reduce cross talk between uncertainties, and the fact that it is very unlikely that all of your measurements are at their maximum or minimum value. Any more details on how are beyond the scope of this book.

1.2.2 Algebraic method

Reminder: if you already know how to take a derivative, jump ahead to the Standard Error Propagation section.

Using algebra, we can develop rules for combining uncertainty when multiplying, dividing, adding, subtracting, or raising variables to whole number powers[8]. These rules will cover many simple situations, but eventually we will need to know how to estimate uncertainty for any function as explained in the Standard Error Propagation section. But for now our goal is to have a method that you can use without calculus.

[8] This is really just multiplying the same value by itself multiple times, so it essentially follows the same rules as multiplication

Here are the rules:

Function		Uncertainty Formula
Addition	$z = x + y$	$\delta z = \delta x + \delta y$
Subtraction	$z = x - y$	$\delta z = \delta x + \delta y$
Multiplication	$z = xy$	$\frac{\delta z}{\lvert z_N \rvert} = \left(\frac{\delta x}{x_N} + \frac{\delta y}{y_N}\right)$
Division	$z = \frac{x}{y}$	$\frac{\delta z}{\lvert z_N \rvert} = \left(\frac{\delta x}{x_N} + \frac{\delta y}{y_N}\right)$
Multiply by constant	$z = ax$	$\delta z = a\delta x$
Powers	$z = x^n$	$\frac{\delta z}{\lvert z_N \rvert} = n\frac{\delta x}{\lvert x_N \rvert}$

Here are some examples on how to use the rules.

Multiplication

Let's start with the area example from the previous section. As a reminder, we measured a book and found that it has a height of $h = 28.4\,\text{cm} \pm 0.2\,\text{cm}$ and a width of $22.2\,\text{cm} \pm 0.02\,\text{cm}$. To find the area, we use:

$$A = hw = (28.4\,\text{cm})(22.2\,\text{cm}) = 630.48\,\text{cm}^2$$

Applying the multiplication rule to our area formula gives:

$$\frac{\delta A}{A} = \frac{\delta h}{h} + \frac{\delta w}{w} \rightarrow \delta A = \left(\frac{\delta h}{h} + \frac{\delta w}{w}\right) A$$

And now with numbers:

$$\begin{aligned}\delta A &= \left[\frac{0.2\,\text{cm}}{28.4\,\text{cm}} + \frac{0.02\,\text{cm}}{22.2\,\text{cm}}\right]\left(630.48\,\text{cm}^2\right)\\ &= [0.00704 + 0.000901]\left(630.48\,\text{cm}^2\right)\\ &= 5\,\text{cm}^2\end{aligned}$$

Therefore, the proper way to report the area of the book would be[9]

$$A = 630 \pm 5\,\text{cm}^2$$

[9] Notice how the value was rounded to match the presicion of the error.

Addition

What if instead of finding the area of the book, we wanted to find its perimeter? For that the formula is:

$$P = 2h + 2w = 2\,(28.4\,\text{cm}) + 2\,(22.2\,\text{cm}) = 101.2\,\text{cm}$$

Using the addition rule, the formula to find the error would then be[10]:

$$\delta P = 2\delta h + 2\delta w = 0.4\,\text{cm} + 0.04\,\text{cm} = 0.44\,\text{cm}$$

[10] Notice that this formula uses both the addition rule *and* the multiply by a constant rule.

Therefore, the proper way to report the book's perimeter is:

$$P = 101.2 \pm 0.44\,\text{cm}$$

1.2.3 Standard Error Propagation

Reminder: this section is for those who already know how to take a derivative. If you do not know calculus, come back to this section when you do.

Partial Derivatives

The standard way to do error propagation involves something called a *partial derivative*. If you already know how to do a regular derivative, partial derivatives are almost the exact same thing. You just change what you consider a constant. For example, when taking the regular derivative of $2x^2$, you do this:

$$\frac{d}{dx}2x^2 = 2\frac{d}{dx}x^2 = 2(2x) = 4x$$

The derivative doesn't do anything to the constant 2. To put that into a symbolic form that you might be a little more used to, if a and n are constants,

$$\frac{d}{dx}ax^n = a\frac{d}{dx}x^n = anx^{n-1}$$

Here is that same statement written as a partial derivative:

$$\frac{\partial}{\partial x}ax^n = a\frac{\partial}{\partial x}x^n = anx^{n-1}$$

Notice that a partial derivative uses a ∂ symbol in place of a d. That change in symbols tells you to treat anything that isn't what you are taking a derivative with respect to (in this case, x) should be treated as a constant. Here's an example. Assuming we have a function $f = 5x^2y^3$, our partial derivatives would be:

$$\begin{aligned}\frac{\partial f}{\partial x} = \frac{\partial}{\partial x}\left[5x^2y^3\right] &= 5y^3\frac{\partial}{\partial x}x^2 = 10y^3x\\ \frac{\partial f}{\partial y} = \frac{\partial}{\partial y}\left[5x^2y^3\right] &= 5x^2\frac{\partial}{\partial y}y^3 = 15x^2y^2\end{aligned}$$

Calculating Error

Here is the traditional[11] way of writing out the formula that will calculate the error in any function $f(x, y, z, ...)$:

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

[11] It's traditional because it is a very good way to *remember* how to calculate the error, not because it is necessarily the best way to *learn*.

For any equation, you create the necessary error equation with these steps (an example will follow):

1. Clearly identify which terms in your calculation have an error or uncertainty.

2. Take the partial derivative of your equation with respect to one of your error terms.

3. Square the number that you get when you plug your measured values into the partial derivative that you just took..

4. Multiply the number that you just got by the error in the term you took the derivative with respect to, squared. (This is one error term)
5. Repeat for all the measurements for which you have an uncertainty, then add all of the error terms squared
6. Take the square root of the total

Here's standard error propagation applied to the book area example from before. As a reminder, we measured a book and found that it has a height of $h = 28.4 \pm 0.2\,\text{cm}$ and a width of $22.2 \pm 0.02\,\text{cm}$. To find the area, we use:

$$A = hw = (28.4\,\text{cm})(22.2\,\text{cm}) = 630.48\,\text{cm}^2$$

Since both h and w have uncertaintity, we have to take the partial derivative with respect to each of them:

$$\begin{aligned}\frac{\partial A}{\partial h} &= \frac{\partial}{\partial h} hw &&= w\frac{\partial}{\partial h} h = w \\ \frac{\partial A}{\partial w} &= \frac{\partial}{\partial w} hw &&= h\frac{\partial}{\partial w} w = h\end{aligned}$$

Now we can calculate the total uncertainty:

$$\begin{aligned}(\delta A)^2 &= \left(\frac{\partial A}{\partial h}\right)^2 (\delta h)^2 + \left(\frac{\partial A}{\partial w}\right)^2 (\delta w)^2 \\ &= (w)^2 (\delta h)^2 + (h)^2 (\delta w)^2 \\ &= (22.2\,\text{cm})^2 (0.2\,\text{cm})^2 + (28.4\,\text{cm})^2 (0.02\,\text{cm})^2 \\ &= 19.71\,\text{cm}^4 + 0.32\,\text{cm}^4 \\ &= 20.03\,\text{cm}^4 \\ \delta A &= \sqrt{20.03}\,\text{cm}^2 \\ &= 4.5\,\text{cm}^2\end{aligned}$$

Therefore, the proper way to quote our calculated area is

$$A = 630.5 \pm 4.5\,\text{cm}$$

1.3 Judging success of an experiment

Now we know how to describe the uncertainty in a measurement and we can even judge if a measurement is a good one using relative uncertainties. We can find final uncertainties after a calculation. But how do we know, based on our measurements, if our experiment is a success?

If we have a known value, we can compare our experimental results to that known value and judge our accuracy. We do this with a percent error[12].

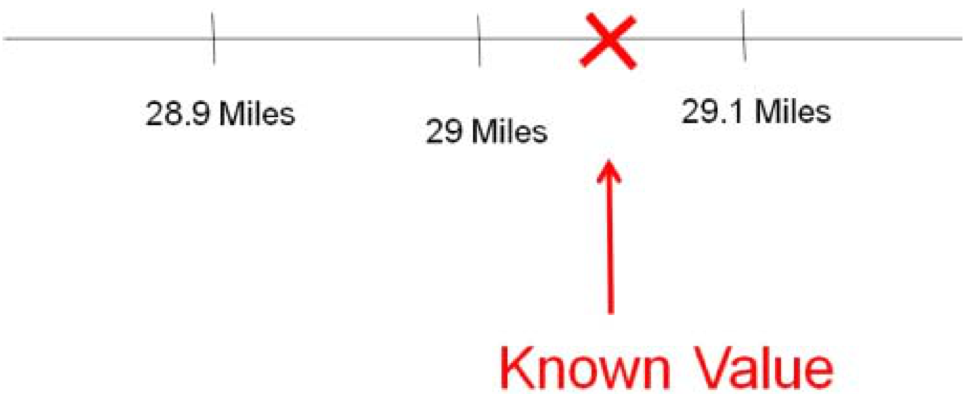
[12] Reminder: You can only use percent error to judge the success of an experiment if you have a known value for comparison.

$$PE = \left(\frac{\|\text{measured value - accepted value}\|}{\text{accepted value}} \times 100 \right)$$

We can compare this to our relative uncertainty

$$RE = \left(\frac{\delta(value)}{\text{nominal value}} \times 100 \right)$$

Let's take our drive to IF as an example. Suppose we have a reliable study that shows the distance to IF is 29.05 mi



And we go to IF and find that our odometer measures 29 ± 0.1 mi.

The percent error is

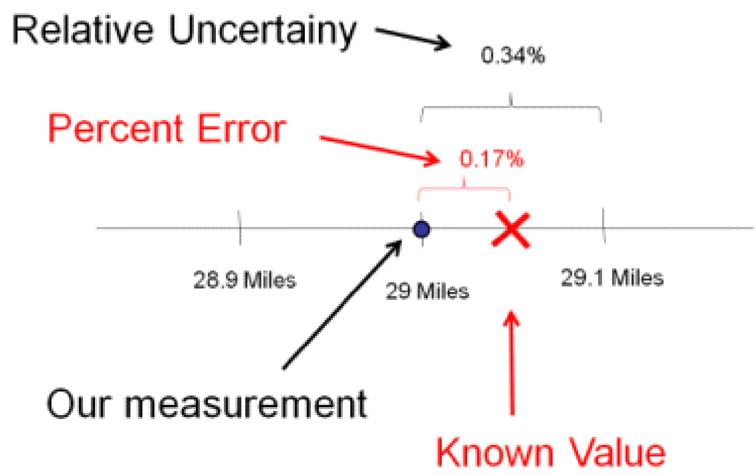
$$\begin{aligned} PE &= \left(\frac{29.05\,\text{mi} - 29\,\text{mi}}{29\,\text{mi}} \times 100 \right) \% \\ &= 0.172\,41 \end{aligned}$$

This is roughly a 0.2% error.

Our relative uncertainty is

$$\begin{aligned} RE &= \left(\frac{0.1\,\text{mi}}{29\,\text{mi}} \times 100 \right) \% \\ &= 0.344\,83\% \end{aligned}$$

Let's see what this means



We can see that we are off from the known value by 0.17%, but remember we are uncertain in our measurement. Our uncertainty tells us we can be anywhere within 0.34% of the value we measured. Since our percent error–how much we are off–is less than the fractional uncertainty—percent off we can be based on our equipment and our technique–we can say that this is an accurate value for the distance to Idaho Falls. More succinctly: if our percent error is smaller than our fractional uncertainty, we are accurate.

But suppose our percent error is larger than the relative uncertainty? Then we are not accurate. It is always good when this happens to try to figure out what the problem could be. There may be a systematic error, or it may be that you failed to recognize some source of error.

Here is a rule of thumb for judging the accuracy of an experiment.

1. If the relative uncertainly is larger than the percent error:
   - The experiment is accurate to within the uncertainty of the experimental technique
   - To improve this measurement, you need better equipment or better technique
2. The relative uncertainty is smaller than the percent error:
   - The experiment is not accurate to within the uncertainty of the experimental technique
   - To improve this experiment, look for systematic errors
   - Consider if you have underestimated the uncertainty

We report this along with our results.

In our first lab, we will get some practice calculating uncertainties and judging accuracies.

Chapter 2

Communicating Results I: Statistical Representation of Data

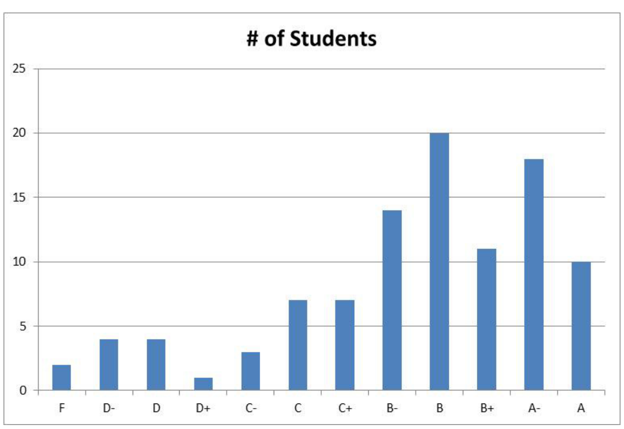
Python You Should Know for This Chapter

- What these terms mean, and how to work with them: *List, variable, index, function*
- How to run a Python program
- How to import the `numpy` library

Recommended reading: Introduction to Scientific Computing in Python, by Nelson and Zachreson; All of Chapters 1 and 3, as well as Sections 4.1 and 4.2

Questions that you should be able to answer by the end of this chapter.

1. What are the mean and standard deviation of these numbers: 3,3,2,4,6
2. You take 10 measurements and find that they have a mean of 5.3 seconds and a standard deviation of 1.1 seconds. What is the uncertainty of your measurement?
3. Using the figure below, how many students earned an A? How many earned less than a C?

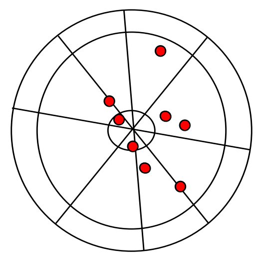


4. Given the histogram above, what is the modal grade (the mode of the grades)?

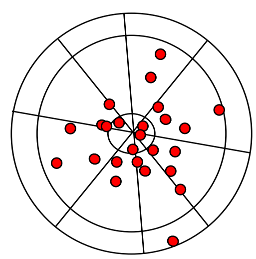
5. Here are a few grades from another class: B,D,A,A,C. What is the median grade?

So far we have talked about repeating experiments, but we have been too pressed for time to actually do that. We should take the time to see how to report data from multiple results. Let's also tie the idea of multiple results to our ideas of uncertainty.

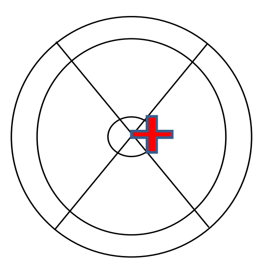
To do this, I would like to go back to our dart board. Suppose I throw the darts, trying for a bull's eye, and I get the following pattern:



We now know that this is fairly accurate, but not very precise. We say that there is a large uncertainty, but that we are aimed about the right direction. We could get a better estimate of how accurate we are by repeating the experiment many times:



and finding an average location of the darts:



This average seems to be just a little right of center. Now we know that we should point the darts a little to the left. Many experiments are like this. We can repeat the experiment many times. The uncertainty might be larger than we want, but if we average over many trials of the experiment, we can find an average value that represents the actual value of the quantity we are trying to find.

2.1 Mean value as our best estimate value

The mathematical process we use to find the mean is simple and you are probably quite familiar with it. We simply add up all the values, and divide the sum by the number of values.

$$\begin{aligned}\bar{x} &= \frac{x_1 + x_2 + x_3 + \cdots x_N}{N} \\ &= \frac{1}{N}\sum_{i=1}^{N} x_i\end{aligned}$$

The last equation uses sigma notation. It is read as " one over N times the sum of x_i for $i = 1$ to N. It is a short-hand notation for the line above. We will use this notation because it makes writing our equations much easier. But that means it is very important that we understand what it means. So let's imagine that we have many values for the x-position for our darts.

$$\begin{aligned}x_1 &= 1.00 \pm 0.01\text{cm} \\ x_2 &= 0.50 \pm 0.01\text{cm} \\ x_3 &= -0.75 \pm 0.01\text{cm} \\ x_4 &= -2.25 \pm 0.01\text{cm} \\ x_5 &= 3.00 \pm 0.01\text{cm} \\ x_6 &= -0.80 \pm 0.01\text{cm} \\ x_7 &= 2.10 \pm 0.01\text{cm} \\ x_8 &= 1.2 \pm 0.01\text{cm}\end{aligned}$$

We have labeled each x with a number. That is what the x_i means. The " i" is an index. It stands for any number from 1 to N. Our sigma notation says we add up all these positions, and divide by $N = 8$ since there are eight positions

$$\begin{aligned}\bar{x} &= \frac{(1.00 + 0.50 - 0.75 - 2.25 + 3.00 - 0.80 + 2.10 + 1.2)\,\text{cm}}{8} \\ &= 0.5\text{cm}\end{aligned}$$

which is a little bit to the right of our zero point.

2.2 Standard deviation as an estimate of our uncertainty

But what is our uncertainty? Each of our position measurements were good to ± 0.01cm. But this can't be what governs our uncertainty. We can see our points are spread out much more than ± 0.01cm. Something in the experiment (the bad dart thrower) is increasing the uncertainty. We could use our algebraic method to find the uncertainty, but that would be tedious and may not include the effects of the dart thrower. It would be great to have a way to use the spread of the points, itself, to obtain a numerical estimate of the uncertainty. The spread must include the effects of the dart thrower.

From your study of statistics, you can guess what we will use to represent uncertainty, but let's reason it out here. We could take how far each point is from where we aimed as an indication of how imprecise our throw was. That would be

$$\Delta x_i = \bar{x} - x_i$$

for each throw. In this equation we are using the Greek Δ to show a difference, and a bar over the x to mean " the average value of the x-position." Then Δx_i is how far off the i^{th} trow from the mean. Sometimes we are off to the right, and sometimes to the left. If we add up all the Δx_i values and average them, they will average to nearly zero most of the time. We can see that zero is not a good estimate of our uncertainty! So the average deviation won't work as a measure of uncertainty.

But we can play a trick. The quantity

$$\Delta x_i^2 = (\bar{x} - x_i)^2$$

is always positive. If we averaged Δx_i^2,

$$\overline{\Delta x_i^2} = \frac{1}{N}\sum_{i=1}^{N} \Delta x_i^2 = \frac{1}{N}\sum_{i=1}^{N} (\bar{x} - x_i)^2$$

nothing would cancel out. And we have solved our calcelation problem. But we have created another problem by doing this, $\overline{\Delta x_i^2}$ is like the square of our how far we are off. So let's take a square root

$$\sqrt{\overline{\Delta x_i^2}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \Delta x_i^2} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\bar{x} - x_i)^2}$$

The quantity $\sqrt{\overline{\Delta x_i^2}}$, represents about how far off we are on average, it does not tend to zero, and has the same units as x_i so it can be an estimate of our uncertainty. It is about how far most of the points are off from the mean. But $\sqrt{\overline{\Delta x_i^2}}$ is a little hard to write, so we usually give this quantity the symbol σ, which is a Greek letter s and is pronounced " sigma." We also give σ a name. We call it the *standard deviation* because it is about how much the average point " deviates" from the mean position. So for our x-position we can write

$$\sigma_x = \sqrt{\sum_{i=1}^{N} \frac{(x_i - \bar{x})^2}{N}}$$

But what does this math symbology mean? To find σ_x, we must first find the average positions to find $\bar{x}$, then we take each x-position (x_i) and we subtract the mean from it $(x_i - \bar{x})$. We square the result. We do this for each of our x-positions. Then we have $(x_1 - \bar{x})^2$, $(x_2 - \bar{x})^2$, $(x_3 - \bar{x})^2, \cdots (x_N - \bar{x})^2$. We add these up, and divide by N to find the average $\sum_{i=1}^{N} \frac{(x_i - \bar{x})^2}{N}$. Then we take the square root.

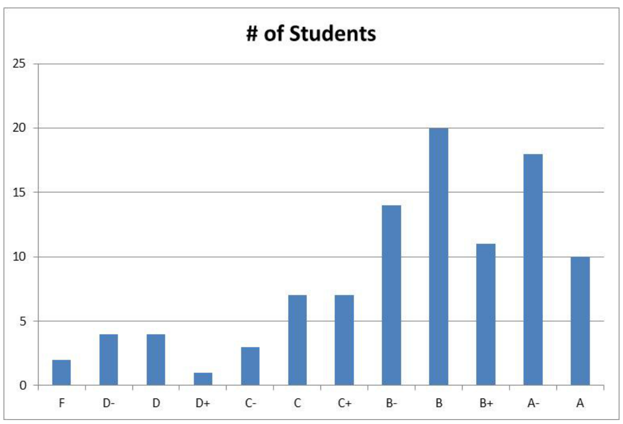
[13] Hopefully this example is enough to convince you that you never want to do this by hand! Normally we will use a computer to do this. You may have already done it in an spreadsheet program like Excel, but I suggest you use Python to do these calculations, and not your calculator.

Here's an example[13] using the dart data from before:

$$\begin{aligned}\sigma_x &= \Big(\big[(0.5-1.00)^2+(0.5-0.50)^2+(0.5+0.75)^2+(0.5+2.25)^2\\ &\quad +(0.5-3.00)^2+(0.5+0.80)^2+(0.5-2.10)^2+(0.5-1.2)^2\big]\,\mathrm{cm}^2/8\Big)^{\frac{1}{2}}\\ &= 1.6\,\mathrm{cm}\end{aligned}$$

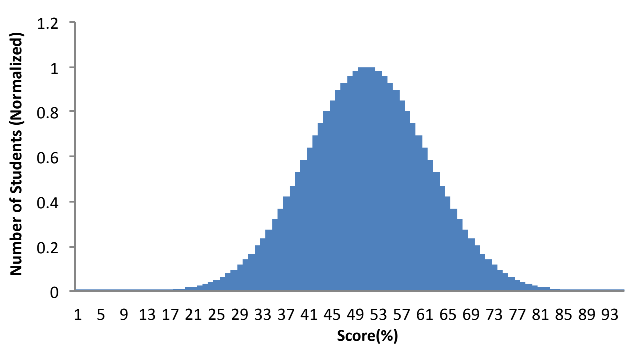
2.3 Histograms

Suppose I plot the results of many, many dart throws. The way I want to plot this is something you have seen from grading for many years. I want the horizontal axis to show the x-position of the dart throws. I want the y-axis to show the number of darts that landed at a particular x-position. This type of graph is called a histogram. You often see grades given like this

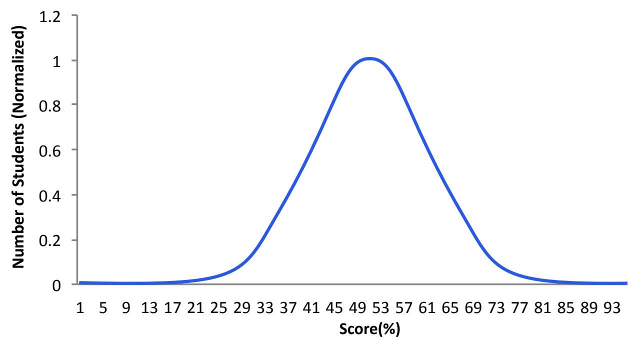


where we understand that the bars indicate how many students got an A (two in this case) and how many got an $A-$ (five in this case) etc.

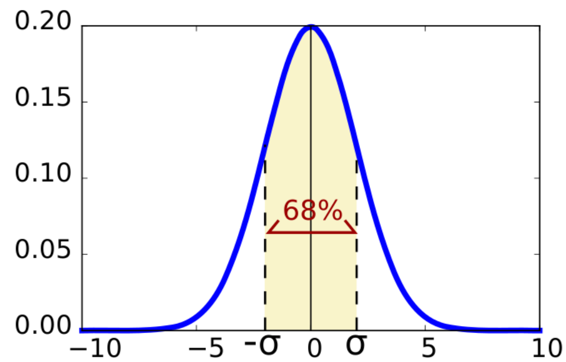
If there are many students we can plot their scores and the shape of the histogram begins to smooth out some



If we had infinitely many students, we would get a perfectly smooth curve. You can see already that coloring in the bars in the graph is not useful any more. So usually we just draw a point for the top of each bar. These points form a curve.



Unlike student scores, dart positions can be negative. So our dart distribution should be centered on zero displacement. We will usually find that 68% of the darts will fall within $\pm\sigma$ of the mean.



We can see that our σ value is very like an uncertainty. But there is a difference. We still have 32% of our experiments outside of $\pm\sigma$, and if we give the uncertainty, δx, then all of the measurements should be within $\pm\delta x$. If you are building a space shuttle and absolutely need to guarantee that your error on your calcuation is within some limit, then you should use a true absolute uncertainty, $\pm\delta x$. But for most experiments, being that certain about our uncertainty is not required, and we can use $\pm\sigma$ as a good approximation to the uncertainty. We will often do this in this class. If losing 32% is not acceptable, but finding the true δx is not practical, it is often good enough to use 2σ or 3σ as the estimate of our uncertainty. 95% of the data will fall with $\pm 2\sigma$, and 99.7% of the data will fall within $\pm 3\sigma$. So these are more conservative estimates than using a single standard deviation. But in this class we will stick with just σ.

2.4 Standard deviation of the mean

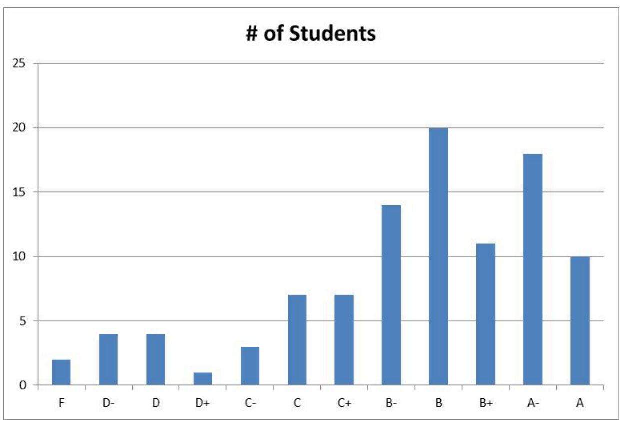
Now you may wonder, does the mean value get better as we take more measurements? That is, do we become more sure about where we are pointing if we throw more darts and include these many darts' locations in our average? I think you will see from our previous reasoning that this is the case. The more trials of an experiment that we take, the closer our mean value is to the " truth" value we are measuring. Since this is the case, shouldn't the uncertainty go down as we perform more trials?

The answer is yes. We won't derive this in our class. But the estimate of the uncertainty should be given by

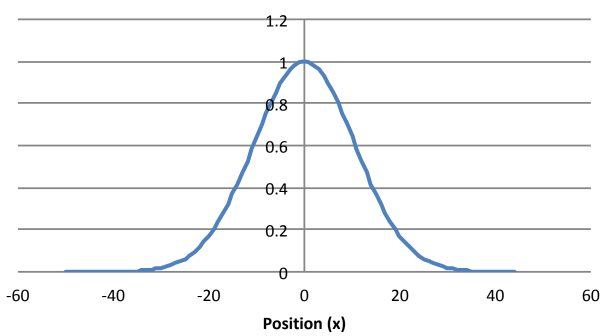
$$\sigma_{\bar{x}} = \frac{\sigma_x}{\sqrt{N}}$$

where σ_x is our standard deviation in our x-position values and N is the number of trials we took. The more trials that go into our average, the lower our uncertainty estimate. The value $\sigma_{\bar{x}}$ is called the *standard deviation of the mean.*

Notice that in some of our grade graphs, the most common score was not a C. Here is an example:



As students, this makes us all happier, but for our error analysis this causes a problem. The error analysis we have talked about so far assumes that our errors are distributed in a very uniform way. If I go back to this graph



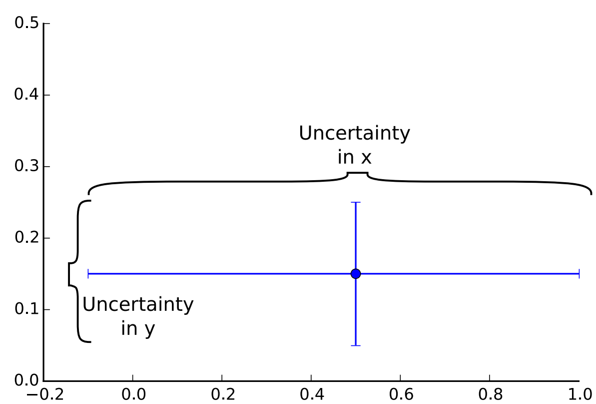
we can see that there are as many darts that landed to the left as there are to the right. This distribution of errors is called the *normal distribution.* Usually our errors in our labs will be normally distributed. That makes all the math we talked about work. But what if they are not, like our grade example? Well, that is a great topic for PH336. So for now we will just assume a normal distribution. But we can check to see how non-normal our data is. We can find the *mode* which is the value that occurs most frequently. For our grade distribution above it would be a *B*.

We can also find the place where half of the trials landed on one side and half on the other. This is called the *median* point. We will calculate both in our lab today. If we have a normal distribution, the average, median, and the mode will all be the same. If this is not the case, then we may worry a little about our error estimate–it may be too small.

2.5 Graphical reporting of the mean (expected value) and standard deviation (uncertainty)

We now have a new view of measurement based on statistics. The mean value is the value that we will say is our measurement. We call this the *expected value*. The standard deviation is the representation of our uncertainty. We can plot this in a way that communicates both at once. If we take our eight data points that we started with earlier, we know the mean , 0.5cm, and we can find the standard deviation of the mean to be 0.6cm. We plot this by making a dot or diamond or some larger point indicator. Then we make a line through the point with little ends that show the size of the uncertainty. The result looks like this:[14]

[14] Most plotting programs will allow you to add error bars to your graphs.



Of course, we could have y -direction error bars as well. These would be vertical, and there is no reason the y-error would be the same as the x-error. We may encounter such situations in future labs.

Chapter 3

Measurement and Uncertainty II

Python You Should Know for This Chapter

- What these terms mean, and how to work with them: *string, int, float*
- How to import the `matplotlib` plotting library and its basic plotting commands.

Recommended reading: Introduction to Scientific Computing in Python, by Nelson and Zachreson; All of chapter 7, Basic Plotting

Questions that you should be able to answer by the end of this chapter.

1. What is $\frac{\partial}{\partial x}\frac{x^2}{yz}$?
2. What is $\frac{\partial}{\partial t}\frac{5}{t^2}$?
3. You want to find out how many moles, N, of an ideal gas are in a container. You can measure its volume, V, temperature, T, and pressure, P. You also know that those quantities are related to the number of moles through the ideal gas law:
$$PV = NRT$$
where R is the universal gas constant and has negligible uncertainty. Write out a symbolic equation to calculate the number of moles of gas in the container and a separate equation for the uncertainty in your calculation.

By now, you should be far enough in your calculus class to know what a derivative is. If not, the most important thing you need to know is that it's a mathematical method to calculate the slope of any function at any point on that function.

3.1 The Power Rule

The most commonly used derivative rule[15] is the power rule[16]:

$$\frac{d}{dx}\left(ax^n\right) = anx^{n-1}$$

[15] Derivatives actually have several different rules that you can memorize: chain rule, quotient rule, trig rules, and many more that you will learn (or should have learned) in your calculus class. Every single one of them comes from taking the limit of $\frac{y_2-y_1}{x_2-x_1}$ as the distance between x_1 and x_2 goes to zero.

[16] Most of what we do in this class will use the power rule, just ask for help if you run into something odd like an arctan function.

that is, if I have a constant, a, times x^n the slope of this curve is the constant, a, times the power, n, times x to the $n-1$ power.

Let's take an example. What is the slope of the function $y = 5x^3$?

$$\frac{d}{dx}\left(5x^3\right) = (5)(3)\,x^{3-1} = 15x^2$$

How about finding the slope of $y = 7x^2 - 2x + 1$[17]

$$\frac{d}{dx}\left(7x^2 - 2x + 1\right) = (7)(2)\,x^1 - (2)(1)x^0 + 0$$

The last term illustrates that the slope of a constant is zero. That makes sense. Constants don't change. So the change in y just due to the last term (1) should be zero. We also remember $x^0 = 1$. So we are left with

$$\frac{d}{dx}\left(7x^2 - 2x + 1\right) = 14x - 2$$

There are many functions where you can use the power rule, but where it is not readily apparent. For example, when x is in the denominator[18]:

$$\frac{d}{dx}\frac{5}{x^3} = \frac{d}{dx}5x^{-3} = (-3) * 5x^{-4} = -\frac{15}{x^4}$$

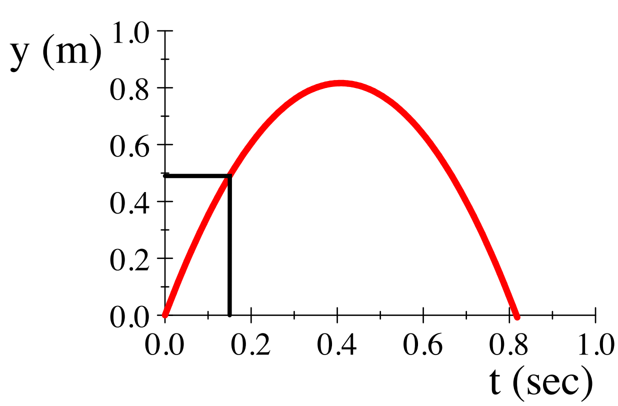
or square roots:

$$\frac{d}{dx}\left(\sqrt{x}\right) = \frac{d}{dx}\left(x^{\frac{1}{2}}\right) = \frac{1}{2}x^{-\frac{1}{2}} = \frac{1}{2}\frac{1}{\sqrt{x}}$$

3.2 How Derivatives Relate to Uncertainty

In Lab 1, we discussed Standard Error Propagation[19] and how you needed to find partial derivatives[20] in order to calculate the uncertainty in our calculated values.

Here's an example of why. Suppose you shoot a ball up into the air and want to predict how high it will be 0.15 seconds after you shoot it. Kinematics predicts that the ball's y postion vs. time graph should look like this



[17] Notice how you can handle each additive term separately.

[18] Remember that $1/x^n = x^{-n}$

[19] If you used the algebraic error propagation methods, you should go back and read the Standard Error Propagation section of that Lab.

[20] Remember: a partial derivative is just a regular derivative where you treat anything except the variable that you are taking the derivative with respect to as a constant. For example,

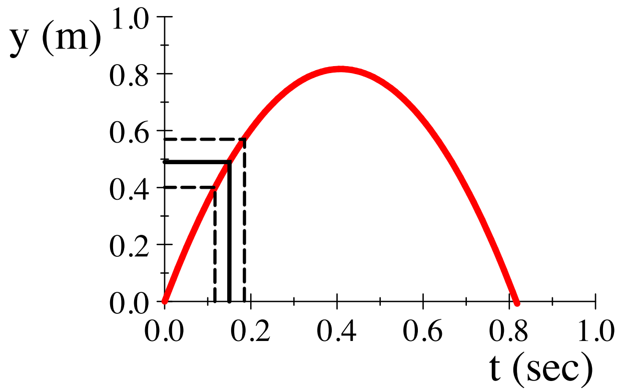
$$\frac{\partial}{\partial x}y^3x^2 = 2y^3x$$

and

$$\frac{\partial}{\partial y}y^3x^2 = 3y^2x^2$$

We use ∂ instead of d to denote a partial derivative.

and that the ball should be at a height of 0.5 meters after 0.15 seconds. However, with the stopwatch we use, the best measurement we can get is 0.15 ± 0.05 seconds. Adding the time uncertainty to the graph



shows that our predicted height could be anywhere from 0.4 m to 0.6 m.

Notice that the line segment between our two dotted lines is roughly linear, and we could approximate it with this equation[21]:

$$y = \left.\frac{\partial y}{\partial t}\right|_{t=t_0} t + b = m_{t_0} t + b$$

[21] Note: $\left.\frac{\partial y}{\partial t}\right|_{t=t_0}$ is the mathematical way to write: the derivative of our function y with respect to t evaluated at t_0.

where $\left.\frac{\partial y}{\partial t}\right|_{t=t_0} = m_{t_0}$ and represents the slope of our line at $t_0 = 0.15$ seconds (where it is crossed by the solid black line).

With a little bit of algebra, we can use this equation to estimate how far away the value at our high point (y_h, where $t_h = 0.2$ seconds) would be from our calculated point, y_h:

$$\begin{aligned} y_h = &\quad m_{t_0} t_h &\quad +b \\ = &\quad m_{t_0}(t_0 + \delta t) &\quad +b \\ = &\quad m_{t_0} \delta t &\quad m_{t_0} t_0 + b \\ = &\quad \delta y &\quad +y_0 \end{aligned}$$

This equation shows us that our uncertainty in y is our slope at our mean value in t (m_{t_0}), multiplied by our uncertainty in time, δt.

Using $\delta y = m_{t_0} \delta t$ is roughly equivalent to using the high/low method from Lab 1. Using a high/low method assumes that any time between 0.1 and 0.2 seconds is equally likely.

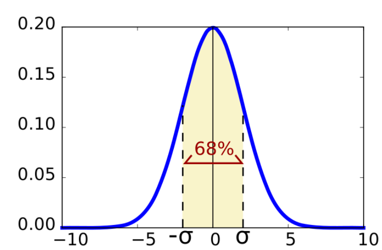


Figure 3.1 A normal distribution. The region within one standard deviation of the mean is highlighted.

But in Lab 2, you learned about normal (also called Gaussian) distributions (see Fig. 3.1 and saw that if a measured time is 0.15 ± 0.05 seconds, you are more likely to measure a time of 0.16 seconds than you are to measure a time of 0.2 seconds.

The rules for standard error propagation[22] use a statistical principle called the *central limit theorem* to take this variance in probability into account. This derivation is beyond the scope of this lab manual, but the result is the equation

[22] Standard error propagation also assumes that your measurements are independent, e.g. how you measure time with a stop watch does not affect how you measured distance with a meter stick. It is generally a safe assumption to make.

from Lab 1:

$$\big(\delta f(x, y, z, ...)\big)^2 = \left(\frac{\partial f}{\partial x}\right)^2 (\delta x)^2 + \left(\frac{\partial f}{\partial y}\right)^2 (\delta y)^2 + + \left(\frac{\partial f}{\partial z}\right)^2 (\delta z)^2 + ...$$

Which you will often see written in summation notation:

$$\delta f = \sqrt{\sum_{i=1}^{N} \left(\frac{\partial f}{\partial x_i}\right)^2 \delta x_i^2}$$

Chapter 4

Experimental Design I: Harmonic Oscillators (masses and springs)

Python You Should Know for This Chapter

- How to create a user-defined function, as well as pass information to that function and get information back out.
- How to import user defined functions.
- How to do math with arrays.

Recommended reading: Introduction to Scientific Computing in Python, by Nelson and Zachreson; Section 4.3, all of chapter 5, and if you have time, chapter 12.

Questions that you should be able to answer by the end of this chapter.

1. A mass oscillating up and down on a spring should have a period T of $T = 2\pi\sqrt{m/k}$ where m is the mass of the mass and k is the stiffness of the spring. In your experiment, the dependent variable is T, and your independent variable is m. What is the proper linearized version of the period equation?
2. Here is a simple Python function:

   ```
   def switch(a,b):
       return b,a
   ```

   How would I use it to switch the values stored in variables `x` and `y`?
3. Last week you calculated the acceleration due to gravity of a tennis ball by timing how long it took to fall. How do the experimental design steps in this reading apply to your process?

4.1 Getting better accuracy by fitting data

In the last lab, we used the mean and standard deviation to find a measurement and its error. We then used our measurements and error propagation to calculate g and the error in our calculation.

What if you wanted better accuracy? One option would be to just repeat the same measurement many different times in hopes of getting a better mean value. However, if the person doing the timing always ends the timer a little too soon, or your scale actually reads 5 kg when it should read zero, repeating the measurement won't help you gain any more accuracy.

But if you change how your measuring a little bit each time, by dropping the ball from a different height, or changing the length of your pendulum string, you can get a better calculated value, even if your measurement technique is off. [23]

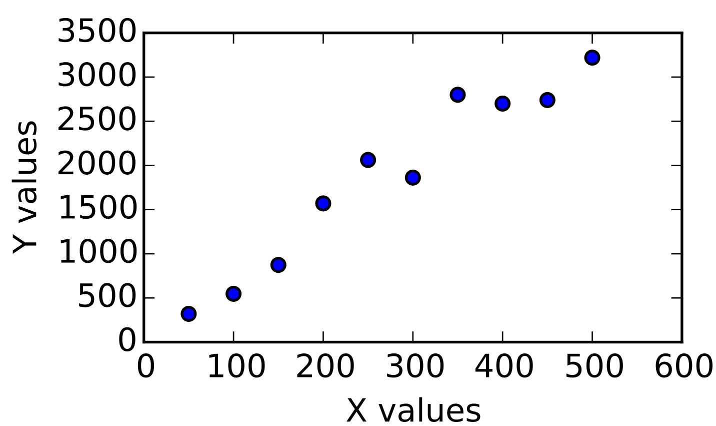
[23] As long as it is consistently wrong. If your scale is 5 grams off when you weight something that is 20 grams, and 50 grams off when you weight something at 100 g, you need a new scale.

This section will show you how.

4.1.1 Linear Least Squares

Linear least squares is the most common type of data fitting (other than just averaging) because it is fast, easy (once you get the hang of it), and it has a known solution. (It's a plug and chug formula) Many types of fits just have a computer try a bunch of values and see which one fits the best. Sometimes the computer will miss the actual best fit for something that is just better than the choices around it, and even if the computer does find the best fit, it will take much longer to get there.

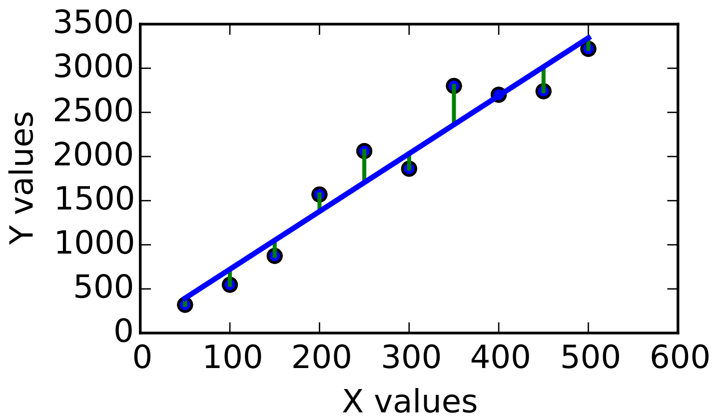
Here is the basics of how least squares fitting works. Imagine that you've taken the data shown below:



The data wiggles all over the place, but it is easy to see that it generally follows a line. Therefore, we'd predict that our data should match:

$$y = mx + b$$

But there is no single line that will go through every single data point. There's always a little bit of error, often written as χ, and it represents the little green lines in the figure below:



My data actually matches this equation:

$$y_i = mx_i + b + \chi_i$$

Where y_i is the y value corresponding to each individual x value, x_i. χ_i gives how far away each data point is from the line. We can solve for how far away from the line each data point is:

$$\chi_i = y_i - b - mx_i$$

and come up with a function that gives us a total error:

$$E_{tot} = \sum_i^N \chi_i^2 = \sum_i^N (y_i - b - mx_i)^2$$

which is just the sum of how far off every single data point is from the line. We use χ_i^2 as an easy way to get the absolute value of each error. We really only care about how far each data point is from the line, not whether it is above or below the line.

Using calculus, we can find the slope and intercept of the line that will minimize the our total error. To minimize, we just take the derivative of our error function with respect to the thing we want to minimize, and set it equal to zero:

$$\frac{\partial E_{tot}}{\partial m} = 0; \ \frac{\partial E_{tot}}{\partial b} = 0$$

If you do those derivatives, and use the two equations to solve for m and b, you get this:

$$m = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2}$$

$$b = \langle y \rangle - m \langle x \rangle$$

where m and b are the slope and intercept of the line that gets closest to all of the data points. The $\langle\rangle$ symbols mean the average of the thing inside. $\langle x\rangle$ is the average of all the xs. To calculate $\langle xy\rangle$ you'd multiply each x value by its corresponding y value, then take the average. Using error propagation, you can find the errors in you fit. The error in the slope is:

$$\sigma_m = \frac{\sigma_y}{\sqrt{N\left(\langle x^2\rangle - \langle x\rangle^2\right)}}$$

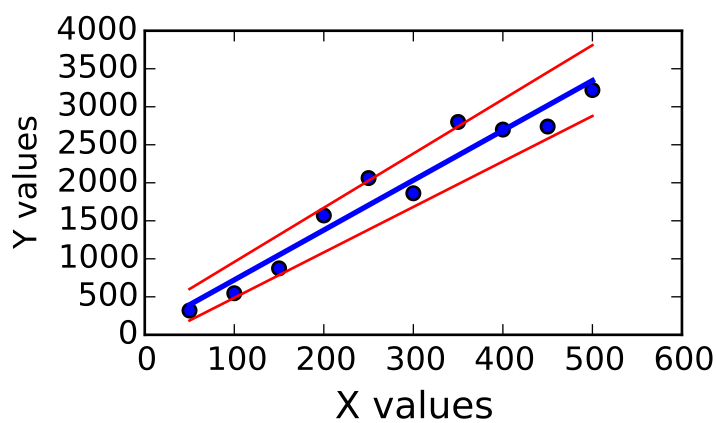
and the error in the intercept is:

$$\sigma_b = \sigma_y\sqrt{\frac{\langle x^2\rangle}{N\left(\langle x^2\rangle - \langle x\rangle^2\right)}}$$

σ_y represents the average error of each y value, and is calculated using:

$$\sigma_y = \sqrt{\frac{1}{N-2}\sum_i^N\left(y_i - b - mx_i\right)^2}$$

For reference, here is a figure with the best fit line for to the data. The red lines mark the high/low values given by the error in the slope:



Here's an example of a Python function that takes in x and y values and returns the slope and intercept of the best fit line[24]. It would be very go practice to read through it and see if you can figure out what each line does.

[24] It will be up to you to add a part that calculates their errors.

```
def linear_least_squares(x,y):
    #Import numpy
    import numpy as np
    #Get the number of data points
    N=len(x)
```

```
    #Make sure x and y are numpy arrays to make array math easy
    x=np.asarray(x)
    y=np.asarray(y)

    #Calculate the average values needed to find the slope and
    intercept
    xbar=np.mean(x) #Average Value of the xdata
    ybar=np.mean(y) #Average value of the y data
    xbar2=np.mean(x**2) #Average value of the xdata squared
    xybar=np.mean(x*y) #Average value of xdata*ydata



    #Use the linear least squares formula to calculate
    #the slope and the intercept of the best fit line
    slope=(xybar-xbar*ybar)/(xbar2-xbar**2)
    intercept=ybar-slope*xbar

    return slope, intercept
```

Once coded, you can use the function like this:

```
x_data = [10,20,30,40,50]
y_data = [20,40,60,80,100]

#Notice that the input/output variable names do not have to match
#what you called them in the function definition.
m,b = linear_least_squares(x_data,y_data)
#Now the slope of my fit is stored in m
#and the intercept is stored in b
```

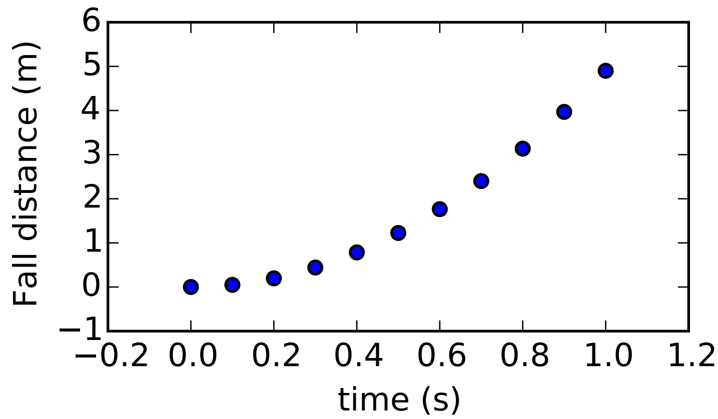
4.2 Linearizing equations

One of the major downsides to linear least squares fitting is that it only works on lines. While most of the relationships in physics aren't readily linear, we can tweak them to make them linear. We call this “linearizing the equation".

Here's an example. We take a video of a ball, dropped from rest, and measure how far it has fallen in each frame. In this instance, we'd predict that the relationship between the distance fallen, y and the time t would be:

$$y = \frac{1}{2} g t^2$$

where g is the acceleration due to gravity. The plot of y vs t is looks like this:

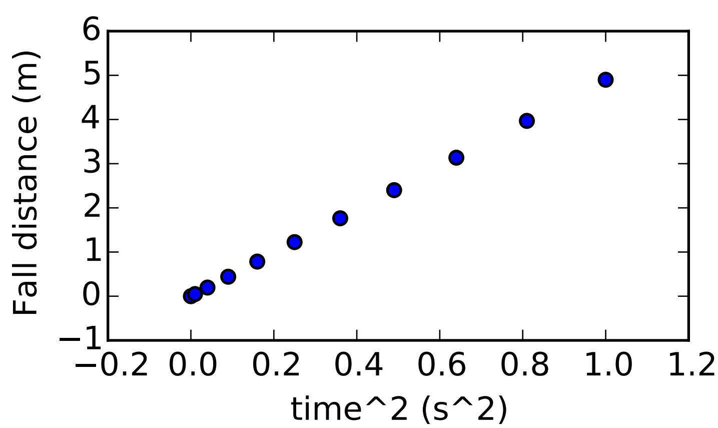


That is definitely not linear, but we can tweak it so that it is. Here are the steps:

1. First isolate the dependent variable, or the thing you measured. In this case, that would be our distance fallen, y.

2. See if it looks like a line. Here's how you can tell. In $y = mx + b$ we have three parts that determine y: m and b are constant, x is something that changes. In our example, we change t as we advance frame by frame, that's called our independent variable.

So, let's check our equation. Do we have a constant that multiplies some function of our independent variable? In this case, we have $\frac{1}{2}a$ as a constant, and t^2 is a function of our independent variable; therefore, we make our slope $m = \frac{1}{2}a$ and our x values $x = t^2$. Since we don't have anything that we are adding, $b = 0$. To add an extra check to our work, here is a plot of y vs t^2:



which is very much linear.

As another example, suppose we wanted to find the density of copper. To that end, we melted some copper and let it drip out a small hole, and let the drops fall as they cooled. A drop will form an almost perfect sphere. (This is actually how they make ball bearings.) By changing the size of the whole, you change how much mass the droplet can accumulate before it falls. (Making mass our independent variable.) As we vary the size of the hole, we record the mass and radius of the copper ball bearings. Density, ρ, obeys the following relationship:

$$\rho = \frac{m}{V} = \frac{m}{(4/3)\pi r^3}$$

If we solve for our dependent variable, r, we get:

$$r^3 = \frac{m}{(4/3)\pi\rho}$$

Notice that I left r^3, rather than solving completely for r. You only need to isolate your dependent variable, not solve for it. Our only thing that is changing on the right hand side of our equation is m, so our y axis should be r^3, our x axis will be m, and our slope will be $1/\left[(4/3)\,\pi\rho\right]$.

Now, we could solve completely for r and get a valid result. If we use:

$$r = \left[\frac{m}{(4/3)\pi\rho}\right]^{1/3}$$

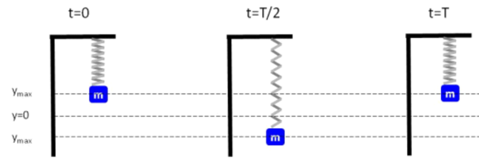
then r would be our y values, $m^{1/3}$ would be our x values, and $\left[(4/3)\,\pi\rho\right]^{-1/3}$ would be our slope.

4.3 Experimental Design

We are going to tackle the subject of experimental design today, but it helps to have an experiment in mind. You are learning or have learned Hook's law for springs in your PH121 class. You understand that when we attach a mass to a spring and stretch or compress a spring we have a force

$$F_s = -kx$$

on the mass, where by k we mean the spring stiffness constant, and by x we mean the displacement from the equilibrium position of the mass-spring system. We can make such a system oscillate. This is really a PH123 problem, but let's pretend that we are scientists in Newton's day and we don't know much about oscillation (because most of us don't yet). We wish to find out more. We know that if we build a mass-spring system we can get oscillation and we define the time it takes for the mass to travel through one full oscillation (so the mass, say, starts from the highest point and it returns to it's highest point) as the *period* of oscillation and abbreviate it with the letter T.



Let's further pretend that you have read Hook's work and from this work have reason to believe that period might be proportional to the square root of the mass.

$$T \propto \sqrt{m}$$

You want to verify this report and build your own model for the period of oscillation of a mass-spring system. You will use this model to make predictions, and by doing so, you will see how well your model works. This is what we want to investigate, now let's see how to design our experiment.

4.4 Designing an experiment

One of our objectives in this course is to learn how to design an experiment so that it will be successful.

Back in grade-school, an experiment was any science related activity (the proverbial building of a volcano model was considered an experiment). But for a scientist, an experiment is a specific thing. It is the testing of a hypothesis. You must test a hypothesis with care because the entire foundation of science depends on the integrity of how we do this testing.

In this lab I will give you a hypothesis to test (the period of oscillation for a mass-spring system depends directly on the square root of the mass). The following steps will help your experiment be successful.

1. **Identify the system to be examined.** In our case it is a mass-spring system. We should identify all the *inputs* to the system. For example, we know there is a mass, m, and you have heard about a spring constant, k. There is also the force due to gravity and tension on a spring. These are inputs. You should describe your system in your lab notebook and list the inputs. These inputs are the things you can possibly change in the design of your experiment.

2. **Identify the model to be tested.** The word "model" means our mental picture of how something works. As physicists, we would prefer to express a model in a mathematical equation. For example, we have a model of how force depends on acceleration. The bigger the acceleration, the more the force. But our model also includes mass. The larger the mass, the larger the force needed to create the same acceleration. This mental model can be expressed in an equation

$$F = ma$$

It is valuable to use both the word description of the model as well as our mathematical representation. In our case today, our mental model is that period of oscillation for a mass-spring system depends directly on the square root of the mass. Think about what this means. If I increase the mass, the period should get longer. But if I double the mass, the period won't double. This is a mental model that allows us to do predictions of behavior. In physics it is almost required to reduce this model to an algebraic equation that can be used to calculate a prediction and an uncertainty on that prediction. For today's experiment that equation is

$$T = C\sqrt{m}$$

where C is a factor that does not depend on mass, but for your experiment that your group designs later in this course you will have to come up with your own mathematical expression of your model. Record your model and the model equation in your lab notebook.

3. **Plan how you will know if you are successful in your experiment.** You are testing a hypothesis, and you are much more likely to succeed in your test if you plan what that success would look like. One way to do this is to plan how you will communicate your results. It is a great idea to think of what graph you will make at the end of the experiment to communicate whether your model works (or not). In today's experiment, a graph of T vs. m or even T vs.$\sqrt{m}$ might be useful along with a curve fit. Notice I am suggesting you plan this before you perform the experiment. I am not suggesting you decide on what the results will be, only how you will report them. This focuses your attention on deciding what measurement you will make. In our case today it is hard to plot T vs. m if you don't measure T and m, planning the graph in advance helps you plan the experiment. Mock up your graph or figure in your lab notebook. Give axis titles and even units (but of course no data yet).

4. **Plan your analysis.** Symbolically layout and solve any needed equations. That way you will know exactly what measurements you need to make, and will not have to try to recreate the experiment when you are analyzing your results.

   If possible, rectify (linearize) your equation. It would be good to be able to use a curve fit to analyze our data. The strongest and most reliable curve fits are straight line fits where the fit equation is something like

$$y = mx + b$$

   So if at all possible, we would like to reduce our equation to the equation of a straight line. In today's lab we can do this. We call this *linearizing the equation.* If we can't find a way to linearize the equation, we at least need to render our equation into a form that we can use to predict the outcome of our experiment. Record your new equation in your lab notebook.

5. **Choose ranges of the variables.** For today's experiment we might have several, but m and k are principal variables. It should be clear when you see the spring that putting a thousand kilograms of mass on the spring would be a bad idea. but how much mass is right? What will give you good results in testing your theory? Hooks law is not valid for all m and k (if you doubt this, think of your Christmas slinky after your brother got to it; it never looked the same again!). What values of m are best for performing the experiment? An error analysis based on your equation is invaluable in making this decision. Changes in mass that produce a change in T that is smaller than the uncertainty in T will not be noticeable. So taking measurements for such small mass changes would be a waste of time and effort. We would like to avoid this. Changes that are likely to break the equipment are also not desirable. And of course you want to plan this before you do the experiment and find that you did not get good data, and therefore must repeat all your work! Record your variable ranges in your lab notebook. As you perform the experiment note any deviations from this plan.

6. **Plan the experimental procedure.** As a group talk your way through the experiment. You might find yourselves saying something like " Then you take the stopwatch and measure the period." and you realize that you did not get a stop watch. For your experiment that you design, you need to find out in advance if we have the equipment you need. So get in the habit of working through the procedure in advance to see if you have forgotten anything. Record your planned procedure in your lab notebook. As you perform the experiment, note any deviations from the plan and the reason for the deviation. Deviations are fine, just make sure you record them.

7. **Perform the experiment and report on it in your lab notebook.** This involves all the things we have been including in our lab notebooks to date:

    - Describing the goal for the work. (This is probably already done in your plan)
    - Give predictive equations and uncertainties for the predictions based on the physical law. (This is probably already done in your plan)
    - Give your procedure you actually followed, recording what you really did as you do it. This will probably not be just a restatement of the plan because things will change as you go. Record the equipment used and settings, values, etc. for that equipment. Did you learn how to use any new equipment? What did you learn that you want to recall later (say, when taking the final, or when you are a professional and need to use a similar piece of equipment five years from now).
    - Record the data you used. If you have a large set of values, you can place them in a file, and then record the file name and location in your

lab notebook. Make sure this is a file location that does not change (emailing the data to yourself is still not a good plan).

- Give a record of the analysis you performed. You planned this above, now record what you actually did
- Give a brief statement of your results and their associated uncertainties.
- Draw conclusions: Do your results support the theory? Why or why not? What else did you learn along the way that you want to record. (This is where we may compare the percent error to our relative uncertainty).

Chapter 5

Experimental Design II: Conservation of Energy

At the end of this class you will have two weeks to carry out an experiment of your own design. Before doing the experiment, you will need to submit a written proposal and have it approved by your teacher (similar to getting a grant proposal funded before beginning research).

5.1 Proposal writing

A Proposal is a document that is intended to persuade someone (your professor, funding agency, yourself, etc.) that you should be given the resources and support to perform the experiment. For our class, the proposal consists of the following parts:

1. Statement of the experimental problem
2. Procedures and anticipated difficulties
3. Proposed analysis and expected results
4. Preliminary List of equipment needed

Note that most of the steps involved in planning an experiment are contained in these for parts of the proposal. Each of the steps is explained in more detail below.

5.1.1 Statement of the experimental problem

This is a physics class, so our experiment should be a physics experiment. The job of an experimental physicist is to test physics models. So your statement of the experimental problem should include what model you are testing and a brief, high level, overview of what you plan to do to test this model.

5.1.2 Procedures and anticipated difficulties

Hopefully, your reader will be so excited by the thought of you solving your experimental problem that he or she will want to know the details of what you plan to do. You should describe in some detail what you are planning. If there are hard parts of the procedure, tell how you plan to get through them. This is essentially steps 1-6 of our experimental design strategy.

5.1.3 Proposed analysis and expected results

You might think this is unfair, how are you supposed to know what analysis will be needed and what the results should be until you take the data? But really you both can, and should make a good plan for your data analysis and figure out what your expected results should be before you start taking data. After all, you have a model you are testing. You can encapsulate that model into a predictive equation for your experiment. Then you can use that predictive equation to obtain predicted results and uncertainties. Using this, you can design your experimental apparatus by putting in the numbers from your experimental design and seeing what the outcome should be. You can see if there is a chance that your experiment will measure what you want with the equipment you have (this is where our differential form of error calculation comes in).

If you don't do this, you don't know what equipment you will need or how sensitive that equipment needs to be. If you are trying to measure the size of your text book, an odometer that only measures in whole miles may not be the best choice of equipment. This might be obvious, but depending on how well you need to measure your text book, a ruler may not work either. You don't know until you have an estimate of the uncertainty. So to know what you need, do the calculations in advance with your range of inputs as the values you take for the prediction.

Of course this means you must include a predictive calculation of the uncertainty. Uncertainty in your result is governed by the uncertainty inherent in the measurements you will take. The uncertainty calculation tells you what sensitivity you will need in your measurement devices. In our text book case, you could see immediately that you need a different apparatus than the odometer. You might also find our ruler to be problematic depending on what precision you need.

I remember a time in my career when the US Naval Research Labs asked us to build a microwave radiometer to measure the sea wind direction from space. We spent some time and our predicted analysis and uncertainty said that it would be a very expensive instrument to be able to successfully measure the wind direction—it would take more money than they were offering. NRL disagreed and built the device themselves at less cost, but to lesser specifications with much greater uncertainties. Then they spent a billion (yes, a billion with a "b") dollars to launch the device into space. The device was a total failure. The uncertainty was so big that the data was totally useless. We want to find out whether our experiment will work before we risk our grades (or a billion dollars) on it. So we will do the prediction ahead of taking the data.

You should do all of this symbolically if you can, numerically if you must, but almost never by hand giving single value results. Some measurements will come back poorer than you anticipated, or some piece of equipment will be unavailable. You don't want to have to redo all your calculations from scratch each time this happens. For example, in the event of an equipment problem, your analysis tells you if another piece of equipment is sufficiently sensitive, or if you need to find an exact replacement. When I perform an analysis like this, I try for a symbolic equation for uncertainty. I like to program these equations into Scientific Workplace, or Maple, or SAGE, or MathCAD, or Mathmatica or whatever symbolic math processor I have. Then, as actual measurements change, I instantly get new predictions. In the absence of a symbolic package, a python script or a spreadsheet program will do fine (and we have Excel on our computers). A numerical program also is quick and easy to re-run with new numbers when no symbolic answer is found.

5.1.4 Preliminary List of equipment needed

Once you have done your analysis, you are ready to list the equipment you need and the sensitivity of the equipment you need (that is, list the uncertainties you need to achieve). Final approval of the project and the ultimate success of your experiment depends on the equipment you choose or are granted. You want to do a good job analyzing so you know what you need, and a good job describing the experiment so you are likely to have the equipment you want available when you start.

5.2 Performing the experiment

Once your proposal is accepted, I will provide you with the equipment we have agreed upon from your proposal. You will have three weeks to perform your experimentation. I will be available for advice and to watch for problems or safety issues. But you and your team will perform the experiment. **You will want to keep good notes in your lab notebook.** You will likely have to change your procedure after you start because of problems. Take careful note of what was actually done, and what your measurements were. Give the reason for the change. Note any unusual things that happen. **Carefully record what you do.**

5.3 Written report

The written report is designed to match a normal format for an applied physics article in a journal like *Applied Optics* or the *IEEE Transactions* journals. It is useful to know now how I will grade the report later so you can make sure you design in all the parts I will look for. There should be an introduction, description of the procedure, description of the data and results, a description of the analysis, and a conclusion. These sections are described in detail in the following table.

Experimentation is a lot of fun if done right. It can be frustrating and discouraging if not done well. Our goal is to learn how to perform and report on an experiment, so that is what will be graded. If you show something new and interesting, that is just more fun. If you show that your original model was not correct–that is science! If you have done a good job designing and reporting your experiment, a negative result is just as good as a positive result.

Report Rubric

Table 5.1:

Section	50-40 pts	40-30 pts	30-20 pts	20-0 pts
Introduction Answers the question "What is this lab about?"	• Answers the question “what is this lab about?” sufficiently that a person who did not perform the lab would understand • Gives enough background so that the lab report makes sense as a stand-alone document • Tells the reader what your expected outcome is based on theory.	• Answers the question “what is this lab about?” sufficiently that a person who was part of your lab group would understand • Gives enough background so that the lab report makes sense to someone who knows the lab topic well	• Mentions what the lab is about • Gives some background	• It is difficult to tell from the introduction what the lab is about • Little or no background provided
Procedure Answers the question "what did you do?"	• This section answers the question “what did you do?” sufficiently so a non-expert can understand what was done. • Describe the entire procedure, especially indicate any deviations from your plan and explain why those deviations were necessary.	• This section answers the question “what did you do?” sufficiently so your lab partner could understand what was done. • Tells where you deviated from the plan	• Major points of the procedure are listed	• It is difficult to tell what you did from your description
Data Answers the question "what did you measure"	• Each measured value is given with units • Each value is given with a good estimate of uncertainty • Only measured values that are needed are given • The data is presented in a way that is easy for the reader to find and read. (e.g. label graphs and table columns)	• Each measured value is given with units • Each value is given with an estimate of uncertainty • Extra values that were not needed are given	• measured values are given	• It is not clear what you measured
Analysis Answers the question "how did I get from my data to my results?"	• It is clear how you got from your measured values to your results • Major equations are given and discussed. • The method of determining uncertainties is discussed	• It is possible to tell how you got from your measured values to your results • Major equations are given • The method of determining uncertainties is discussed	• It is possible to tell how you got from your measured values to your results • Major equations are given • Method of determining uncertainty is not discussed	• It is not possible to tell how you got from your measured values to your results • Major equations are missing • Method of determining uncertainty is not discussed

Continued on next page

Table 5.1: (Continued)

Section	50-40 pts	40-30 pts	30-20 pts	20-0 pts
Results Gives the results of your analysis	• There is a clear, understandable answer to the question the lab asks. For example, if I ask you how fast a car is going, the result would be a calculated speed, with its calculated uncertainty and units. • Report percent error or percent difference • Report fractional uncertainty	• There is a an answer to the question the lab asks with uncertainty and units • Report percent error • Report fractional uncertainty	• There is a an answer to the question the lab asks • uncertainty and units are missing • Percent error or fractional uncertainty is missing	• There is no clear answer to the question the lab asks • Percent error or fractional uncertainty is missing
Conclusion Answers the question "did the experiment show what was intended?"	• There is a clear discussion of whether the experiment was supported or falsified the theory. • This discussion includes a comparison of the percent error and fractional uncertainty • If there were difficulties, they are discussed here • There is a statement of what you learned from this experiment. Note any problems and how you would resolve them if you were to redo this experiment.	• There is a general discussion of accuracy (often with percent errors quoted) • There is some mention of whether the predictive theory is supported • Problems are noted and how you would resolve them if you were to redo this experiment is discussed.	• There is no comparison of the percent error and fractional uncertainty • There is a statement of what you learned from this experiment.	• There is no outcome of the accuracy of the experiment • There is no comparison of fractional uncertainty and percent error • There is no clear conclusion about the predictive theory • There is little mention of what was learned

Chapter 6

Experimental Design III

There is no introduction for this Lab in order to give you time to work on lab proposals.

Chapter 7

Numerical modeling of Projectile motion

Python You Should Know for This Chapter

- How to construct a `for` loop.

Recommended reading: Introduction to Scientific Computing in Python, by Nelson and Zachreson; All of Chapter 6.

Questions that you should be able to answer by the end of this chapter.

1. At what point(s) should I calculate my acceleration when using Euler's method?
2. How do I find my next position and velocity when using Euler's method? What does "next position" actually mean?

7.1 Review of Kinematics

There are times when we would like to predict the motion of an object, but we would like to make a computer do the hard work involved in solving the equations to get a numeric answer, so we don't have to do it. For example, we might want to calculate the position of a satellite as it orbits. Or we might want to calculate the position of the planets as they orbit the Sun. For simple cases, we might be able to do this by hand, but predicting by hand where all the planets will be on July 4, 2200 might get tedious without a computer.

As physics students, it is good to know how to approach solving problems on a computer.[25]

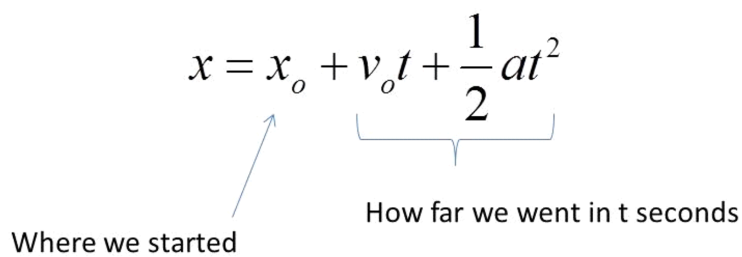
[25] Most calculators qualify as computers now, since they are usually programmable. But we are talking about going beyond the built in functions of calculators or even spreadsheet functions on computers.

Let's start with some simple problems that we know how to do algebraically so we can tell if the computer solution is reasonable by comparing to the exact answer. Then we will tackle things we can't do algebraically.

Consider a ball moving under constant acceleration in one dimension. This is a kinematics problem. So we can use the kinematic equations you have learned (or will very shortly learn) in PH121. Let's remind ourselves of what these are:

$$x = x_o + v_o t + \frac{1}{2} a t^2$$
$$v = v_o + a t$$
$$v^2 = v_o^2 + 2a(x - x_o)$$

The first equation tells us the position, x, as a function of time. This could be position in any direction, up and down, or side to side. If it is a ball being shot into the air, it might be better to write y instead of x. But for now, we will use just x for position in any direction. What this first equation tells us is that, once the ball is moving, we can find the new position of the ball by starting with the initial position of the ball, x_o, and adding to this initial position how far it has gone in t seconds.

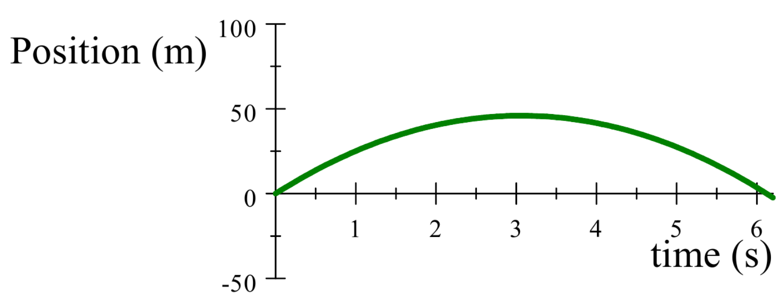


If the ball were going at constant speed, the additional distance would be just $v_o t$. This comes from speed being distance over time.

$$v = \frac{d}{t}$$
$$d = v t$$

But our ball is accelerating, so we have to add in a little more distance because our speed is changing. That is what $\frac{1}{2} a t^2$ does.

Since the position is a function of t^2, we know that the position vs. time graph for our ball will be a parabola.



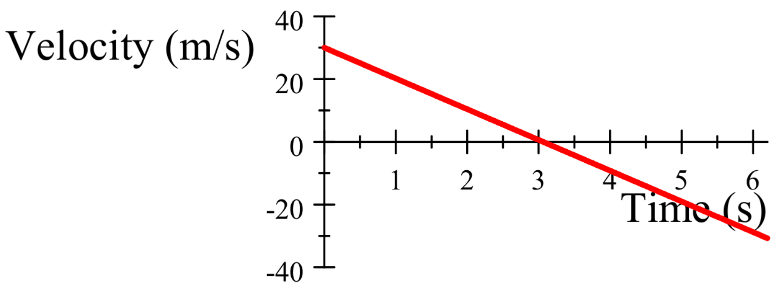
The second and third equations from our kinematic set give the speed of the object. The second is the speed as a function of time. Since for this first problem in programming we will only allow one dimensional motion (at first) we could say this is the velocity of the object and allow negative values to mean going the opposite way of positive values. Because the ball is accelerating, the speed will change. We can see that the velocity changes linearly with time. Think of a straight line

$$y = mx + b$$

our velocity equation is

$$v = at + v_o$$

on a velocity vs. time graph, this has the form of a straight line. So the velocity vs. time graph will be a straight line.



7.1.1 Changing Acceleration

The kinematic equations only work when acceleration is constant. You should have learned in your physics class that you can get around this limitation by using the ending of one acceleration region as a starting point for the next. For example, imagine we have an acceleration that starts at $3m/s^2$, but once a second goes down by one until it hits zero, like so:

t (s)	a (m/s^2)
0	3
1	2
2	1
3	0

Since the acceleration is changing, we can't connect the beginning to the end with just one set of kinematic equations. However, if we knew that our initial velocity and position were both zero, we can find where we are and how fast we are going at one second:

$$x_1 = x_0 + v_0 t + \frac{1}{2}at^2 = 0 + 0 * 1 + \frac{1}{2} * 3 * (1)^2 = 1.5$$

$$v_1 = v_0 + at = 0 + 3 * 1 = 3$$

And then use that information to find out where we are and how fast we are going at 2 seconds:

$$x_2 = x_1 + v_1 t + \frac{1}{2}at^2 = 1.5 + 3 * 1 + \frac{1}{2} * 2 * (1)^2 = 5.5$$

$$v_2 = v_1 + at = 3 + 2 * 1 = 5$$

and so on until we fill out the table:

t (s)	a (m/s^2)	v (m/s)	x (m)
0	3	0	0
1	2	3	1.5
2	1	5	5.5
3	0	5	10.5

7.2 Euler's Method

Euler's method works under this same principle (starting/ending new sets of kinematic equations), but it uses an idea from calculus[26] that we can take an infinitesimally small amount of time for our experiment. We will call this very small amount of time a "time step" and label it Δt. Over this time interval, the acceleration will be very close to constant, which lets us use kinematics. Our velocity equation looks the same:

[26] Not to worry if you are concurrently taking calculus. We will use very little calculus in this reading, so little that you probably won't notice it.

$$v_f = v_i + a\Delta t$$

but our position equation will change a little bit. If Δt is small, Δt^2 will be very very small, so the position question becomes:

$$x_f = x_i + v_i t + \frac{1}{2}at^2 \approx x_i + v_i \Delta t$$

since $\frac{1}{2}a\Delta t^2 \approx 0$[27]. In the limit as $\Delta t \to 0$ our equation will be exact.

[27] If acceleration is large enough that Δt^2 doesn't make this term almost zero, you need to pick a smaller Δt

Euler's method lets us calculate motion even when acceleration changes by following these basic steps:

- set up initial conditions (v,x)
- Repeated section:
  - Calculate my acceleration for my current position and velocity (for example: include a drag force that depends on how fast you are going)
  - Use the position equation and current velocity to find my next position
  - Use the velocity equation and current acceleration to find my next velocity

7.2.1 Euler and Simple Systems with Not-So-Simple Dynamics.

As an example, suppose we throw a ball, but we have air resistance. Your PH121 class only has you throw balls in a vacuum–something that is only fun if you have a space suit. Real balls have air resistance. To model air resistance is really a PH123 problem. So I will just give you a formula for the force due to air resistance here. There is a resistive force

$$F_R = \frac{1}{2} D\rho A v^2$$

that depends on the cross-sectional area of the ball, A, the density of the air, ρ, the speed of the ball, v, and the ball's drag coefficient, D, that contains the effects like surface roughness and shape of the ball. So we have two forces working on the ball now, the force due to gravity, and the resistive force. If we throw the ball straight up, then we only have forces in one dimension. We can use our Newton's second law method to find the acceleration!

$$\Sigma F_x = ma = -F_g + F_R$$

so

$$a = \frac{-F_g + F_R}{m}$$

or

$$\begin{aligned} a &= \frac{-mg + \frac{1}{2} D\rho A v^2}{m} \\ &= -g + \frac{D\rho A v^2}{2m} \end{aligned}$$

Notice that this acceleration changes when v changes. And we know that v does change as the ball goes up. We can't use the kinematic equations at all with a changing acceleration. But we can use our Euler method.

$$\begin{aligned} x(t+\Delta t) &= x(t) + v(t)\Delta t \\ v(t+\Delta t) &= v(t) + a(t)\Delta t \end{aligned}$$

or

$$\begin{aligned} x_{n+1} &= x_n + v_n \Delta t \\ v_{n+1} &= v_n + a_n \Delta t \end{aligned}$$

The only difference is that now our acceleration is not $-g$, but $a = -g + \frac{D\rho A v^2}{2m}$. So instead of[28].

$$\begin{aligned} x_{n+1} &= x_n + v_n \Delta t \\ v_{n+1} &= v_n - g\Delta t \end{aligned}$$

[28] The x_{n+1} and x_n is a standard notation to mean the next x we find (x_{n+1}) compared to our current x (x_n, the x for right now). There are other algorithms that involve more points that might reference the previous (x_{n-1}) point we calculated, for example.

we will have

$$x_{n+1} = x_n + v_n \Delta t$$
$$v_{n+1} = v_n + \left(-g + \frac{D\rho A (v(t))^2}{2m}\right)\Delta t$$

The rest of the method stays the same. In fact, to change physical systems we usually only need to change the acceleration function. This implies that we could build a general Euler solver, and then just modify the acceleration part. That is such a good idea that there are standard notations for this.

7.3 Implementation

You are probably wondering how you would actually make a computer do all of this calculation. In this section let's consider how to write an Euler method code for our lab, then I will comment on versions to use in further studies (later in your career). You don't need to read this section before class. What follows is a tutorial that will guide you step-by-step through the first part of the lab. But if you want to, read on just to get a feel for what we will do.

7.3.1 The Program

This program uses Euler's method to model an object falling straight down, with no air resistance. Try to read through it and understand what it is telling the computer to do. I will break it down piece by piece in the next section.

```
"""
One Dimensional free-fall Euler Code

PH150
Brother Zachreson


This code will calculate the exact solution for a ball in free fall
having been shot straight up using Euler's method.

"""
#Import numerical and plotting packages
import numpy as np
import matplotlib.pyplot as plt


#Initial conditions and physical setup constants
v0=30.0 #Initial velocity in m/s
x0=70 #Initial Postion in m

#Set up the time steps and number of calculations
delta_t = 0.01 #Time step in seconds
```

```
t0=0 #Start time in seconds
t_max = 7.0 #Final time

#Calculate the number of timesteps we need to make
N=(t_max-t0)/delta_t

#Make sure N is an integer
N=int(N)


#Make lists to hold our positions, velocities, and times
x=[x0]
v=[v0]
t=[t0]

#Now perform an Euler's method calculation.
for i in range(N):
    #Find the current acceleration
    a=-9.8

    #Find the next velocity
    v.append(v[i]+a*delta_t)


    #Find the next position
    x.append(x[i]+v[i]*delta_t)

    #increment our time
    t.append(t[i]+delta_t)

#End of Euler loop =========================================
#Plot the results
plt.plot(t,x,linewidth=2,color='red')
plt.xlabel('time (s)')
plt.ylabel('Position (m)')
plt.title('Position vs. time for a ball in free fall')
```

7.3.2 The Program - Broken down

This section will break the program down piece by piece. Here's the first section:

```
"""
One Dimensional free-fall Euler Code

PH150
Brother Zachreson


This code will calculate the exact solution for a ball in free fall
```

```
having been shot straight up using Euler's method.

"""
#Import numerical and plotting packages
import numpy as np
import matplotlib.pyplot as plt
```

Everything inside the triple quotes `"""` counts as one long comment. If you ever need to add comments that will take more than one line, you can start and end it with `"""`. It's a great way to leave yourself notes about what each program does. Below the introductory comment, I import Numpy and Matplotlib since I know that I'll be doing number crunching and plotting later on.

This next part sets up all of our constants and inputs. It's a good idea to put it at the beginning of the program so that they are easy to change.

```
#Initial conditions and physical setup constants
v0=30.0 #Initial velocity in m/s
x0=70 #Initial Postion in m

#Set up the time steps and number of calculations
delta_t = 0.01 #Time step in seconds
t0=0 #Start time in seconds
t_max = 7.0 #Final time
```

The part with `x0` and `v0` is where I set the starting position and velocity. `delta_t` is how far forward in time we will go in each calculation. Remember, Euler's method works by assuming that acceleration is roughly constant over short time intervals. `delta_t` sets that time interval.

Once we know our initial values, we need to tell the computer how many calculations to do. Here's the piece of code that does that:

```
#Calculate the number of timesteps we need to make
N=(t_max-t0)/delta_t

#Make sure N is an integer
N=int(N)
```

The above code calculates the number of timesteps needed, then saves it in the variable `N`.

Python keeps two types of numbers: "Floats" - numbers with decimal points and "Integers" - whole numbers. For example: 2.0 is a float, 2 is an integer. Even though they have the same value, Python sees them as different things. You can only get an item from a list using an integer. `myList[2]` works, but `myList[2.0]` won't. The command `N=int(N)` takes whatever number `N` is and converts it to an integer. The `int` command truncates instead of rounding, so `int(2.9)` becomes `2`, not `3`.

```
#Make lists to hold our positions, velocities, and times
```

```
x=[x0]
v=[v0]
t=[t0]
```

Notice that all three of these are in square brackets, but there aren't any on on the variables I set above (look back to where I set `x0`). Putting square brackets around something tells Python that it is a list. Since we?re going to be adding more x,v, and t values, we need to warn Python that these are going to be lists. Having the square brackets is essential on these. Whereas, `x0` is just a single value and will always be a single value, so it didn?t need square brackets.

This next section of code is what we call a loop:

```
#Now perform an Euler's method calculation.
for i in range(N):
    #Find the current acceleration
    a=-9.8

    #Find the next velocity
    v.append(v[i]+a*delta_t)


    #Find the next position
    x.append(x[i]+v[i]*delta_t)

    #increment our time
    t.append(t[i]+delta_t)

#End of Euler loop =========================================
```

Breaking down the for loop

For most of you, a loop will be a new idea. Loops are very helpful when you need to tell a computer to do something over and over again. Here's the general structure:

```
for <Thing that Changes>:
     Stuff the computer does to/with the thing that changes

Not part of the loop
```

Let's start by looking at the `Thing that Changes`. As an example, here is a shopping list: bananas, apples, bread, milk. If I wrote this program:

```
shopping_list = ['bananas','apples','bread','milk']
for food in shopping_list:
    print(food)

print(shopping_list)
```

When I run this program, I first load my shopping list. The for food in shopping_list tells Python to look at each thing in my list one by one, and call it food. Therefore, this program would first print bananas, then when it sees that print(shopping_list) isn't indented, it knows to go back to the for line, but changes food to the next thing on the list: apples. So, after printing bananas, then apples, it would print bread, then milk. Since the list ends at milk, the computer would then know to continue on and print the whole shopping list at once. (Thanks to the print(shopping_list) command)

Now let's look at our for statement: i in range(N). The range(N) command makes a list of integers from zero to N-1. If N was five, range(N) would be |[0,1,2,3,4]|. Therefore, the first time we go through the for loop i=0, the second time i=1, and so on until we reach N-1.

Now let's look at the guts of the loop:

```
    #Find the current acceleration
    a=-9.8

    #Find the next velocity
    v.append(v[i]+a*delta_t)


    #Find the next position
    x.append(x[i]+v[i]*delta_t)

    #increment our time
    t.append(t[i]+delta_t)
```

This is where we actually implement Euler's method. First we find the acceleration. (-9.8 in this case) Then, we have this command:

```
v.append(v[i]+a*delta_t)
```

The append command takes whatever is in the parenthesis and adds it to the bottom of of whichever list is before the . . In this case, v.append(v[i]+a*delta_t) just adds v[i]+a*delta_t to the bottom of the v list.

We know from kinematics that $v_f = v_i + a\Delta t$, so the program just uses v[i] as v_i, to calculate our "final" velocity. Each time the computer goes through the loop, it will use a different value for i, and therefore a different value for v[i]. If we start with v=[0], the first time the computer gets to the loop, it will set i=0. When it gets the the velocity line, it will go look up v[0], which is just the first thing in v, or in our case, 0. If delta_t is 0.1, then v[0]+a*delta_t=0-9.8*.1=-.98. v.append(-0.98) would make our |v=[0,-0.98]|. The second time through the loop, i=1, so v[1]=-0.98, since v[1] will give us the second thing in v. Therefore v[1]+a*delta_t=-.98-9.8*0.1=-1.96, and with the append, our velocity list becomes |v=[0,-0.98,-1.96]. And the computer will continue repeating this process until we reach i=N-1|.

We can also use our velocity to calculate our position. We know from kinematics that $x_f = x_i + v_i \Delta t + \frac{1}{2} a \Delta t^2$. In order for Euler's method to work, $\frac{1}{2} a \Delta t^2$ must be very small, so we ignore it.

Once the loop is finished, the program just builds a plot of the data:

```
#Plot the results
plt.plot(t,x,linewidth=2,color='red')
plt.xlabel('time (s)')
plt.ylabel('Position (m)')
plt.title('Position vs. time for a ball in free fall')
```

Chapter 8

Numerical modeling of a Mass-Spring System

In our last lab we started using Euler's method to solve physics problems. We started with a problem we could do algebraically (a ball toss with no air resistance), but then considered a problem that we could not do with just algebra (a ball toss with air resistance). We did this by using the coupled equations

$$\begin{aligned} x(t+\Delta t) &= x(t)+v(t)\Delta t \\ v(t+\Delta t) &= v(t)+a(t)\Delta t \end{aligned}$$

where for the first case

$$a(t) = -g$$

and for the second

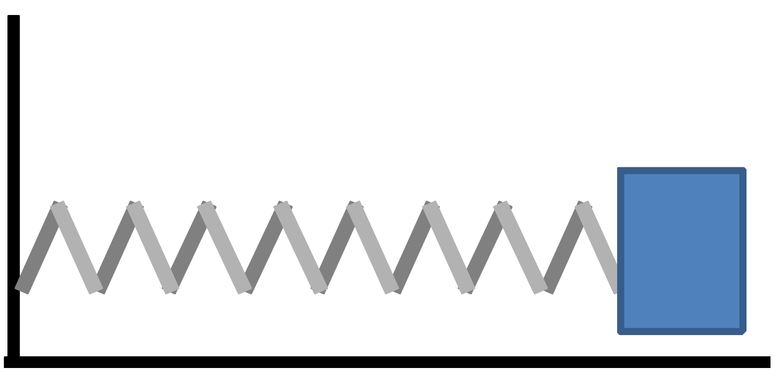
$$a(t) = -g + \frac{D\rho A v(t)^2}{2m}$$

We said that to take on different problems using Euler's method, we only have to change the acceleration term. And we can see that this was true for our moving ball cases. We are going to take on a very different physical system today, and we will see that even though it is very different, we can still use Euler's method to describe the motion, and all we have to do is to change the acceleration term again.

8.0.1 Mass Spring Systems

Let's review Hook's law for springs. Consider a system consisting of a mass attached to a spring lying on a frictionless surface, as depicted in next figure.[29]

[29] A mass-spring system is often called a harmonic oscillator, it will show up under this name in your PH121 book.



We assume that the spring strictly obeys Hook's law, $F = -kx$, where k is the spring constant and x is the displacement from the equilibrium position and F is the spring force. There are only four inputs to this system: the spring constant, the mass, and the initial velocity and initial displacement of the mass. We turn to Newton's second law again to find the acceleration. There is only one horizontal force.

$$\Sigma F_x = ma = -kx$$

so

$$a = -\frac{k}{m}x$$

so we can write a set of coupled equations for the mass-spring system

$$\begin{aligned} x(t+\Delta t) &= x(t) + v(t)\Delta t \\ v(t+\Delta t) &= v(t) + -\frac{k}{m}x(t)\Delta t \end{aligned}$$

This is our Euler's method solution. We can use the same code as we started with last lab for our one dimensional ball motion, but change the acceleration to $a = -\frac{k}{m}x$.

8.1 The exact mass-spring solution

Of course, if you have taken PH123 you know we can also find an exact solution for the motion of a mass-spring system's motion. Let's introduce or review this here. It has quite a bit of calculus, so if you are concurrently taking M12, don't worry. all you need is the answer at the end.

Given the simplicity of this system, it is fairly easy to write down an equation that describes the motion. If your calculus skills are up to it, we can rewrite this using differentials. Recognizing that acceleration is the second time derivative of position,

$$a = \frac{dv}{dt} = \frac{d}{dt}\left(\frac{dx}{dt}\right) = \frac{d^2x}{dt^2}$$

we can rewrite our spring acceleration as

$$\frac{d^2x}{dt^2} = -\frac{k}{m}x$$

We call this type of equation a *differential equation*. In our case, we have an equation that involves a quantity, x, along with it's second derivative. Because we have the second derivative of x, it is called a second order ordinary differential equation. If you stare at this equation long enough, the solution becomes somewhat apparent. Ask yourself: for what function can I take a derivative of twice and get the same function back along with a negative sign and a constant? If you are far enough in your M112 class the answer is obvious, a sine or a cosine.[30] Of course it could also be a combination of sines and cosines, so we can write a general solution

$$x(t) = a\sin(\omega t) + b\cos(\omega t)$$

[30] But if you are not far along in M112, don't worry. I am just trying to motivate the calculus savvy reader to believe the position as a function of time will be a sine or cosine function.

where A and B are constants and either could be zero. If we substitute this in for x in our differential equation, and take a couple of quick derivatives we will see that our solution works so long as

$$\omega = \sqrt{\frac{k}{m}}$$

Note that there are two undetermined constants in this expression, namely a and b. Using a little trig[31], we can rewrite this in an equivalent way[32]

$$x(t) = A\sin\left(\sqrt{\frac{k}{m}}t + \phi\right)$$

with the undetermined constants A, which represents the amplitude or maximum displacement of the oscillation, and ϕ which is referred to as a phase angle. This is true because the sum of a sine and a cosine function is another sinusoidal function (think of all those trig identities you memorized in your trig class!).

If we know what the initial displacement and velocity are, it is easy enough to determine the values of A and ϕ. Doing so at this point would distract from our discussion, so we will forgo the exercise for now. In any event, the solution is oscillatory, as you would expect from a mass on a spring.

[31] OK, a lot of trig.

[32] This is where you should start paying attention if you are concurrently taking M112.

8.2 Adding friction (damping)

A frictionless mass-spring system is obviously not too realistic. Many oscillating systems will not fit our exact equation too well. Real masses have frictional forces acting on them. A mass spring that experiences friction is said to be *damped*.

To modify our Euler code to use damping, we go back to Newton's second law one more time. We need a new expression for the acceleration. There are two forces that act on the spring: the spring force $F = -kx$ which acts opposite the displacement and a damping force $F = -bv$ that acts opposite the velocity. Here b is the damping coefficient and is constant for a given situation.

The dynamic equation that describes the system is then

$$\Sigma F_x = ma = -kx - bv$$

We get a new acceleration

$$a = \frac{-kx - bv}{m}$$

and use this in our coupled Euler equations.

8.3 Review of the Euler Method

Let's review how to turn this into an Euler solution.

Starting from the basic kinematic equation

$$\Delta x = v_o t + \frac{1}{2} a t^2$$

and assuming that for our time interval the change in velocity (acceleration) is essentially zero, then

$$\Delta x = v_o \Delta t$$

or

$$x_f = x_i + v_o \Delta t$$

remember that we have only let our experiment run for a very small Δt, so even though it is accelerating, and the acceleration varies with x, it would not have changed velocity much. And our equation will nearly work. In the limit as $\Delta t \rightarrow 0$ our equation will be exact, just as we said before.

Of course we want go an additional small time interval. That is, let the experiment run from $t = \Delta t$ to $t = t + \Delta t$. Then we can predict that the next position will be

$$x(t + \Delta t) = x(t) + v(t)\Delta t$$

As before we need to estimate a new velocity at the end of our time step, Δt. We again use

$$v = v_o + at$$

assuming a very small time interval (small enough that the acceleration does not change a whole lot over the interval) I can write

$$v(t + \Delta t) = v(t) + a(t)\Delta t$$

where I don't necessarily know that the acceleration is constant in time. But just like in the last equation for position vs. time we did not assume the velocity was constant (we just assumed it was essentially constant over a small Δt) we will assume the acceleration is nearly constant over the interval Δt.

We found (above) that for a frictionless mass-spring system Newton's second law gave us

$$a = -\frac{k}{m} x$$

This acceleration depends on position. It is not constant so technically we can't use the kinematic equations. But it is constant enough over a very small Δt.

Knowing the acceleration, we can use it in our velocity equation.

$$v(t + \Delta t) = v(t) - \frac{k}{m} x(t)\Delta t$$

Now, we have two equations that project the position and speed forward in time, based on what the position and speed currently are:

$$\begin{aligned} x(t + \Delta t) &= x(t) + v(t)\Delta t \\ v(t + \Delta t) &= v(t) - \frac{k}{m} x(t)\Delta t \end{aligned}$$

Just like before, these expressions are coupled: the new position depends on the current velocity, and the new velocity depends on the current position. To find the solution we will take a small step in time, Δt, and calculate a new position. We will also calculate a new speed. Then we will use the new speed to find the next new position Δt later, and the new position to find another new speed. Once we step these two quantities forward in time, we can repeat the process indefinitely.

In short hand these equations become

$$\begin{aligned} x_{n+1} &= x_n + v_n \Delta t \\ v_{n+1} &= v_n - \frac{k}{m} x_n \Delta t \end{aligned}$$

We have labeled each new x and v by how many time steps, Δt, we have taken. The first Δt will be $n = 1$, the second $n = 2$, and so on. Just to be clear, let's again compare

$$\begin{aligned} x(t + \Delta t) &= x(t) + v(t)\Delta t \\ x_{n+1} &= x_n + v_n \Delta t \end{aligned}$$

This means that we start with the initial velocity, v_0 at the initial position x_0. We use these to calculate the position x_1 and the speed v_1 a time $t = \Delta t$ later. Then we start with the initial velocity, v_1 at the initial position x_1 and use the equations to calculate the position x_2 and the speed v_2 an additional time Δt later ($t = 2\Delta t$). Then we use x_2 and v_2 to find x_3 and v_3 and so forth.

8.4 General form for the mass-spring system

We can write our coupled equations in our general form for Euler's method if we want to. Here is how they would look.

Starting with the second order differential equation

$$\frac{d^2x}{dt^2} = -\frac{k}{m} x$$

we write this as two coupled first order differential equations:

$$\begin{aligned} \frac{dx}{dt} &= v \\ \frac{dv}{dt} &= -\frac{k}{m} x = a \end{aligned}$$

To relate this to our general form,

$$\begin{aligned} \frac{dx}{dt} &= v(t) = f(x, v, t) \\ \frac{dv}{dt} &= a(t) = -\frac{k}{m} x(t) = g(x, v, t) \end{aligned}$$

We can write these for time t_n as

$$\begin{aligned} f(x_n, v_n, t_n) &= v(t_n) \\ g(x_n, v_n, t_n) &= -\frac{k}{m}x(t_n) \end{aligned}$$

or more compactly,

$$\begin{aligned} f(x_n, v_n, t_n) &= v_n \\ g(x_n, v_n, t_n) &= -\frac{k}{m}x_n \end{aligned}$$

which gives us what we need to understand our general form

$$\begin{aligned} x_{n+1} &= x_n + (\Delta t) f(x_n, v_n, t_n) \\ v_{n+1} &= v_n + (\Delta t) g(x_n, v_n, t_n) \end{aligned}$$

Of course this is really just

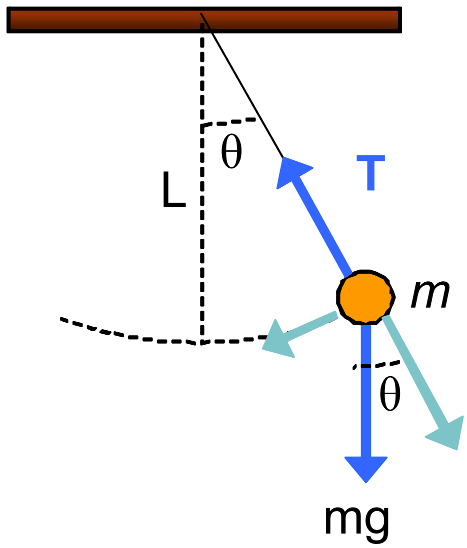
$$\begin{aligned} x_{n+1} &= x_n + (\Delta t) v_n \\ v_{n+1} &= v_n + (\Delta t)\left(-\frac{k}{m}x_n\right) \end{aligned}$$

for our specific case. But sometimes it is easier to write a general routine and then provide $f(x_n, y_n, t_n)$ and $g(x_n, y_n, t_n)$ for our specific case in a subroutine[33]. This way the main routine can be easily used over and over and only the subroutines need to be rewritten if we change specific cases.

[33] The terms "routine" and "subroutine" will be familiar to the computer science majors. For the rest of us they mean "program" and "sub-program" or in MatLab "script" and "function."

8.5 Harder oscillation systems

There are problems worse than damping. After all, if you have taken PH 123 you know that we can find a solution even to the damped case that is a single equation (but it is hard to do). If I can solve a differential equation, why would I need to work out a numerical solution? The answer is that not all differential equations have nice, neat analytical solutions like the one that describes the mass-spring system. For example, consider a pendulum bob on a massless string, as shown in next figure. We confine the motion of the bob to the plane of the page, so that the displacement of the bob can be conveniently described by the angle θ that the string makes with the vertical.



There are two forces acting on the pendulum bob: gravity, which pulls downward, and the tension in the string. The acceleration of the bob can be expressed in terms of a centripetal acceleration (parallel to the string) and a tangential acceleration (perpendicular to the string). There is never a change in velocity in the direction of the string, so the component of gravity parallel with the string must equal the tension. The tangential acceleration of the bob, which is what changes the bob's speed, is thus provided by the component of gravity perpendicular to the string. If you do the geometry, you find that this force has a magnitude of

$$F = -mg\sin\theta$$

Putting this into Newton's second law, we end up with the relation

$$ma_{tan} = -mg\sin\theta$$

The tangential acceleration is related to the angular acceleration by[34]

$$a_{tan} = L\alpha$$

where L is the length of the string, and the angular acceleration (α) is the second time derivative of the angular position, i.e.

$$\alpha = \frac{d^2\theta}{dt^2}$$

Inserting these latter two expressions into our dynamic equation, we get

$$mL\frac{d^2\theta}{dt^2} = -mg\sin\theta$$

At this point, the masses cancel and we can isolate the derivative on the left hand side of the equation:

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\sin\theta$$

[34] Understanding the details here is not so important as noticing that we are again using Newton's second law to find the acceleration.

[35] The analytic solution involves a few not-so-straightforward operations involving differential calculus, as well as integrating the differential equation twice. The second integration is an elliptical integral of the first kind which generally can't be done without resorting to a numerical method.

Despite it's relatively simple appearance, this second order nonlinear ordinary differential equation does *not* have a simple analytic solution[35]. At this point, most introductory physics texts use what is known as a small angle approximation, i.e. for small angles $\sin\theta \approx \theta$, where θ is in radians. Once that approximation is made, the differential equation has exactly the same form as the equation for the mass-spring system, and can be solved analytically. However, we recognize that the solution will only be valid for small angles, generally less than about 15° (it is only really very good for very small angles < ~4°).

But what if you want to know something about the motion of the pendulum for larger angles? You have two choices. Either you can go through the painful process of solving the differential equation and numerically solving the resulting integral (you physics majors will be able to do this by the time you are seniors), or you can numerically approximate the solution to the differential equation in a way that gives us useful information about how the system behaves, even when the angles are large. In other words, you can numerically model the pendulum. It is generally true that, with the introduction of some intermediate variable (in our case v), that any second order differential equation can be broken down into two coupled first order ordinary differential equations. For the pendulum, these first order equations are

$$\begin{aligned} \frac{d\theta}{dt} &= \omega \\ \frac{d\omega}{dt} &= -\frac{g}{L}\sin\theta \end{aligned}$$

This system (or any system like it) can be progressed forward in time using Euler's method just as simply as the mass-spring system. Thus we can solve any second order differential equation using Euler. For the pendulum, we can find the motion without the pesky requirement that the angle be small. Where no simple or easy analytic solution exists, we can still solve the system numerically. This is a fantastic benefit that you physicists and engineers will appreciate more as you take higher level classes!

It is worth noting that there are many more physical systems for which analytic solutions simply cannot be found (not all differential equations have analytic solutions). In those cases, you must solve numerically, there is no other choice![36]

[36] You computer science majors have great job security!

The easiest way to numerically model the dynamics of a system is to ask three questions:

1. What is the current position or state of my system?
2. How fast is it moving, and in what direction?
3. Can I guess where the system will be a short time from now?

This is just what we did in using Euler's method.

Chapter 9

Measurement and Uncertainty III

We should tie our numerical modeling knowledge into our understanding of uncertainty. We want to be able to numerically predict the outcome of an experiment. But that prediction should come with an uncertainty. How can we find an uncertainty for something we found numerically?

Think back to our lab where we measured g. One way to find the uncertainty was to use our equation to find the combination of values that gave the largest estimate for g. Then find the combination of values that gave the smallest estimate for g. For example, when you found g using the pendulum your equation should have been something like

$$g = 4\pi^2 \frac{L}{T^2}$$

we can find the uncertainty using the high/low method by taking:

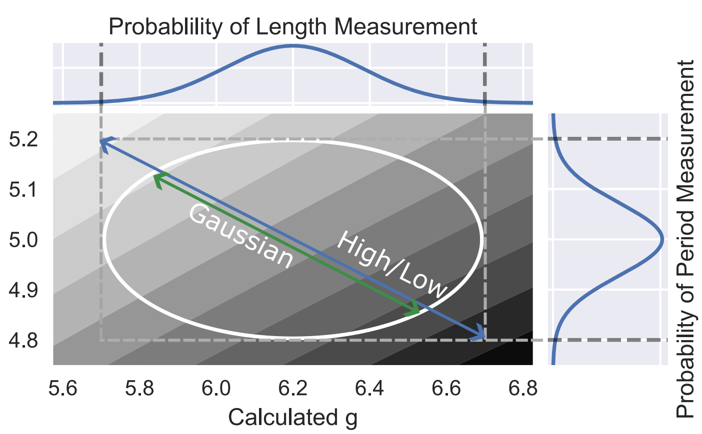
$$g_{\text{max}} = 4\pi^2 \frac{L_{\text{max}}}{T_{\text{min}}^2}$$

where $L_{\text{max}} = L + \delta L$ and $T_{\text{min}} = T - \delta T$. We can also find

$$g_{\text{min}} = 4\pi^2 \frac{L_{\text{min}}}{T_{\text{max}}^2}$$

And then an estimate for the uncertainty would be half the difference between g_{max} and g_{min}.

But, we learned a better method: Gaussian (or standard) error propagation (the one with the derivatives). This graph shows the two methods side by side:



We expect each measurement we take to be close the the mean, and we expect measurements away from the mean to be less and less probable as we move away.

If we say we have a 95% confidence interval (a standard in many fields), what we really mean is that we expect our measurement to fall in that range 19/20 times. The graph above shows our expected measurements for period and length, as well as what value we would calculate for g based on those measurements. The white ellipse marks where we expect 95% of our combined measurements to fall.

We shouldn't expect to get the highest probable period measurement at the same time we get the lowest probable length (which is what high/low method assumes). The derivatives in Gaussian error propagation help take into account that the extremes aren't as likely as something in the middle, and will closely match that ellipse where we expect 95% of our values to fall.

With simple equations, the Gaussian method is best and is a standard across all sciences. Now imagine applying it to Euler's method. Every single timestep you'd have to calculate an uncertainty in your acceleration and then figure out how that impacts your velocity and position. Then you'd have to figure out how the uncertainty in your positions and velocities creates uncertainties in the next position and velocity, then repeat 1,000, 10,000 or even 100,000+ times.

Instead, there's a better method.

9.1 Monte Carlo Error propagation

Instead of calculus, we roll dice (a.k.a. use random numbers). Named after the famous gambling destination, Monte Carlo[37] error propagation uses random numbers to figure out what the uncertainty would be.

[37] If you ever see "Monte Carlo" in the name of a physics method, that just means that we use random numbers to do it. This includes methods we used before computers existed. For example, physicists working on the Manhattan Project literally rolled dice.

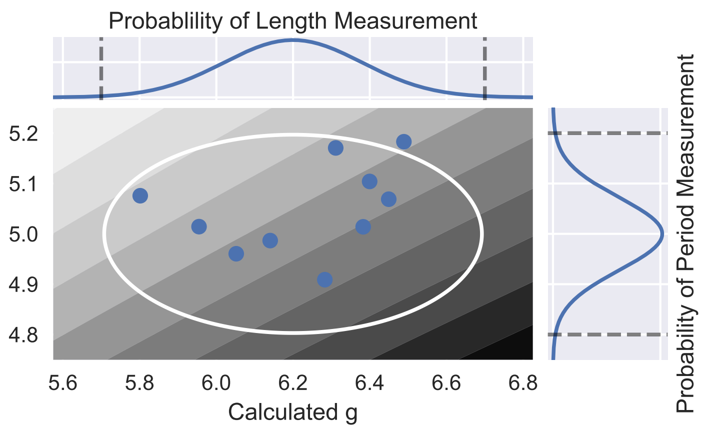
Numpy has a function called "normal" that will randomly generate numbers based on a Guassian distribution. Here's an example on how to use it. Assume that we measured something to be 5 ± 0.2 m long. I could recreate a random measurement with this program:

```
import numpy as np

mean = 5 #Expected measurement (the reported part)
se = 0.1 # Standard deviation of the mean (or standard error).
             # For error, we usually report 2 times this number
random_number = np.random.normal(mean,se) # make a simulated random
    measurement
```

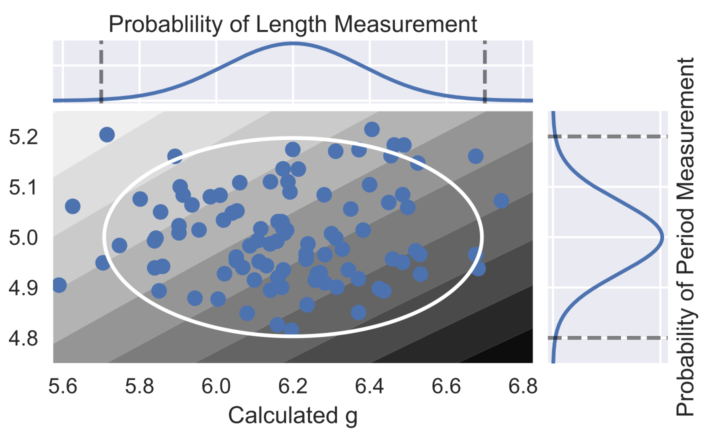
If you need to create more than one random number at a time, you can add an additional argument:

```

random_number = np.random.normal(mean,se,10) # make 10 simulated
    random measurements
```

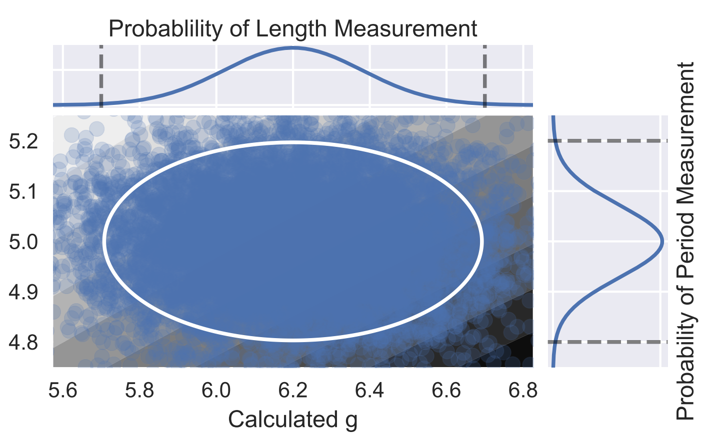
Going back to our pendulum example, this is what you get if you use a program to generate 10 random lengths and periods using our uncertainty parameters:



That's still pretty scattered, but most of the dots fall in our expected circle. We could try going up to 100:



It's better, but still a little sparse. Watch what happens when I go all the way to 10,000:



Notice that the circle is completely filled in, though we still have some stragglers out side of it. To get a final estimate for g, I could just calculate it using all of those randomly generated periods and lengths, then find the mean and standard deviation of the calculated gs.

This method is overkill for something like our simple period equation, but its one of the only ways to do error analysis with Euler's method.

Imagine we measure these values as the inputs we need for our Euler program:

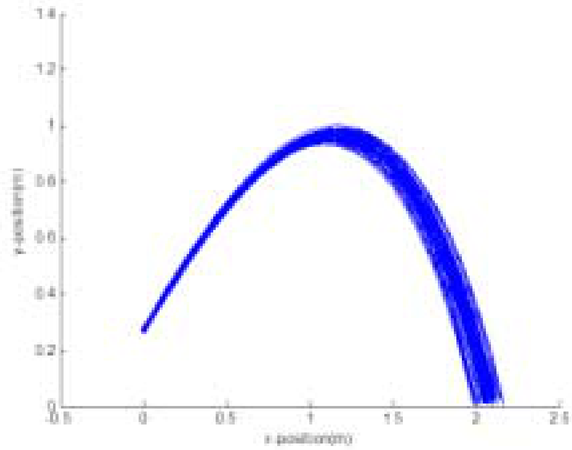
- The initial y position of the projectile $y = (27.00 \pm 1)\,\mathrm{cm}$
- The initial speed of the projectile. $(6.8 \pm 0.2)\,\mathrm{m/s}$
- The launch angle for the projectile (measured from the positive x axis) $(45 \pm 1)^\circ$
- The mass of the projectile.$(0.0003 \pm 0.00001)\,\mathrm{kg}$
- The radius of the projectile.$(0.0125 \pm 0.0001)\,\mathrm{m}$
- The drag coefficient for the projectile 0.46 ± 0.2
- The density of the air the projectile travels through $(1.23 \pm 0.1)\,\mathrm{kg/m^3}$
- the acceleration due to gravity is $(9.8004 \pm 0.001)\,\mathrm{m/s}$

We could set up our Euler code like this to randomize our inputs[38] :

[38] Notice that all of the errors are halved. I'm assuming that the quoted uncertainty is two standard deviations of the mean. If you've directly calculated a single standard deviation of the mean, don't cut it in half.

```
import numpy as np

#Shorten name for randomizer function
normal = np.random.normal

#Set up initial conditions
y0 = normal(0.27,.005) #Starting position
v0 = normal(6.8,1) #Starting Velocity
theta = normal(45,0.5)*np.pi/180 #launch angle, converted to radians
mass = normal(.0003,.00005) #mass
r = normal(.0125,.00005) #radius

d_coeff = normal(.46,.1) #drag coefficient
p = normal(1.23,.05) #air density
g = normal(9.8002,.0005) #acceleration due to gravity
```

If I repeat my program several times and plot the results, I'll get something like this:



To find out where I expect to land, I would just have to save all of the landing positions, then take the average. You could get the uncertainty through calculating the standard deviation and other statistics, or just directly get a 95% confidence interval using Numpy's quantile function.

Part II

Lab Assignments

Lab 1

Measurement and Uncertainty I

1.1 Lab Notebooks

Hopefully you noticed that a lab notebook is required for this class. The lab notebook is designed to be a record of what you did. If you had to repeat today's experiment five years from now, could you do it based on what you write today?

At most professional labs and major engineering companies your lab notebook is considered the property of the company or organization and will stand as a legal document. It is the proof that you did the experiment that you say you did, and that you got the results you say you got. It has to be readable and understandable to someone who did not participate in the lab with you. This is a pretty tall order.

You should write in your lab notebook as you go, not leave it until the end of the day[39]. It will be much easier, and will take you less time as you go. To help you plan your entries, here are the criteria I will use to grade your lab book:

[39] A lab notebook is not a lab report. You just need to take well organized notes on what you did and what you found.

- Describing the goal for the work
  - Usually this takes the form of a physical law we will test.
- Give predictive equations and uncertainties for the predictions based on the physical law.
  - This usually involves forming a mathematical model. You should record any assumptions that went into the model (e.g. no air resistance, point sources, massless ropes, etc.).
    - In lab today we will find the volume of the room. Your mathematical model will likely be $V = \ell \times w \times h$. The mathematical model is not necessarily something complicated, but the reader needs to know how you are doing your calculations.
- Give your procedure
  - Recording what you really did (not the lab instructions), tell what changes you make in your procedure as you make them.
  - Record as you do the work.
  - Record the equipment used and settings, values, etc. for that equipment (see next item).

- Did you learn how to use any new equipment? What did you learn that you want to recall later (say, when taking the final, or when you are a professional and need to use a similar piece of equipment five years from now).

- Record the data you used. The data are all the measurements you took plus your best estimate of the uncertainties in the measurements. Record any values you got from tables or published sources (or from your professor) and state where you got these values. You don't always want to write down all the data you use. If you have a large set of values, you can place them in a file, and then record the file name and location in your lab notebook. Make sure this is a file location that does not change (emailing the data to yourself is not a good plan).
- Give a record of the analysis you performed. You should have given some idea of how you got your predictive equation. Now, what did you do to get the data through the equation? Were there any extra calculations? Did you obtain a set of "truth data" (data from tables or published sources, or from an alternate experiment) for your experiment? If so, did you do any calculations, have any uncertainty, etc. associated with the truth values?
- Give a brief statement of your results and their associated uncertainties.
- Draw conclusions
  - Do your results support the theory? Why or why not? What else did you learn along the way that you want to record.
  - This is where we may compare the percent error to our relative uncertainty.

1.2 Assignment: Practice with Measurement and Uncertainty calculations.

1.2.1 Part 1 Percent Error: Mass of a Cylinder–the hard way

- Given the density of a metal cylinder, use this density to determine the mass the cylinder.
  - You cannot directly measure the mass of the cylinder, You will be provided a mass of the cylinder by your instructor to compare with your calculated value.
  - Report your method for obtaining the mass of the cylinder in your lab notebook (not just your result, but tell yourself in your notebook *how you got your result*).

[40] Notice that the uncertainty in the calculated volume isn't in this list. You don't have to find it yet.

- Report the following results:[40] 1) Density of the cylinder, 2) Predicted Mass of the cylinder, 3) Actual Mass of the cylinder. Comment on the accuracy and precision of your measurement.
- Resources: You may use any equipment or other resources found in the lab or on the internet

1.2.2 Part 2 Combining Uncertainty: Volume of the room

- Determine the volume of this room, *including uncertainties*. Describe your method fully in your lab notebook, including which measuring instruments you used and why, and the uncertainty[41] in each of your measurements.

- The *absolute* uncertainty in the volume is

$$\delta V = \left(\frac{\delta L}{L} + \frac{\delta H}{H} + \frac{\delta W}{W}\right) V$$

from our algebraic method multiplication rule or

$$\left(\frac{\delta V}{V}\right)^2 = \left(\frac{\delta L}{L}\right)^2 + \left(\frac{\delta H}{H}\right)^2 + \left(\frac{\delta W}{W}\right)^2$$

- Compare your answers with those from your neighboring research institutions at the other tables. Are your answers the same to within the values of your uncertainty? If not, explain why they aren't.

[41] Even though the uncertainty equations are given below, you should try to show where they came from in your lab notebook. If you need help doing this, be sure to ask for it.

1.2.3 Part 3: Tie to Experimentation

- We will learn in this class that you should understand the uncertainties in our measuring devices *before* you start performing an experiment. From what you have experienced so far today, why do you think this is so?

1.2.4 Part 4 Combining Uncertainty: Determine the Volume of a Stack of Paper

- Determine the volume of 20 pieces of paper (you can use more, but if you do, replace the number 20 with your actual number in the equation below).

- Determine the uncertainty in your measurement.

- Use your measurement to find the volume of one sheet of paper by dividing. Also determine the uncertainty in your calculation. This should be something like

$$\delta V_1 = \frac{\delta V_{20}}{20}$$

explain what this means in your lab notebook.

- Now measure the volume of one piece of paper directly using instruments (I might recommend a micrometer–ask if you have not used one before).
- How do your measurements compare?
- Which one is more accurate? Which is more precise? Why?

Lab 2

Communicating Results I: Statistical Representation of Data

Complete this lab in an organized fashion in your lab notebook.

Everyone should write their own programs, but you should work together on them as a group. Once you complete a step, stop and help your lab mates until they are caught up with you.

2.0.1 Statistical Data I: How long does it take to walk?

We will repeat one measurement, how long it takes to walk to a destination, many times. Each person in our class will take the measurement once.

1. We'll start by determining a walking destination as a class. Our destination is: _________.

2. Each person in the class should get a digital timer and time their walk to the destination and back. Walk at your normal walking speed. We will stagger when you leave, to avoid walking in groups. While you are not walking, you can begin working on part II.

3. When you return, record your walking time on the board to the nearest second.

4. Record the times for all class members in a table

5. *by hand* determine the mean walking time, the median walking time, and (if appropriate) the modal walking time for the first 5 walkers. Determine the standard deviation of the walking time of the first fve walkers*by hand* (show all your work).

6. Using a computer: (see instructions below)

   - Calculate the mean walking time, median walking time, and (if appropriate) the modal walking time of the class.
   - Calculate the standard deviation of the walking times using both
     - the built in `numpy` function
     - the standard deviation formula coded into python:

$$\sigma_x = \sqrt{\sum_{i=1}^{N} \frac{(x_i - \bar{x})^2}{N}}$$

   - Make a histogram of the walking times.

2.0.2 Numerical Analysis in Python

Running your first Program

Part of this class is learning to solve physics problems with a computer. Today, you will be using the language Python to find the median, and standard deviation of the walking data that we took. Additionally, you will use it to make a histogram of the data.

When programming, it can be really helpful to make notes in your lab notebook about what each program does, things you learned about different functions, etc. At the bare minimum you should include you final program, any graphs it makes, as well as where you saved it, and what name you saved it under. That will make it easier to find in the future.

To begin writing your program, open up your favorite plain text editor. I like notepad++ for windows or text wrangler on mac. Or, if you downloaded Python from Enthought or Anaconda you can use Canopy (Enthought) or Spyder (Anaconda). Enter the walking data like so: (Be sure to use the class data, not this sample data.)

```
#Our walking data
data = [34, 38, 33, 38, 38, 36, 35, 47,36, 32, 40, 40,
       45, 36, 43, 38, 48, 40, 40, 38, 43, 40, 39, 36, 46,
       34, 37, 33, 32, 34 ]
print(data)
```

The first line is called a comment. The `#` tells the Python interpreter (the thing that runs Python) to ignore that line. You should use comments to describe what your are doing in your program, that way you remember what it was later, or if anyone else has to read it they'll know what you did. There will not be any more example comments in this tutorial. *You will have to come up with your own.*

Let's look at the next set of lines:

```
data = [34, 38, 33, 38, 38, 36, 35, 47,36, 32, 40, 40,
       45, 36, 43, 38, 48, 40, 40, 38, 43, 40, 39, 36, 46,
       34, 37, 33, 32, 34 ]
```

They load our walking data into what is called a list. It's a way to save our data under a different name for easier access. The `print(data)` command tells the computer to print out what we've saved in data.

Save your file with the extension `.py`. (Example: `myFile.py`) That tells your computer that it is a Python script.

If you are using Canopy or Spyder, you can run your script by clicking play or hitting f5. If you aren't using one of those, open up your command line on Windows or the terminal on Linux or Mac. Navigate the where you saved your file (ask the instructor for help if you need it) and type in the command:

```
Python 'myFile.py'
```

but insert your file name. That tells your computer to run the Python script. You should see your data printed out on the screen.

Finding the Mean, Median, and Standard deviation

First, remove[42] the `print(data)` line from your program. We don't need to have the computer spit that out again and again.

Add these lines to your script:

```
import numpy as np
dataMean=np.sum(data)/len(data)
print('Mean:  {0:.2f}'.format(dataMean))
```

The first line loads a library called `numpy`. Python keeps a lot of functions in separate libraries. Loading one is sort of like grabbing a book with the right set of instructions in it. Now, all of the `numpy` functions are stored in the letters `np`.

The next line creates a variable called `dataMean`. The numpy `sum` command adds up all of the values in `data`, and `len()` gives you how many items are in the `a` list. (`n` is short for length).

The third line prints `dataMean`. Save your program again and run it to see what happens. Try changing the 2 in the print command line to a 3, save it, run it again, and see what has changed.

Numpy has a function that will calculate the mean for you. Here's our script from above with one addition: `dataMeanNp` saves the mean of the data as calculated by numpy.

```
import numpy as np
dataMean=np.sum(data)/len(data)
print('Mean:  {0:.2f}'.format(dataMean))
dataMeanNp=np.mean(data)
print('Numpy Mean:  {0:.2f}'.format(dataMeanNp))
```

Your assignment for this part is to *add* these parts to your program:

1. a part where the program calculates and prints the standard deviation using the formula in the reading. Hint: if you've been following the lab, `data-dataMeanNp` will subtract the mean from every data point stored in `data`.

2. a part that uses numpy to find the median and standard deviation of our walking data. The numpy function that finds the median is `median` and the numpy function that finds the standard deviation is `std`.

Once you find the median and standard deviation, tell your script to print the median with no decimal places and to print the standard deviation to three decimal places.

You should also include *comments* in your program. Comments are notes for people to read, but that the computer will ignore. You can start a comment

[42] You can remove code by either deleting it, or comment it out by putting a # at the beginning of the line. Commenting out old pieces of code can be really helpful if it's something you'd like to remember, or might use again.

with the # character. Part of keeping a good lab notebook is printing out copies of your programs, complete with comments. Here's the example program, all in one place, with good comments added:

```
#Load the class walking times into the variable "data"
data = [34, 38, 33, 38, 38, 36, 35, 47,36, 32, 40, 40,
       45, 36, 43, 38, 48, 40, 40, 38, 43, 40, 39, 36, 46,
       34, 37, 33, 32, 34 ]

#Importing the numpy library for easy calculations
import numpy as np

#Calculate the mean of data using the mean formula
dataMean=np.sum(data)/len(data)
#Print the mean as a float with two decimal places
print('Mean:  {0:.2f}'.format(dataMean))

#Calculate the mean of the data using numpy's mean function
#If I did dataMean correctly, dataMeanNp should give the same result
dataMeanNp=np.mean(data)
#Print the mean calculated from numpy's function.
print('Numpy Mean:  {0:.2f}'.format(dataMeanNp))
```

Making a histogram

Adding the following code to your script to make the histogram:

```
import matplotlib.pyplot as plt
plt.hist(data, 20, normed=0, facecolor='green', alpha=0.75)

plt.xlabel('My x axis Label')
plt.ylabel('My y axis label')
plt.title('My Title')
plt.savefig('myPlot.pdf')
```

This script will make a histogram with 20 bins and save it to the file `myPlot.pdf`. In your program you should:

- Change the number of bins to something more appropriate for your data. If your fullest bin only has one or two items in it, you have way too man bins. If everything fits into three or four bins, you have too few.
- Fix the plot title and axis labels to match what *you* are plotting.
- Add appropriate comments. If you have enough comments, it can be very helpful to refer back to this program in future weeks. If you don't have enough, you will forget what this program does, and it won't be helpful to you.

Print out your completed program and histogram and add them to your lab notebook.

Lab 3

Measurement and Uncertainty II

The goal of today's Lab is to measure the acceleration due to gravity, g, three different ways. For each case, determine an experimental value for g *along with its uncertainty*[43].

Record how you find g and its uncertainty for each method in your lab notebook. Try to obtain the best value you can for each method.

[43] Remember that if you take several measurements, you can report your value and its error as the mean and the standard deviation of the mean ($\sigma/\sqrt{N}$)

3.1 Finding g

3.1.1 Method 1: Timing a ball drop

Using a stop watch and a tennis ball, drop the ball over a known height and determine a value for g.

3.1.2 Method 2: Using a pendulum

You will learn in PH123 that a pendulum oscillates back and forth at a certain rate. If you don't plan to take PH123, you still know that the pendulum of a grandfather clock sets the rate at which the clock will run. The time it takes the pendulum to go back and forth is called the *period of oscillation.* That period is given by the following equation

$$T = 2\pi\sqrt{\frac{L}{g}}$$

where for some reason the letter T stands for period, and L is the length of the pendulum (measured from the pivot point to the center of mass of the weight), and g is the acceleration due to gravity. Build your pendulum, and measure the period of oscillation using a photogate. From this obtain a value for g.

3.1.3 Method 3: Smart Phone Camera

Take high speed video of a falling ball. Important things to do as you take your video:

- Include a meter stick or something of known length in your video, and make sure that it is about the same distance from the camera as the ball. You do not need to have the ball fall in front of the object.

- Try not to move the camera as you take the video

- If you record with high speed on a cell phone, make sure that the frame rate is constant. Many smart phones will let you start in real time, slow it down, then speed it up again. Do not do this.

Use *Logger Pro*[44] software to analyze the video. The steps to do this in Logger Pro are outlined in the Logger Pro help under “video analysis."

[44] Logger Pro should be installed on all lab computers.

Fit a curve to your data that comes from the video. From you PH121 experience you know that the acceleration due to gravity is constant, so we can use the equation[45]

[45] How would fitting to this equation give us a value for g?

$$y = y_o + v_o t + \frac{1}{2} a t^2$$

to indicate the type of curve to use for our fit. If you have trouble finding the curve fit function in Logger Pro, or have trouble using Logger Pro, call your instructor over.

3.2 Plot Your Results

Create a plot that shows your three different calculated values for g, along with errorbars

A spreadsheet program (e.g. MS Excel or LibreCalc) can graph data, and so can LoggerPro. You may know how to make a graph in one of these tools.

In this class, we are using Python, so you should try making your plot in Python. Last week, we used matplotlib to build a histogram. The command for building a plot with errorbars is very similar. Assuming that you've imported matplotlib as `plt`, the command looks like this:

```
plt.errorbar(x,y,xerr=xerr_variable,yerr=yerr_variable,fmt='o')
```

`xerr` and `yerr` are optional commands. If you you want error bars in the x direction, and you've saved the size of your x error in the variable `my_x_err`, sometime after your x and y lists, you'd include the command `xerr=my_x_err`. If you don't have any x-error bars, leave out `xerr`.

Try to make a plot of the three different values you found for g, with errorbars, by using the `errorbar` command and by borrowing and adapting parts of last week's program. The commands for labeling the axes, title, etc. for an `errorbar` plot are the same as the commands for a histogram.

Additionally, you can change the numbers marking the x-axis to string labels if you put these two lines of code[46] somewhere after you make your plot, but before you save it:

[46] This code assumes that you saved your x-values in the variable `x`.

```
labels = ['Stopwatch','Pendulum','Video']
plt.xticks(x, labels)
```

Lab 4

Experimental Design I: Harmonic Oscillators (masses and springs)

This week's assignment follows the experimental design process outlined in the introduction to the lab. We are trying to determine whether or not I have mostly designed this experiment for you. So this week I want you to identify the design parts and put them in our design process order. For our next design lab, you will have to design the experiment yourself. This week's lab is to get familiar with the process. Perform this experiment as a group.

4.1 First Part: Data Collection

1. Our system will be the mass-spring system and it's hanger. Obtain a set of weights, a spring, a weight hanger, a stand, and a stopwatch. Attach the spring to the stand, and the weight hanger to the spring. Determine the inputs to this mass-spring system that may affect the output quantity of interest (the period of oscillation). Determine whether each of these inputs will affect the period of oscillation. If so, explain how you will control for that input. If not, give justification for why you can ignore that input.

2. Build a mathematical model beginning with the suggestion you got from reading Hook's work (lab introduction).

3. Determine how you will measure the period of oscillation. Remember that you want to minimize the amount of uncertainty in your measurement. Techniques we have learned in previous labs may help. Record your method. You should plan any graphs you will make and in general plan how you will report your data and whether or not your experiment is successful.

4. Discuss how you might go about making your equation look linear by a proper substitution of variables. Explain why this might be useful.

5. Select a range of variables. (e.g. $m = 20\,\mathrm{g}$, $30\,\mathrm{g}$, $40\,\mathrm{g}$, $50\,\mathrm{g}$, and $100\,\mathrm{g}$). Don't use $80\,\mathrm{g}$ because I want to reserve this value for a special purpose below. Stop at about $100\,\mathrm{g}$.

6. Plan your procedure and record your plan in your lab notebook.

7. Perform the experiment. Your plan probably includes determining the period of oscillation for masses that you have selected. Be sure to record any measurement uncertainty. Make the graphs of your data that you planned including the appropriate error bars and a best fit line (See second part). Attach the graphs in your lab notebook. Record what you do and highlight any deviations from your planned procedure. Record your data or

your data file name and location. Show your analysis and give your results. Draw conclusions. We will check these conclusions in the third part. But state whether you believe that $T \propto \sqrt{m}$

4.2 Second Part: fitting a line

In order to fit a line to your data, you'll need to teach Python how to do a linear least squares fit using the equations and the example function in the lab introduction. You will be doing many linear least squares fits, so it is in your best interest to create a least squares function in its own file that you can use in the future. That way you don't have to copy and paste it over and over again. These steps will teach you how.

1. Linearize your equation. Remember, linear least squares fitting only works on lines.

2. Type out the example program given in this lab's intro, and save it in the folder where you have been saving your other programs from lab. Make sure that you match the indentation from the example. Python uses indentation to tell where functions end. Here's a quick example:

```
def line(x,m,b):
    return m*x+b
```

   Notice the difference in indentation between the two lines. The "def" command tells Python that you want to create your own function. In this case, the function is called `line` and takes inputs `x`[47],`m`, and `b`. The return command tells Python what you want to get out of your function. You could define a function this way and get the same result:

```
def line(x,m,b):
    y=m*x+b
    return y
```

   But you will get an error if you enter this:

```
def line(x,m,b):
    y=m*x+b
return y
```

   Indentation is very important in Python. When you define a function, everything that is indented below it is counted as part of that function. Not indenting the return statement tells Python that you've ended your function, and it won't know what to return. Here's one more example:

```
def line(x,m,b):
    y=m*x+b
```

[47] This function assumes that `x` is a numpy array. If it is a regular list, the multiplication won't work the way you expect it to.

```
    return y

y=line(5,2,1)
```

The previous piece of code tells Python what we want our function to be. Then, the very last line (notice that it is not indented) tells Python that we want to use our function, and that we want `x=5`, `m=2`, and `b=1`. Python will perform the operation and save the number 11 (the result of 5*2+1) into the variable `y`. Python will also treat any variables inside of functions as separate from the ones outside, meaning the `y` in our function is independent of the `y` in the last line. Once you define a function, you can use it over and over again in your program.

3. Test your function. Put these lines of code *after* your function[48]:

```
if __name__ == "__main__":
    xdata=[1,2,3,4,5,6]
    ydata=[1,2,3,4,5,6]

    #Run linear least squares fit on the data
    slope, intercept = linear_least_squares(xdata,ydata)

    #Print out the test values
    print('Slope: {}'.format(slope))
    print('Intercept: {}'.format(intercept))
```

If your program is working properly, it will print:

```
Slope: 1.0
Intercept: 0.0
```

[48] The `if __name__ == "__main__":` line tells Python to only run this part of the program when you are running this particular file. If you leave it out, Python will run this test whenever you load your function into another file.

4. It is also very important to be able to calculate error. Modify the `linear_least_squares` function so that it also returns the error in the slope (σ_m) and intercept (σ_b). As a reminder, these equations calculate the error:

$$\sigma_m = \frac{\sigma_y}{\sqrt{N\left(\langle x^2\rangle - \langle x\rangle^2\right)}}$$

$$\sigma_b = \sigma_y \sqrt{\frac{\langle x^2\rangle}{N\left(\langle x^2\rangle - \langle x\rangle^2\right)}}$$

where

$$\sigma_y = \sqrt{\frac{1}{N-2}\sum_i^N \left(y_i - b - m x_i\right)^2}$$

The error in both the slope and intercept should be zero if your fit is working properly.

Now start building a script to fit this week's data.

1. Open up a new Python script and save it in the same directory as your least squares file. Put this command at the top of your file: (For this example, the file with the least squares fitting functions in it was saved with the name linear_least_squares.py, change the name to match accordingly.)

```
import linear_least_squares
```

   This line tells Python to load all of the functions in linear_least_squares.py. You can now use your least squares fitting function in this program.

2. Load your data into your program. Doing math with data sets is easier in Python if you use the numpy library. It can be very helpful to save our data like this[49]:

```
import numpy as np
x_data=np.asarray([1,2,3,4,5])
y_data=np.asarray([2.1, 3.9, 5.8, 8.4, 11])
```

   As an experiment, tell Python to print(x_data*y_data) and see what happens. Numpy arrays can only have numbers in them, and so multiplying two numpy arrays will multiply each item in the list individually. Since Python lists can have anything in them (numbers, words, other lists) Python doesn't have a built in way to do math on entire lists without doing quite a bit more work. That's mostly because it doesn't make any sense to say 5*'hello'.

3. Do a least squares fit to your data using your fitting functions.

4. Check your fit equation by using your fitted slope and intercept to calculate a new set of y values (yfit=m*x+b) and then plotting them. (Data points with errorbars, as well as the fit line) The matplotlib.pyplot command for plotting a regular line is plot(x_points,y_points).

[49] It would be good to give your data more descriptive names than x_data and y_data, and add comments.

4.3 Third Part: Interpolation and Extrapolation

We would like to test the equation or "law" you developed in the last part. We will use the equation to predict periods for masses you have not yet used.

1. **By interpolation, predict the period of oscillation for an** 80 g **mass.** Record your methods and results. Interpolation means to predict an output value (in this case, a period) for an input value that falls within the range of the input values you have used in your measurements. If you measured periods for 20 g, 30 g, 40 g, 50 g, and 100 g, then 80 g is within this range. Using the curve fit equation generated by the data we measured, we can plug in 80 g

and predict the period for our spring with an 80 g mass. This is interpolation. This will test our model to see if it works for new inputs. If it does not, our model is probably not good.

2. **By extrapolation, predict the period of oscillation for a** 300 g **mass.** Record your methods and results. Extrapolation means to predict an output value (in this case, a period) for an input value that falls outside the range of the input values you have previously measured. If you measured periods for 20 g, 30 g, 40 g, 50 g, and 100 g, then 300 g is outside this range. Using the curve fit equation generated by the data we measured, we can input 300 g and predict the period for our spring with an 300 g mass. Extrapolation is more risky. The conditions of our experiment might change outside our range (think, in a limiting case, we could break the spring, and get an infinite period!). But if things are done carefully, this is also a test of the validity of our model.

3. Measure the period of oscillation for the 80 g and 300 g masses. Be sure to account for all uncertainties. Compare your measurements with your predictions, and comment on the level of agreement.

4. Now that we have tested our mathematical model for the relationship between period and mass for a mass-spring system, you can report it. Determine values for your constants, including uncertainties . Record your methods and results.

4.3.1 Third Part: Further Discussion

1. An often useful tool, especially when your data is not naturally linear, is to plot it on a logarithmic scale. Create such a graph using Python and attach it to your lab notebook. The matplotlib function is `semilogy(x_data,y_data)`. Comment on what you see.

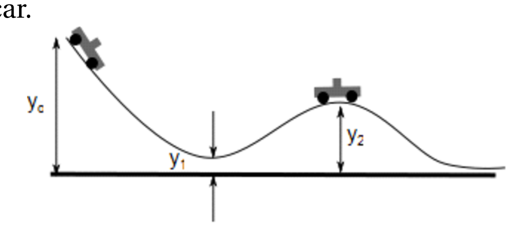
2. Don't forget to make good comments on what you did and how you did it in your lab book.

Lab 5

Experimental Design II: Conservation of Energy

This week we will practice experimental design with a new context. I wont spell out all the steps, so your lab group and you will have to work through the experimental design steps.

Suppose you have been told that energy is conserved (I hope you have by now in PH121). This is our model–the idea that energy is conserved. That is, that it is never lost, just transferred from one form of energy to another. A colleague suggests a method to test this model. He builds a pine-wood derby track and a pinewood derby car.



Your colleague suggests to you that if energy is conserved, you should be able to predict the velocity of the car at points y_1 and y_2.

Another colleague steps in and suggests that you need to be concerned about the energy tied up in the rotational kinetic energy of the wheels. You may not have heard about this in your PH121 class yet. She says that is OK, because you can do a quick fix. She suggests that you should include a factor of about 10% of the translation kinetic energy. That is, compute the kinetic energy in this case as

$$K = (1.1)\frac{1}{2}mv^2$$

that should account for the wheel rotational kinetic energy.

5.1 Assignment

5.1.1 Part 1:

As a group design an experiment to test the model. Your design should include the steps from lab 4 (briefly repeated here)

1. Identify the system to be examined. Identify the inputs and outputs. Describe your system in your lab notebook.
2. Identify the model to be tested. Express the model in terms of an equation representing a prediction of the measurement you will make. Record this in your lab notebook. (If you have not solved this problem in your PH121 class yet, call me over and we will go through it together).
3. Plan how you will know if you are successful in your experiment. Plan graphs or other reporting devices. Record this in your lab notebook. For today's lab, I will provide photogates and the car and track. If you need other equipment, ask.
4. Rectify your equation if needed. Record this in your lab notebook.
5. Choose ranges of the variables. Record this in your lab notebook.
6. Plan the experimental procedure. Record this in your lab notebook.
7. Perform the experiment. Record this in your lab notebook. Go through all the steps of performing an experiment. End with a conclusion that clearly states whether your experiment supported the model, falsified the model, or, if neither was possible, try to explain why.

5.1.2 Part 2

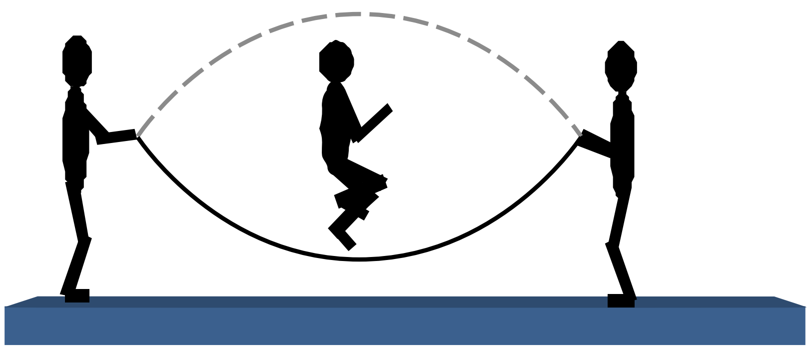
Work on your proposals. They are due in two weeks.

Lab 6

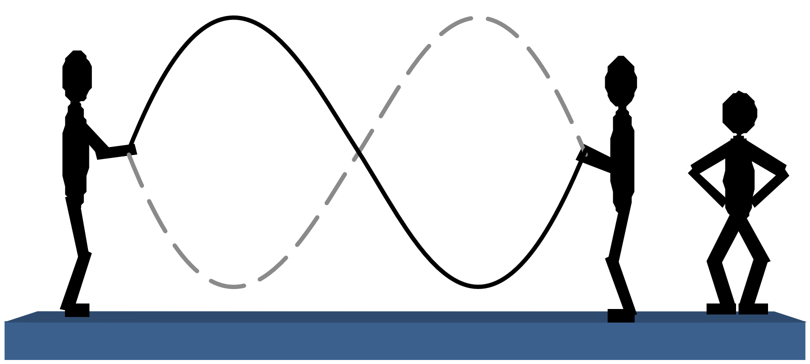
Experimental Design III: Standing Waves on Strings

6.1 Introduction

Standing waves result from making waves that reflect back on themselves, or by making waves on both ends of a string. When we were children we formed a type of standing wave with jump ropes.

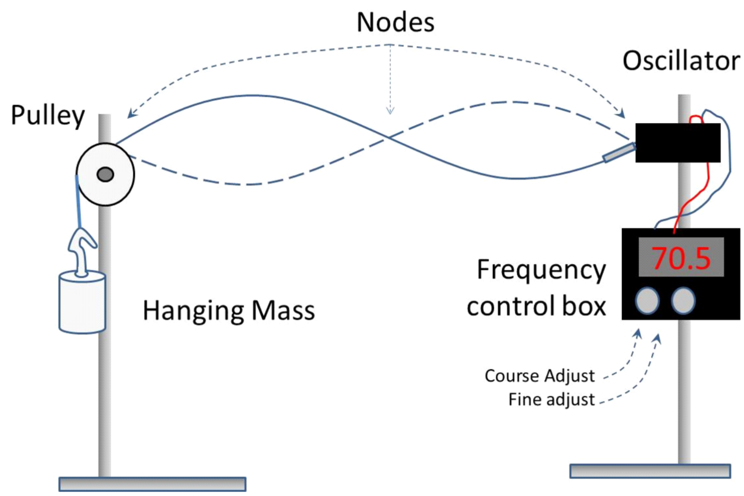


This standing wave had a part that went up and down in the middle and two parts that did not move much on each end (called nodes). But if you had a bored kid, you might have seen a standing wave that looks like this.



This is not so good for jumping, but makes an interesting picture. The part in the middle that does not seem to be moving is called a node. Really there are also nodes on each end of the rope as well. So altogether there are three nodes in this picture. We can make standing waves that have many nodes. If you try this with a jump rope, you will find that the more nodes you have, the faster you have to shake your end of the rope. Another way to say this is that the frequency of your wave you are making must increase with the number of nodes. This is part of today's model.

In the setup on your table, ring stands are holding strings, and there is an oscillator on one end that has a frequency control box. The other end has a pulley and a hanging mass to provide tension on the string.



Experimentally we find that not all frequencies will make standing waves. So our model includes the idea that only some frequencies produce standing waves. Our model also includes the idea that the weight of the string will change which frequency will make a standing wave. If you play a stringed instrument, you may have noticed that some strings are thicker than others. Thicker strings have different standing wave frequencies than thinner strings. If you study this model (some of you will in PH123), you can derive an equation that tells which frequencies will work

$$f = \frac{n}{2L}\sqrt{\frac{Mg}{\mu}} \tag{6.1}$$

where n is an integer ($n = 1, 2, 3\ldots$). This integer for strings is the number of nodes minus one $n = n_{nodes} - 1$. So you can form a standing wave, then count the places that don't seem to move (remember the ends!) and subtract one to find n. The frequency that creates the standing wave should be a function of n. **Our model tells us that if we know n, we should be able to predict the frequency**. You will find that for each way you can make a standing wave, a small range of frequencies will make the standing wave, not just one, single frequency. But the frequency that produces the largest standing wave is the one we want (biggest amplitude–or the one for which the wave looks bigger) . That was the f that was included in forming our model equation. Of course there are other variables in our equation, so we should find out what they are.

The quantity, μ, is the linear mass density. It is defined as mass of the string divided by the length of the string. So μ tells us how massive the string is.

The quantity, L is the length of the string that is participating in the waving, $g = 9.8004\,\text{m/s}^2$ is the acceleration due to gravity, and M is the hanging mass tied to the end of the string beyond the pulley.

One way we could verify our model equation is to use it to predict one of the input values. Let's use μ. **The idea is to use our model equation to somehow find μ and then measure μ to see if the model equation prediction is good.**

It might be tempting to just solve the above equation for μ and report the answer from one measurement. And of course that will work. But I want to see if you can use some of the things we learned.

We learned earlier that we can take more than one measurement, and use those measurements together with a curve fit to solve for a fit parameter. I want you to do this. The quantity μ should be in one of the fit parameters. Then you can solve for μ using the fit parameter given by your Python linear fit code. This is a more robust way to find μ, and it is the way I want you to proceed (Even if you solve for μ several times and take a mean and standard deviation–it will work and it is a good experimental technique–but I want to see if you can find and use your linear least squares code, so using your code gives the most points).

You may have to adjust the amplitude knob on the frequency controller for some frequencies to keep the apparatus from shaking itself apart. The frequency controller has a fine and a course frequency adjust knob, and a digital frequency display.

6.2 Assignment

As a group design an experiment to test the model. Your design should include the steps from lab 5.

1. Identify the system to be examined. Identify the inputs and outputs. Describe your system in your lab notebook.
2. Identify the model to be tested. Express the model in terms of an equation representing a prediction of the measurement you will make. Record this in your lab notebook. (If you have not solved this problem in your PH121 class yet, call me over and we will go through it together).
3. Plan how you will know if you are successful in your experiment. Plan graphs or other reporting devices. Record this in your lab notebook. For today's lab, I will provide the frequency control box, oscillator, rope, and pulleys. If you need other equipment, ask.
4. Rectify your equation if needed. Record this in your lab notebook.
5. Choose ranges of the variables. Record this in your lab notebook.
6. Plan the experimental procedure. Record this in your lab notebook.
7. Perform the experiment. Record this in your lab notebook. Go through all the steps of performing an experiment. End with a conclusion that clearly states whether your experiment supported the model, falsified the model, or, if neither was possible, try to explain why.

Lab 7

Numerical modeling of Projectile motion

7.1 Assignment

Because we are learning a major new skill. we will take three weeks to complete the experiment we are starting today. This will affect your lab notebook. Complete each part of the lab each week in an organized fashion in your lab notebook. Make graceful end-of-class entries so you can start up again the following week. As always, part of the grade will be based on neatness and organization!

1. Create a Python script that will numerically model the motion of a spherical projectile shot straight up and then falling back down using Euler's method. Assume there is no air resistance. As input quantities you should provide the following:

   - The initial y position of the projectile $y = 70.00$ meters
   - The initial speed of the projectile. $30.0\,\mathrm{m/s}$
   - The time step size.0.1 seconds
   - The acceleration due to gravity $9.8\,\mathrm{m/s^2}$

   Make these quantities variables so you can easily change values and recalculate. The program in the precious section will walk you through this part. Save your script and record where you saved it. Include a scatter plot of y vs t in your lab notebook.

2. Copy and then modify your Python script to numerically model the motion of a spherical projectile being launched from a cannon using Euler's method. As input quantities you should add the following (Make these quantities variables so you can easily change values and recalculate):

   - The initial x position of the projectile $x = 0.00$
   - The launch angle for the projectile (measured from the positive x axis) 45 degrees Include a plot of y vs x in your lab notebook. Save your script and record where you saved it.

3. Copy and modify your script to include air resistance. Air resistance will add a new resistive force that is proportional to the square of the projectile's velocity, i.e.

   $$F_R = \frac{1}{2} D \rho A v^2$$

   where D is the drag coefficient, ρ is the density of the air, A is the cross-sectional area of the projectile presented to the air (in our case $A = \pi r^2$), and v is the speed. The force is directed opposite the velocity of the projectile. You will need components of this force. You should convince yourself that

$$F_{R_x} = -\frac{1}{2} D\rho A v(n)\, v_x(n)$$
$$F_{R_y} = -\frac{1}{2} D\rho A v(n)\, v_y(n)$$

where

$$v(n) = \sqrt{(v_x(n))^2 + \left(v_y(n)\right)^2}$$

and $v_x(n)$ and $v_y(n)$ are the components of the velocity. This form of the components of the resistive force avoids having to calculate the angle at each of our Euler steps. As input quantities you should provide the following (Make these quantities variables so you can easily change values and recalculate):

- The mass of the projectile: $0.020\,\mathrm{kg}$
- The radius of the projectile: 0.02 meters
- The drag coefficient for the projectile: 0.2
- The density of the air the projectile travels through: $1.23\,\mathrm{kg/m^3}$

4. Include a scatter plot of your x and y positions for each time step in your lab notebook

Lab 8

Numerical Modeling of a Mass-Spring System

8.1 Assignment

Complete this lab as a group and record your experience in an organized fashion in your lab notebook. Part of the grade will be based on neatness and organization!

1. Copy, then modify your one dimensional Euler code and then modify it to numerically model the motion of a mass-spring system with a mass of 200.0 grams, a spring constant of 0.500 N/m, an initial displacement from equilibrium of 15.0 cm, and an initial velocity of zero. Use a time step of 0.01 seconds, and simulate the motion for a total of 20 seconds. Graph your results, and comment.

2. Repeat problem 1, but this time using a damped oscillator. Damping is a resistive force that is opposite the direction the mass is traveling. The resistive force has the form $F_R(n+1) = bv(n)$ where $v(n)$ is the mass speed and b is a constant that tells how much resistance the system has. In your modeling, use $b = 0.05$ kg/s.

3. Investigate what happens with each of the above modeling scenarios when you increase or decrease the time step.

4. If there is time, try different masses, spring constants, damping coefficients, and so on.

Lab 9

Numerical Modeling and Uncertainty

9.1 Assignment: Modeling and Uncertainty

In this lab we will finally test our numerical modeling of a projectile. We will use our spring cannons and shoot balls that have large drag coefficients. **Be careful not to shoot people, including you–don't look down the barrel of the cannon! Also, don't load the cannon by pushing down on the ball in the cannon, set the spring first and then gently put in the ball If you push on the Styrofoam balls you will have flat balls that don't match our models.**

There is too much to do in this lab to have everyone do all the parts. You will have to split up your group. So how do you record all the experiment if you only do part of it? The way this is done is to write down the results obtained by the other subgroups, and refer to their lab notebooks for the details.

To help determine how you will break things up, there are four major parts to this lab:

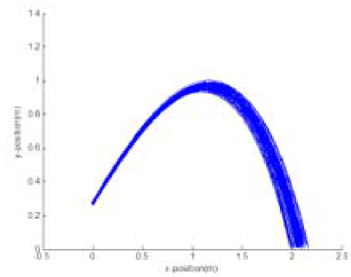
1. Predicting the range of the ball using Euler's method plus Monte Carlo error propagation.
2. Measuring the drag coefficient of the Styrofoam ball
3. Measuring the initial velocity of the Styrofoam ball
4. Actually firing the cannon and confirming your measurements

9.1.1 Predicting the range of the ball

1. **Predict** the maximum range for the path of a Styrofoam ball (or a plastic ball if we don't have any Styrofoam balls) shot out of the spring cannon. Do this with a numerical model of the ball's flight path written by one (or more) of the members of your group that has been changed so that it can find the error estimate as described in today's lab reading.
2. Determine the uncertainty in your prediction with your new code that chooses values for the input parameters randomly[50] from within the parameter uncertainty ranges. This is actually more efficient than trying to pick values in a non-random way to cover all possible combinations of uncertainties! You will need to take about 100 or more sets of parameters to get a good estimate, but if you built your code so you can easily change your parameters, this will go quickly (if you put actual numbers into your equations in the previous labs, it might be faster to fix your Python code first to make this part go more quickly. If you need help, call me over).

[50] Make use of the normal function from numpy as described in the chapter introduction.

3. You want to do this for every variable that has uncertainty, and you want to do the calculation many times so you get a good idea of the precision of your answer. Here is a figure showing an example result:



4. Additionally, you need to be able to find where the ball hits the ground. You can do this by either changing your Euler `for` loop into a `while` loop, or by inserting an `if`[51] statement.

[51] The command `break` will tell your program to exit whatever loop it is currently in.

9.1.2 Finding the drag coefficient

1. You will need to find the drag coefficient. A subteam should find a value (and uncertainty) for the drag coefficient. One way to do this is to use our high speed cameras and film the ball dropping. Since there is significant air resistance, the ball will reach terminal speed. At this point, there is no acceleration, so

$$\begin{aligned} \sum F &= 0 = R - F_g \\ F_g &= R \\ mg &= \frac{1}{2} D \rho A v^2 \end{aligned}$$

so

$$D = \frac{2mg}{\rho A v^2}$$

Since we know or can measure everything except D, we can solve for D. Use $g = (9.8004 \pm 0.0001)\,\mathrm{m/s}$ and $\rho = (1.23 \pm 0.1)\,\mathrm{kg/m^3}$.

- The rest of the values you will need to measure. Use the digital camera and Logger Pro to estimate the terminal velocity. You will have to drop the ball from high up to get a good value (you have to give it time to reach terminal velocity). Near the ground there can be problems due to the air flow hitting the ground, so don't use positions that are close to the ground.
- I suggest you have a team of people from your group do this while a second team modify your Python code. Remember you need an uncertainty in D.

9.1.3 Finding the launch speed of the styrofoam ball

1. You will need to find the launch speed of the ball, with its uncertainty. Air resistance makes a big enough impact that using Logger Pro and Video Analysis isn't feasible. Instead, you can find the velocity using a photogate set up where the ball exits the cannon. A few notes:

    - As a reminder, a photogate shows the time that a laser was blocked, so to turn that into a speed, you'll need to divide a distance by the time
    - You can get uncertainty in time through making several measurements, then finding the average and standard deviation of the mean.
    - The distance to use is a little more complicated than just the uncertainty in the diameter of the ball. If the laser isn't perfectly aligned with the center of the ball as it goes through, then the length of the ball that blocks the laser will be shorter than the diameter (i.e. less ball will travel through the laser beam). With a bit of geometry, you can show that

      $$d_{beam} = d_{ball}\sqrt{1-\alpha^2}$$

      where d_{beam} is the amount of ball that went through the beam, d_{ball} is the actual diameter of the ball, and α is how far from the center (as a ratio of the radius) of the ball the beam is.

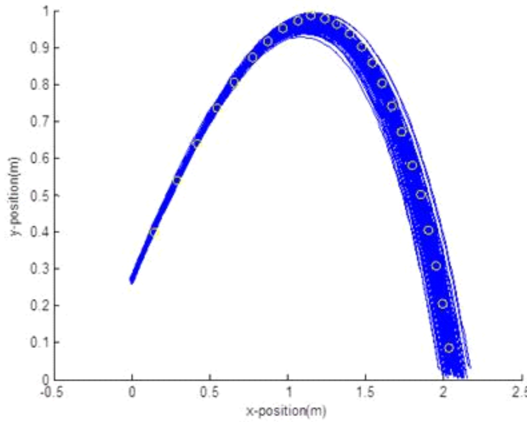
      Here's an example on how to use this equation: imagine I have a ball with a diameter of 2cm, and I think that I can confidently keep the laser within 0.5 cm of the center of the ball (50% of the radius), the smallest diameter that I would expect to pass through the beam would be:

      $$d_{beam} = 2cm * \sqrt{1-0.5^2} = 1.73cm \approx 1.7cm$$

      Since I'd expect my diameter to be somewhere between that number and 2 cm, I'd quote the diameter that passes through the beam to be 1.85 ± 0.15 cm.

9.1.4 Actually Firing the cannon

1. Verify the prediction. Did the ball land within your error range? If you use the digital cameras again for this part, you can determine if the flight path falls within the range of possible flight paths. A figure like the following might help you decide:



2. Does the data support your modeling prediction? If not, what is likely the problem?

    - In your discussion make sure your other team members understand the part of the project that you did.
    - You might consider having a third team collect data for the ball shot with the digital camera while the first two teams are working on the drag coefficient and the numerical prediction.