Physics 1401V: Honors Physics I
TA/Instructor Manual

Fall 2011

An important note:

Until very recently, the PHYS1401 and 1402 labs were run in an *ad hoc* manner, with much left to the whims of TA's, instructors, and students. Over the last several years, the lab has slowly converged towards a more conventional style, in which each of the groups does the same exercise in a given week. Some of the exercises are "borrowed" from 1301, and some of them are unique to this course. The new "system" is much easier for the departmental lab supervisor to administrate, and allows for virtually any TA to walk in and teach a lab.

The labs are, on the whole, much more demanding than 1301, and the student lab manual is written in a very different style. Unlike the 1301 labs, there are numerous points (such as in the air resistance lab) where a reasonable amount of care is required to obtain good data and analyze it. The good thing about honors is that there will always be students who can meet your standard, no matter high how you set it. At the same time, I have learned over the years that students are going to pack up and leave at the end of two hours, if for no other reason than they have another class. **They cannot afford wasted time because a TA was not prepared.**

1. You must complete the lab yourself each week before the students walk in the room. If there is a problem, then work with Sean Albiston to fix it. He will always be able to set up the lab on Thursday afternoon of the previous week. This gives you ample time to do the lab.

2. You must also complete the analysis for the lab. Note that you are not merely expected to know how the apparatus works. You must know how to execute the experiment in the best possible way. This means doing the analysis, and if necessary repeating the experiment until you figure out how to do it correctly.

3. You must test all of the set-ups (at each table) each week. Once again, if there is something missing or some other problem, work with Sean Albiston to fix it. Arrive early for your lab period and make sure all of the hardware is working and is configured correctly. Understand the pitfalls and remind the students if there are common problems. **In past years, I have seen an extraordinary amount of time get wasted because TA's did not inform the students how to set up the Vernier hardware properly. This is your responsibility, as in many cases full instructions are not in the lab manual. The usual pitfall is the data collection rate.**

4. If something bad happens during the lab: a) find Sean to make sure it is fixed, and b) check in with the other TA's. Sean has spare equipment and a lot of experience. Chances are that he can fix your problem in 5 minutes, but only if he knows about it.

Before you get started with labs…

Get yourself a copy of 1301/1302 TA manuals. You can ask for these at the front desk of the Physics building, or ask the lab coordinator, Sean Albiston. If you get them from the front office, have the professor teaching this course send along permission for the hard copies. Electronic copies and other resources are available at the following website:

http://groups.physics.umn.edu/physed/Research/Lab%20Manuals/Lab%20Manuals.html

We share quite a bit of material and equipment with these courses and the manuals have many useful tips on using it, along with some data. The appendices of the student course manuals are also quite helpful for these purposes.

We also use the same video acquisition LabView® program, VideoRecorder, as the 1301 class. For the anlaysis, we use a freeware program called *Tracker*. You can find information and documentation at the following site:

http://www.cabrillo.edu/~dbrown/tracker/

Tracker does have its limitations, so, in conjunction, we use Microsoft® Excel for the actual data analysis. The data can be easily copied from *Tracker* in to a spreadsheet. Become familiar with this program.

Another piece of software that we use in both 1401 and 1402 is Vernier's Logger Pro®. This is used in the regular 1202 and 1302 classes, but we've found use for it in the first semester as well, primarily for photogates. If you are not familiar with this program, there are instruction manuals available online.

In 1401, we use the Vernier photogates and motion detectors. You must make sure these are working properly before each lab period. Most of the problems that occur with these have to do with software settings and not failure of the hardware.

This is the lab schedule based on Crowell's experience in 2010. The rotations lab is different this year. Also a lot of extraneous material has been removed from both the student manual and this manual. Your instructor will probably make his or her own changes. Note that this schedule does not necessarily correspond to the order of the labs in either this manual or the student lab manual. In the past, the first open week (in October) has been needed, because the students have not quite finished the previous labs. The collisions lab is still a tad long for two weeks. The second open week is Thanksgiving week.

Week of	**Topic (Numbers refer to sections of 2011 Student Manual)**	**To be graded or handed in**
13-Sep	Intro/Kinematics (2.1)	
20-Sep	Kinematics (2.2)	
27-Sep	Kinematics (2.3)	
4-Oct	Collisions (Sec. 3.1 and 3.2)	Lab book graded in lab (Kinematics)
11-Oct	Collisions (Sec. 3.3 and 3.4)	
18-Oct	Catch-up and papers	Lab book graded in lab (Collisions)
25-Oct	Rotational Motion (4.1)	First paper due at beginning of lab
1-Nov	Gyroscope (4.2)	
8-Nov	Gravity Simulations (5)	Lab book graded in lab (Rotations)
15-Nov	Gravity Simulations (5)	
22-Nov	Open week (write papers)	
29-Nov	Damped SHM (6..1 and 6.2)	Lab book graded in lab (Gravity)/Second paper due at beginning of lab
6-Dec	Driven SHM and Resonance (6.3)	
13-Dec	Wrap-up	Lab book **handed in** at end of last lecture Revised paper (optional) **handed in at end of last lecture.**

Kinematics

Galileo's Experiment: Measuring Acceleration Due to Gravity

Purpose:

In this lab students will drop similarly sized balls (of various masses) and record the vertical fall of each using a video camera. They will analyze the video and fit the trajectory with a parabola that will yield each ball's acceleration, which is approximately **g**, 9.806 m/s^2.

The focus of this lab should be on how the students relate the motion of the falling ball with the parabolic plot of its position as a function of time. Also, as with all labs in this course, students should also be paying close attention to accuracy, precision and uncertainty of their measurements, and how these relate to their equipment and procedure.

Materials:

Students will need the following materials:

- a meter stick (and a ring stand - optional),
- tape,
- 1 set of 1" diameter balls
  - steel (65.4 g),
  - white plastic (11.5 g),
  - polyethylene plastic (8.8 g),
  - wood (5.0 ± 0.3 g),
  - aluminum (23.3 g),
  - tungsten carbide (126.9 g)
- a ball-dropping mechanism,
- bubble wrap and tray,
- a video camera and tripod, and
- a computer with VideoRecorder and *Tracker* programs.

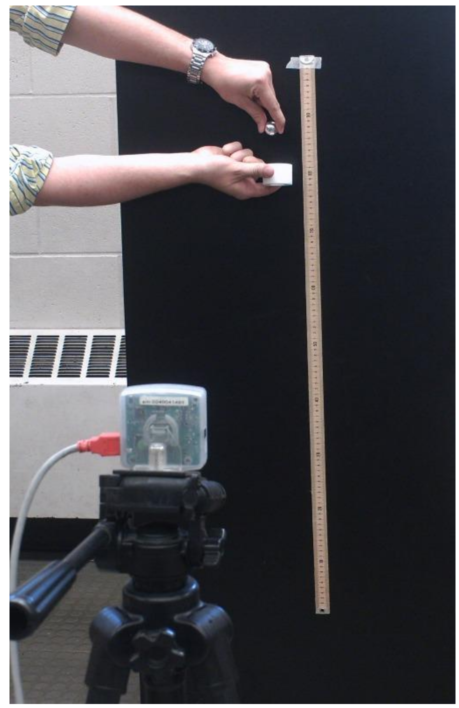
Expect the students to use as many of the different balls as time permits for this first experiment.

The ball-dropping mechanism is a simple grey cylinder with an inner diameter just over 1". These should be used to make each release consistent. Use bubble wrap in a tray to catch the balls. **Do not let them hit the ground**. Damaging the balls will impact the results for the next lab.

Setup:

This lab is essentially the same Lab 2, Problem 1 in the Physics 1301 lab manual. This may be a good resource for preparing for lab or leading discussions.

Make sure that the cameras are set up properly: i.e. short exposure time, large gain. Make sure you know how to adjust all of the camera settings and make sure that your students have set it up correctly.



Tape the meter stick to a surface so that it is completely vertical (the ring-stands work well for this). This will be used for calibrating the length per pixel in the video analysis program. Aim the camera on the tripod so that its line of sight is horizontal and centered on the 50 cm mark of the stick. Also make sure that the camera is focused properly on the ruler and ball. Try to get the entire meter stick in view. You should remind your students that a longer range will produce the best results.

Procedure:

Hold the one of the balls in the plane of the meter stick in the view of the camera. Capture the fall of the ball using the LabView® program, VideoRecorder. Once you have recorded a good video, save it to the desktop of your computer.

Because the balls do tend to stick to your during the release, you may use the dropping-mechanism to consistently drop the ball as vertically as possible; however, using this mechanism means that you cannot capture the position of the ball just as it is released. You will have to account for this in your analysis. Keep in mind that it is important that the ball fall in the same plane as the meter-stick relative to the camera. If not, then the calibration will not represent the actual distances the ball has moved.

Repeat this process for each of the balls.

Data and analysis:

Analysis of the video is similar to that which is done in the 1101, 1201 and 1301 classes, except that you will be using the *Tracker* program instead of LabView®. The advantages to the *Tracker* program are that you can always go back and recalibrate or re-enter data points whenever you want. There are no predictions to enter in, and the data easily exports to a Microsoft® Excel file.

See appendix __ for information about using the *Tracker* program. Students should export the data to Excel and use Excel's fitting function to determine the acceleration of each ball.

They should also note that there may be some random variance in the horizontal direction due to the placement of the cursor, (if the ball had fallen straight down). Make sure that the students are aware of this random error. If the ball had not fallen straight down, they should notice a general trend in the horizontal direction.

Make a plot of the acceleration against the mass of each ball. The results should reasonably show that the effective acceleration of each ball is inversely proportion to its mass.

Notes and Comments:

You may notice for this lab that the parabolas of the position versus time graphs are curved a bit more than they should be. This is mostly due to a wide-angle distortion from the camera lens. The effect is not very important for the 1201 or 1301 labs, but since we are focusing on accuracy and careful measurement, this can affect students' measurements (you can even see the effect on an object moving with constant velocity).

One question that tends to arise is: "Where on the ball should you measure from?" The key here is to be consistent. If the ball is moving downward, use the very bottom of the ball. You know that there should be 1/30 of a second between each frame, so there must be 1/30 of a second between each time that the shutter closes. Using the center of the ball will not be as accurate because the blurred images will not give you a clear position to measure from.

You will not have too many data points, and the best ones will be with the ball moving slowly. With that in mind, the lab could be done by throwing ball upward and capturing the top part of the parabola. A variation of this technique in 1301 involves bouncing a ball off a table. Note, though, that it is more difficult to keep the ball in the plane of the ruler using this method.

Galileo's Experiment 2: Measuring Air Drag and a Better Value of *g*

Purpose:

This lab is a more precise version of the free fall with camera. Students use photogates to time the passage of a dropped ball. The focus here should be on careful setup and measurement.

Materials:

The materials needed for this lab are the following:

- 1 Pasco® aluminum track (2 m in length),
- a c-clamp and block of wood,
- 2 Vernier photogates with thumb-screws and square nut,
- a Vernier SensorDAQ® or LabPro® interface,
- a platform with dropping mechanism and plumb-bob,
- 1 set of 1” diameter balls
    - steel (65.4 g),
    - white plastic (11.5 g),
    - polyethylene plastic (8.8 g),
    - wood (5.0 ± 0.3 g),
    - aluminum (23.3 g),
    - tungsten carbide (126.9 g)
- basket with bubble wrap
- a computer with Vernier Logger Pro® software tool.

Before the students arrive:

- Make sure the 2m tracks are properly clamped to the tables with the bottom end against the floor.

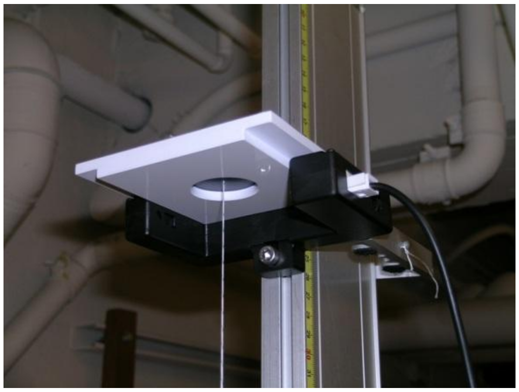
- Make sure that all of the balls fall neatly through the dropping jig. When we were using sand, it tended to get stuck inside the cylinder. That problem should be gone. They should not need to push the ball through the cylinder.

- Install both photogates and make sure that they are attached to the tracks **using the correct hardware**. Note that the students will have to tighten these so that they do not move. **A ball should be able to hit the bottom photogate without it moving**.

- Make sure that the photogates are working and that they are properly set up on the computer.

- Install the jig and plumb bob in the top photogate. You can leave the alignment for the students.

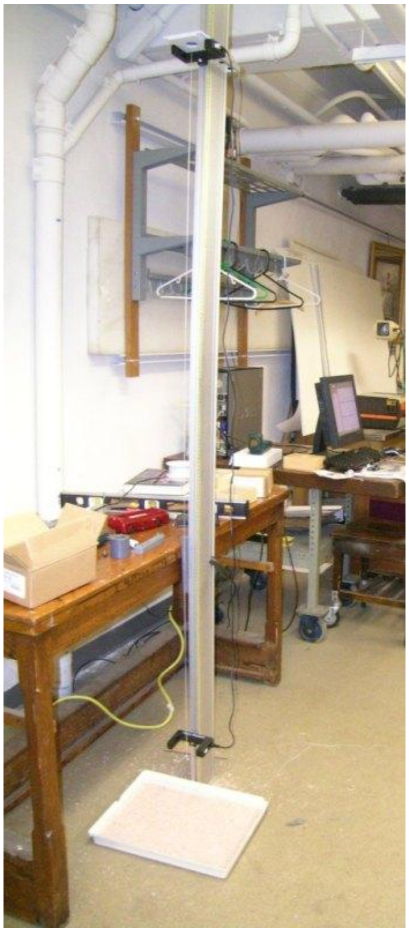
- Make sure the bubble wrap and tray are in place.



The dropping platform/jig (right) was made specifically for this class and for this lab. The platforms are likely permanently attached to the photogates. If not, then simply slide the platform onto the photogate. Double-sided tape should be used to keep the platform from moving during the experiment.

A plumb-bob and cylinder were made to fit into the platform. The plumb-bob will be used to align the photogates in the setup. The diameter of the cylinder is just over 1”. When aligned correctly, this will produce drops that do not miss the gates and are repeatable to about 1 millisecond if your gates are over 1 meter apart.

See the note above about the cylinder, which is essential for getting reproducible drops!



The photogates have two modes: using internal and external light sources. Only use the internal source. You can switch to this mode by unblocking the detector on the inside of the photogate.

Setup:

To begin the setup, mount the track vertically by clamping it to the edge of a table with a c-clamp and block of wood, as shown in the figure to the left. The track should stay in place when the bottom is resting on the floor. This also allows for easy adjustments to the alignment. Place the tray with bubble wrap at the base of the track.

Mount the photogates on the sides of the track with the nuts and ¼-20 allen head screws. Make sure that the gates are attached are very tightly, so that they do not rotate when a ball hits them. Set the photogates as far apart as possible (1.7-1.9 m), and record this separation as accurately as possible. Make sure that the students measure the distance correctly! With the dropping mechanism available, the lab should be done with only two photogates. The dropping mechanism should be attached to the upper photogate (see the figure on the previous page).

The major challenge here is properly setting up and aligning the gates. **Note that this must all be done very carefully to observe the effects of air resistance. For example, for a fixed drag coefficient (using balls of identical diameter), a solid plastic ball will take about 3 milliseconds more to fall 1.5 meters than a steel ball. This type of precision can be achieved with care. This is an excellent lab**. Lean on the students to do it well and they will learn a lot.

Place the plumb-bob into the platform on the top gate. Use this to align the track so that it is vertical and to ensure that the ball will fall directly through the two gates. Both gates should be level.

The alignment of the top photogate is critical, because in order to get the initial velocity correctly, you need to make sure that the center of the ball passes through the top gate (and then the velocity is the diameter of the ball over Δt). The bottom photogate is less critical; just make sure that the plumb-bob passes through the light beam. When aligned correctly, the cylinder will produce drops that do not miss the gates and are repeatable to about 1 msec if your gates are over 1.5 meters apart.

Connect the gates in a daisy-chain mode: each gate connects into the other, and the last one into the Vernier Logger Pro® interface (as shown left). Click on the "Experiment" menu in the Logger Pro® window, and then select "Data Collection." Check the option for "Continuous data collection" and **adjust the sample rate to somewhere around 3000 samples per second (~5000 is the maximum sample rate). If the rate is not set properly, this experiment will not work.**

Procedure:

Use the set of the 1" balls for this lab. These start with the (steel) balls and end with the lightest (hollow plastic) balls. There are also tungsten carbide balls, but these have to be shared. Note that dented balls will have less repeatable drops, so be careful not to let the balls stray and hit the photogates or other equipment. The rubber ball has a noticeable seam and is less repeatable than the others.

Press the "Collect Data" button at the top right of the Logger Pro® window. Carefully drop the balls through the dropping mechanism into the tray below to avoid any bounce. A good drop will result in 4 recorded times (two from each gate – when the ball enters and leaves each gate). Copy and paste the time data into an Excel file.

Repeat this process for all of the different balls.

Data and Analysis:

If you think this through, it is all a matter of determining an effective g from the usual formula:

$$\Delta y = v_i t + \frac{1}{2} g t^2 .$$

Let t_1 and t_2 be the two times from the top photogate. Then

$$v_i = d/(t_2 - t_1),$$

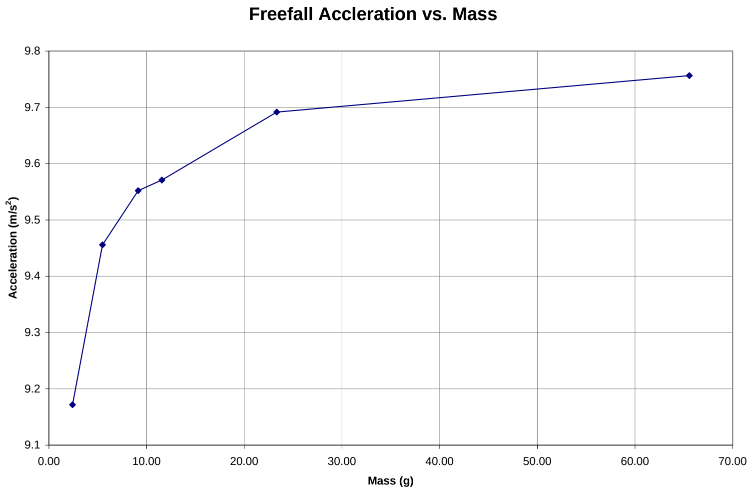
and the drop time t is the difference between the two times when the ball is at the center of each gate:

$$t = \frac{(t_3 + t_4)}{2} - \frac{(t_1 + t_2)}{2} .$$

The students can measure Δy precisely using the scale on the edge of the tracks. It is of course very difficult to determine exactly where the light beam is, but both gates are identical in size, and so they can just measure the distance between the top or bottom edges of the two gates. Then

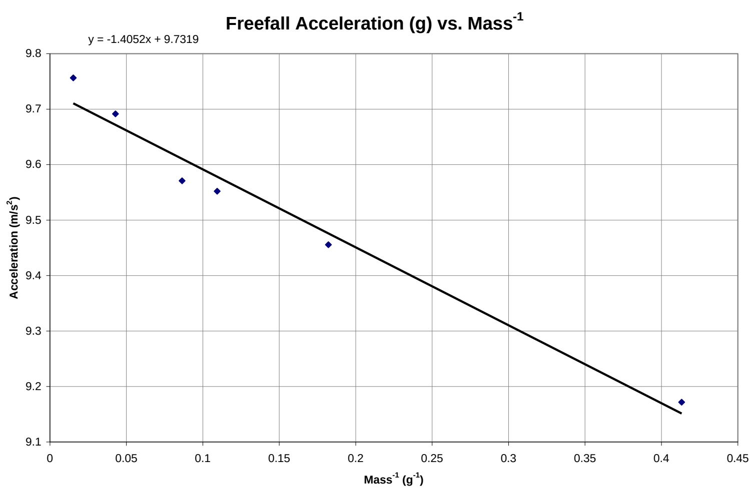
$$g_{eff} = 2\frac{(\Delta y - v_i t)}{t^2} .$$

You should plot the effective **g** of each ball as a function of its mass. If you do, you will see that there should be a horizontal asymptote near the accepted value of 9.806 m/s^2, as shown below.



A plot of the freefall acceleration of each ball as a function of its mass.

A plot of the effective acceleration vs. one over the mass should give a straight line with a negative slope. The fit will not be perfect, since there were some assumptions made, like those in equation 2.12 of the lab manual.



Another plot of freefall acceleration, as a function of the inverse of its mass. They would probably be best leaving off the lightest ball.

Notes and Comments:

The heaviest balls are the best for measuring **g** itself, and the students should start with those. You should be able to get within 1% of the correct g with the steel and tungsten balls. The lightest balls will give an effective g that is about 6 – 7% too small. The students can plot the effective *g* as a function of the mass of the balls.

The second thing that students can look at in this lab is the prefactor in the inertial drag term, assuming that the drag is quadratic in velocity. This can be obtained from the slope of the line when g_{eff} is plotted vs. *1/m*.

The students might be tempted to determine g_{eff} using the kinematic equation $v_f^2 - v_i^2 = 2ad$. They should rapidly find that this approach will not work, because it is virtually impossible to ensure that the ball passes through the lower photogate on a line passing through its diameter.

For further references, check the Physics 4052 lab manual for their ball-dropper experiment, or Kurt Wick's article in the *American Journal of Physics* (volume 67, no. 11, pp. 962 to 965).

Projectile Motion in Two Dimensions

Equipment

- Pasco **short range** projectile launcher with plastic balls
- 1 photogate
- 1 ring stand
- Vernier LabPro computer interface
- 1 desktop computer
- 2 wooden blocks
- Cork board wrapped in a fresh (unwrinkled) piece of aluminum foil.
- measuring tape (Make sure that Sean provides you with enough of these. They are the long tape measures on a wheel.)
- 1 table clamp
- 1 stopwatch

This should be a relatively easy lab for your students. After they determine the muzzle velocity (which does require the computer), they can then disperse into the hall to do the rest of the lab. It is essentially a matter of verifying the range equation. They set up the cannon, position the cork board, and then measure where the ball lands. The velocities are small enough that air resistance is not so important. They do this for several angles and plot the range vs. angle.

In principle they could vary the muzzle velocity, but the geometry of the hallway (ceiling height) does not provide much freedom. The ambitious group might to outside.

I had intended to put something in the write-up on air resistance, and have them use Python to compute the actual trajectory and compare it to the ideal one.

I could use some sample data for this lab.

Collisions and Momentum

<u>Colliding Carts: 1-D Collisions</u>

Purpose:

This is a classic 1-D collision lab. Students will confirm the conservation of momentum in 1 dimension as precisely as possible using the Vernier photogates. This lab could be very instructive in building intuition about the final motion of the carts based on their masses.

For this lab, we intend to use the photogates for measurement, since they are much more accurate. The lab could be done with cameras and video-analysis software, instead, but we find that more time consuming and the parallax effect to be too annoying to deal with.

Materials:

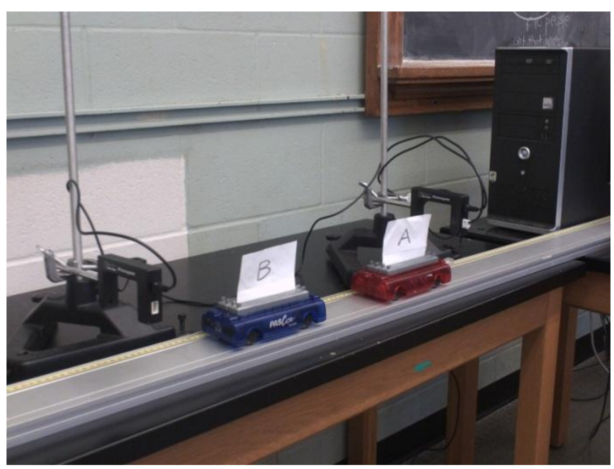
The following materials may be used for this lab:

- a Pasco® aluminum track (2 m in length),
- 2 Pasco® carts (Velcro® and magnetic sides),
- about 4 Pasco® 250 g bricks for the carts,
- 2 Vernier photogates,
- a Vernier SensorDAQ® or LabPro® interface,
- 2 ring stands and clamps,
- a balance,
- a meter stick,
- some scrap paper,
- tape, and
- a computer with the Vernier Logger Pro® program.

Make sure that you have only "good carts!" Bad bearings = Bad data. Sean keeps a nearly infinite supply of new carts. Make sure any bad cart is REMOVED from the lab and replaced!

Setup & Procedure:

There are a number of things that can be done with this lab, but it may be most useful to determine how the final motions of the carts after a collision are affected by each cart's mass (essentially the same as lab 6, problem 1 in the 1301 lab manual). However, you should encourage the students to explore different collisions, whether elastic or inelastic.



To begin, set the track on a level surface. Next to the track, place the two ring stands, each about 1/3 of the track's length from each end. Clamp one photogate onto each of the ring stands such that the infrared beam of the gate is horizontal (as shown to the right).

Tape a square piece of paper with well-cut edges to the side of each cart so that the paper stands vertically. This paper will be used to pass through and trigger the photogates. The paper should be square, and the length of the paper measured so that you can measure the velocity of the carts.

For elastic collisions, position the carts so that both their magnetic sides will collide together. For inelastic collisions, have the Velcro® sides face each other.

With the photogates daisy-chained and the Logger Pro® program open on the desktop, open the "Experiment" menu, and select "Data collection." Check the option for "Continuous data collection" and set the sample rate to somewhere between 500 to 1000 samples per second.

Have one cart begin at rest between the two photogates (cart "B" in the diagram above). Press the "Collect" button at the top right of the Logger Pro® window. Launch or push the other cart (cart "A") from the end of the track toward the stationary cart ("B"). This moving cart should pass through one of the photogates before the collision. After the collision both carts should pass through one of the photogates. Which of the photogates it passes through depends of the masses of the carts and whether the collision is elastic or inelastic. Pay attention to when each cart passes through a gate so that you can double-check whether times were recorded properly.

Data and Analysis:

Divide the length of the paper on the carts by the times it took for each to pass through the gates. This will be the carts' speeds just before and after the collision. Multiply these values by the respective masses to get the momenta.

In each of the cases, check to be sure that the total momentum before and after the collision is conserved. You and students may notice that there is some friction, and so a minimal speed which will produce acceptable results. The goal is to get it within 5%, which is feasible.

You should also be able to predict what the final momenta will be based solely on the masses (m_1 and m_2) and initial velocities (v_{1i} and v_{2i}) of the carts – we are assuming that cart 2 is initially at rest. For inelastic collisions, you only need the conservation of momentum – kinetic energy is not conserved in these collisions. The final velocity, v_f, is

$$v_f = \frac{m_1}{m_1 + m_2} v_{1i} .$$

For elastic collisions, the equations become a little bit more complicated, but you should find that the final velocity for the first cart, *v*1f, is

$$v_{1f} = \left(\frac{m_1 - m_2}{m_1 + m_2} \right) v_{1i}$$

and the velocity of the second cart, *v*2f, is

$$v_{1f} = \left(\frac{2m_1}{m_1 + m_2} \right) v_{1i} \, .$$

Note that the sign of the final velocity of the incoming cart is dependent on the masses of both carts and that it is zero when both carts have the same mass.

Notes and Comments:

As noted above: replace any bad carts immediately.

The photogates may not record the passage of a cart when another cart is blocking the other photogate at the same time. One source of the problem may be a low sample rate. In this case, the solution would be to increase it. Another solution to the timing problem would be to stagger the photogates so that only one photogate is triggered at one time.

2D Inelastic Collisions

Purpose:

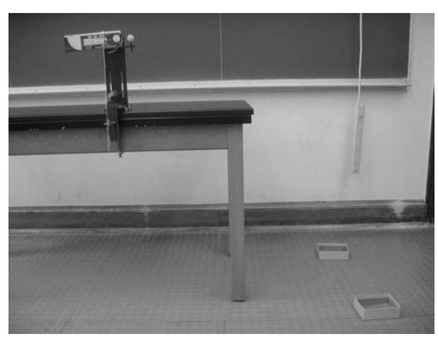
In this exercise, students will verify the conservation of momentum in a precise manner. *Note from 2008: As with many of the other labs, make sure that the students take multiple data points. They can use different mass balls, different angles (from 30 – 60 degrees). Note that the best way to measure the initial velocity of the ball is to use kinematics and not by using a photogate.*

Materials:

For this experiment, students will need to use the following materials:

- a Pasco® projectile launcher (short range) & 2D collision accessory,
- a Pasco® table-clamp or C-clamp,
- 2 yellow plastic 1" diameter balls,
- Other plastic balls can be used as the target
- a Vernier photogate,
- a custom photogate mount for the launcher,
- a Vernier SensorDAQ® or LabPro® interface,
- 2 pieces of cork board with fresh aluminum foil (to determine where the balls land)
- a balance,
- a meter stick and tape measures
- a protractor,
- string, and
- a computer with the Vernier Logger Pro® program.

Setup:



Begin by mounting the projectile launcher on a table with the table-clamp. Attach the 2D collision accessory by mounting it to the underside of the launcher's barrel, and turn it so that it is between 30˚ and 60˚ from the launcher's line of fire (this changes the angle of the collision). One of the balls is meant to rest on the tip of the small tube on the accessory. Use the thumb screws to set the launcher so that it will fire horizontally (the plumb-bob on the side should read 0˚). To reduce any significant recoil from the launcher, be sure that it is securely fastened and does not wobble, even under some reasonable pushing. Aim the launcher towards an open space in the lab room.

Attach the photogate to the launcher with the custom mount. It is essential that the photogate be positioned correctly, or else you will not be able to read the exit velocity of the ball correctly. The most important detail is to have the photogate positioned so that the center of the ball passes through the beam. Also have the infrared beam pass as close to the end of the launcher's barrel as possible. Note from 2008: The launching mechanism is quite repeatable, and hence a good way to measure the initial velocity is to remove the target ball and simply measure how far the incident ball travels. You can then use kinematics to deduce the initial velocity.

We now use pieces of corkboard wrapped in aluminum foil to see where the balls land. Each group should start with a fresh piece of aluminum foil. The impact of the ball will leave a nice dimple.

Use a ruler or a plumb bob to place the protractor directly below where the center of the collision will occur. Tape 3 long strings to the center of the protractor and tie some heavy object to the other end of each string. One string should show the line of the trajectory of a launched ball with no collision. The other two strings will be used to measure the distance each ball traveled and in what direction.

Procedure:

Load one ball into the launcher. The second ball should be placed on the collision accessory. Start collecting data with Logger Pro (if you are using a photogate), and launch the ball. Both balls should have landed in the sand trays.

Copy the time data and the diameter of the ball into an Excel spreadsheet. Measure the distances from the collision to each of the balls, and measure the angles that each ball deflected from the original trajectory.

Data and Analysis:

With the data you have collected, you should be able to extrapolate back and determine the velocity of each ball immediately after the collision. The data from the photogate will tell you the exit velocity of the first ball. *As noted above, it is probably better to skip the photogates and just use kinematics.*

Notes and Comments:

Students had no problems showing that momentum was conserved in the horizontal direction. Problems tended to occur in the direction straight out from the barrel, and there are a couple of sources of error for this. One is that the accessory may be placed to close to the barrel. The deflected ball may end up hitting the barrel again.

Note from Crowell: The problem identified in the previous paragraph is due essentially to the error in determining the initial velocity. When I did this carefully, there was no problem.

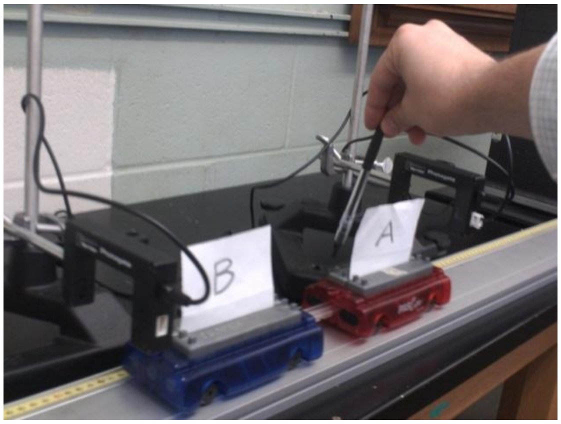
Another source of this error is probably due to spin. However, if setup carefully, you should be able to create reproducible trials. *Spin does become a problem at grazing incidence.*

Explosions & Spring-Loaded Carts

Purpose:

Two carts start from rest in contact with each other. They will be launched apart like in a 1-D explosion (a total momentum of zero). This can be used either as momentum conservation or a spring potential-to-kinetic energy conversion lab.

Setup and procedure:



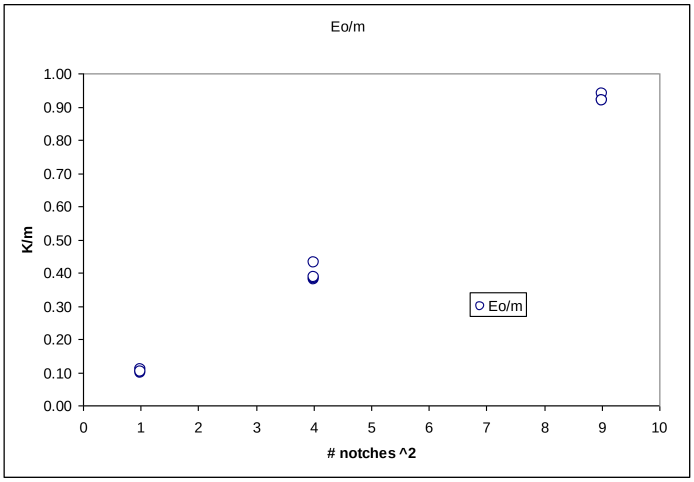
The materials and most of the setup for this lab are identical to the previous collision lab. The two carts start near each other, one of them with its spring-launcher loaded. Both carts can (and should) have various mass blocks on them.

Set the carts between the two photogates and begin collecting data with the Logger Pro® software. To launch the carts, carefully tap on the spring-release button on top of the one cart. What works best is striking it with the flat side of a pen in a careful but fast vertical motion.

Data and Analysis:

Now you can check that the total momentum (which should be zero) is conserved.

If your students are interested, they can look at the energy of the carts after the collisions. By using three different spring levels, you can compute the final kinetic energy of the carts, and check that indeed it goes as number of notches squared. This works pretty well. Additionally, one can check the spring constant by loading the spring with some weight and measuring the sag (keep the release button down).



The graph above shows kinetic energy (in arbitrary units) of the system after separation as a function of square of number of notches used on the spring.

Notes and Comments

We have found that this is a good exercise to remind students that momentum is a vector. Many will automatically say that momentum is not conserved in the explosion because the carts began with no velocity, but then both ended in motion. However, the direction of each cart's motion is important, and it is good to remind the students of this. They should also know that the total momentum will not be affected because the "explosion" is an internal force.

With that in mind, as an additional challenge you may ask the students to loosely attach the carts on their Velcro® sides (with the carts' springs loaded). If done very carefully, they can release a spring while both carts are in motion together, and measure the resulting velocities after they have separated.

<u>Ballistic Pendulum: Ball Capture</u>

Purpose:

This conservation of momentum experiment is also meant to be very precise. Students should take careful measurements in order to verify the exit velocity of a ball from a launcher. This velocity will be measured with a photogate, and then derived from data on the pendulum.

Materials:

The following materials may be used for this lab:

- a Pasco® projectile launcher (long range),
- a Pasco® table-clamp or C-clamp,
- a steel 1" diameter ball,
- a Pasco® "projectile catcher" accessory,
- a Vernier photogate,
- a custom photogate mount for the launcher,
- a Vernier SensorDAQ® or LabPro® interface,
- a ring stand, clamp and extra rod,
- a balance
- a meter stick,
- tape,
- string,
- a video camera and tripod, and
- a computer with Logger Pro®, VideoRecorder and *Tracker* programs.

Setup:

Begin by mounting the launcher onto the table as you had done for the 2-D collisions lab, but aim the launcher over the length of the table. Make sure that the projectile will fire horizontally, and that the launcher is securely clamped.

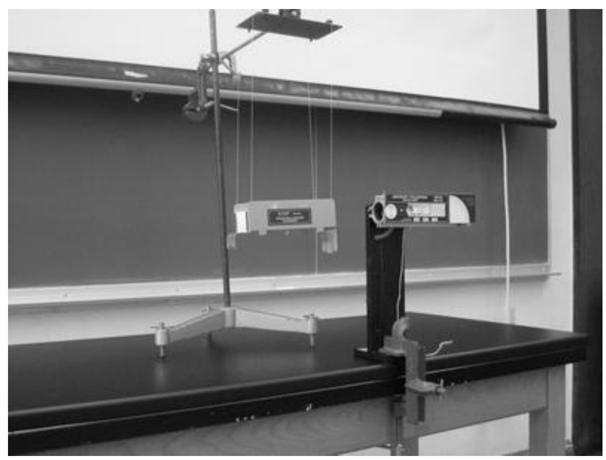
Now position a ring stand on the table close to the launcher. With a rod-clamp, attach one of the shorter rods horizontally from the ring stand about 60 cm or so above the projectile launcher. You will be suspending the catcher accessory from this rod as a pendulum.

The catcher accessory comes with a black plate that has a thumb screw in each corner and a block in the middle. The block is used to attach the plate to the ring stand. Slide it over the horizontal bar and tighten the screw to securely fix it.

Cut a very long length of string- about 3 to 4 m, but this greatly depends on how long you would like the pendulum to be. Loop one end of the string around one of the screws on the plate and tighten the screw to fix the end in place. Thread the string through the small, adjacent hole in the plate, through the front two holes in the catcher, and back up to the corresponding hole on the plate. Then thread the string through the back end of the plate, down to the catcher again, through the holes at the back end, and up to the plate once more. Adjust the pendulum to an appropriate length and then tie the end of the string onto the plate. Using longer string for the pendulum makes the swing of the catcher larger and easier to capture and measure. The catcher should be aligned with the launcher so that the ball can be fired directly between the two without significant changes in the ball's velocity.

Mount the photogate onto the projectile launcher using the custom mount. Connect the gate to the Vernier interface. Place a camera so that it looks onto projectile and catcher, and place a meter stick behind the apparatus for the calibration of your video. Your final product should appear as shown in the figure below.

Procedure:



Load a steel ball into the long-range projectile launcher. Begin collecting data with Logger Pro and VideoRecorder. Your computer may not be able to do both operations at once – in that case, record the speed of the ball with the photogate for several trials to get an average. With the video recording, shoot the ball into the projectile catcher. The catcher, which is now essentially a pendulum, goes through a half-period of an oscillation before it runs into the launcher again. Save the video to the desktop.

Repeat the experiment by using a different setting on the launcher, or adding mass to the catcher.

Data and Analysis:

From the data from the photogate, determine the velocity of the projectile as it exits the launcher. With the video analysis software, you will determine the maximum vertical displacement of the catcher.

From the conservation of momentum and energy, you will be able to find that the final height of the catcher should be

$$h = \frac{v_{balli}^2}{2g}\left(\frac{m_{ball}}{m_{ball} + m_{catcher}}\right)^2 .$$

Rotations

Week 1: 1301 Rotational Dynamics

Equipment:

- Rotational dynamics apparatus (includes spools on shaft, rotating disk, , extra ring, pulley, string, and mass set)
- Meter stick
- Stopwatch (not mentioned in student manual)
- Video camera

The first week of the rotation lab uses the Pasco rotational dynamics apparatus. As always, make sure it is set up properly, but this lab is otherwise straight-forward.   It is true that the 1301 manual spreads this over a nearly infinite number of terribly written pages, but the basic idea is for students to vary the torque or the moment of inertia and see that the laws of rotational dynamics work.

You do want to make sure that the torque being applied is large relative to the frictional torque on the bearings. Essentially, if the mass is too small, the lab does not work well. Given the speed of the cameras, which you are using to record the acceleration of the falling mass, there is plenty of dynamic range.

A debate rages in the pedagogical community about whether students should record the motion of both the falling mass and the rotating platform. This is in my opinion completely silly, provided that the students verify that the string is not slipping.   In other words, it is obvious that the speed of the point where the string is unwinding from the wheel must be the same as the speed of the falling mass.

Week 2: Gyroscope

This is a new lab in 2011, and I will need your cooperation in making sure that it works and in obtaining sample data that can be inserted into this manual. Please email me data. Incidentally, this lab is based on the TeachSpin demo in the lecture-demo area.

Equipment:

- Cue ball with embedded magnet
- Aluminum post and sliding weight that can be inserted into the post on the ball
- Air bearing mounted on a base plate at the correct height
- Fish tank air pump make sure each set-up has hoses and fittings, and that the hose is long enough so that pump can be put out of the way A set of Helmholtz coils in lucite frame (diameter = 22 cm, 150 turns)
- Sorenson DC power supply and cables
- Video camera
- Ruler
- Stopwatch
-

Miscellaneous for room: Sharpie marker, Squid line and some brass nuts to use as a reference point when they are measuring precessions, masking tape, scissorsAs usual, you should check each set-up before the lab. Note that the students do not know anything about magnetism. As far as they are concerned, we are using the magnetic field to provide a variable gravitational torque. To this end:

1. You should make sure that the Helmholtz coils are all wired up properly. If you turn on the supply and the spinning ball does not precess, then one of the coils is backwards (or neither is connected).

2. You should practice to make sure that you know how to get the ball spinning well and how to correct its wobble with a pen or the tip of your finger. Rotational frequencies between 2 and 8 revs/sec. are fine. The precessional frequency is of course much lower. If the air bearing is working properly, the ball should spin for several minutes on its own. If there is a problem, consult Sean Albiston. As of August, 2011, we think we have eliminated all of the bad air bearings.

3. Do not bend the aluminum post. It is fragile.

4. The intent is for the students to measure the spin frequency of the ball with the camera and the precession with the stopwatch. They

must make sure that the frame rate is set to 1/30 second. Yes, they could do it all with the camera, but that will be tedious.

Section 4.2.3 is pretty easy. In (A), they should figure out that the sine drops out. Sections B and C should be no problem. In section (D), they will have to plot $\frac{\Omega}{\omega}\ vs.\ \tau$.

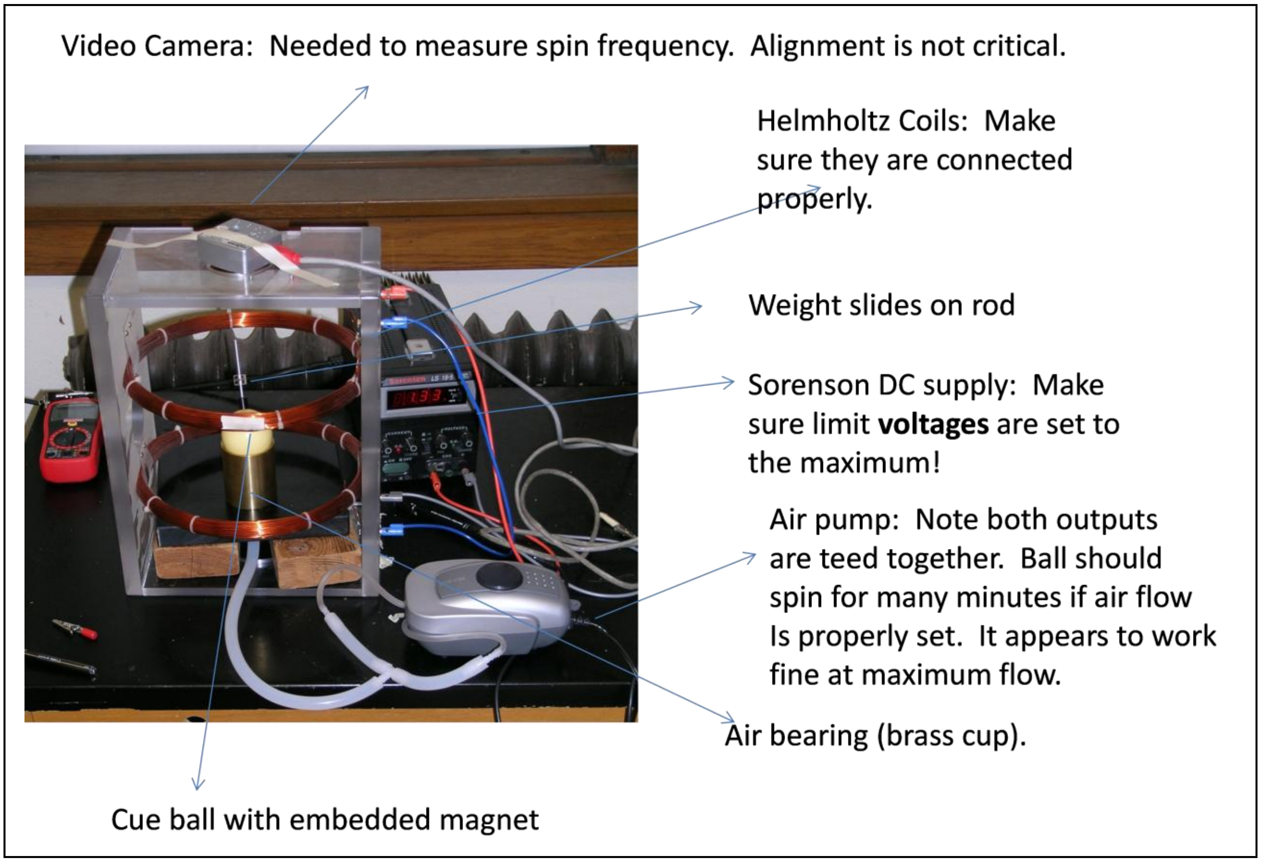
Section 4.2.4 does require them to use the current supply. You might as well set the voltage limits as high as possible and then run it on current control. Just make sure they do not leave it at maximum current for long periods of time.

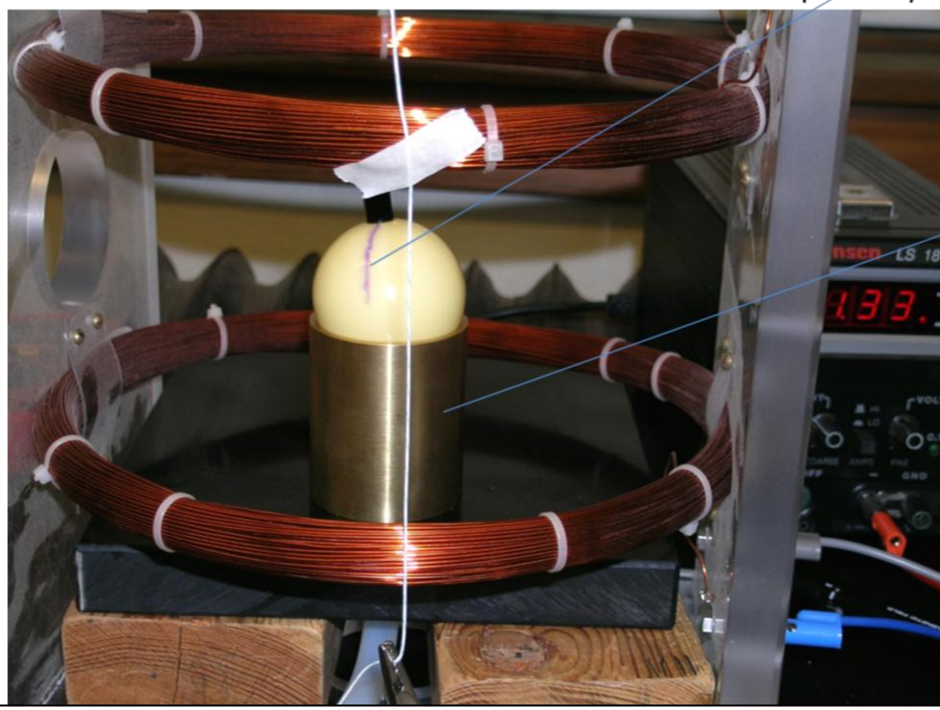
They will have to plot $\frac{\Omega}{\omega}\ vs.\ \tau$.

4.2.5 is relatively easy.

4.2.6 is reasonably advanced, but they can observe the two limits of nutation (with and without loops).

4.2.7 Whether or not you do this section may depend on what is going on in the lecture part of the course. The fact that they can change the sign of the effective gravitational field is instructive. They can also measure the amplitude-dependence of the period.

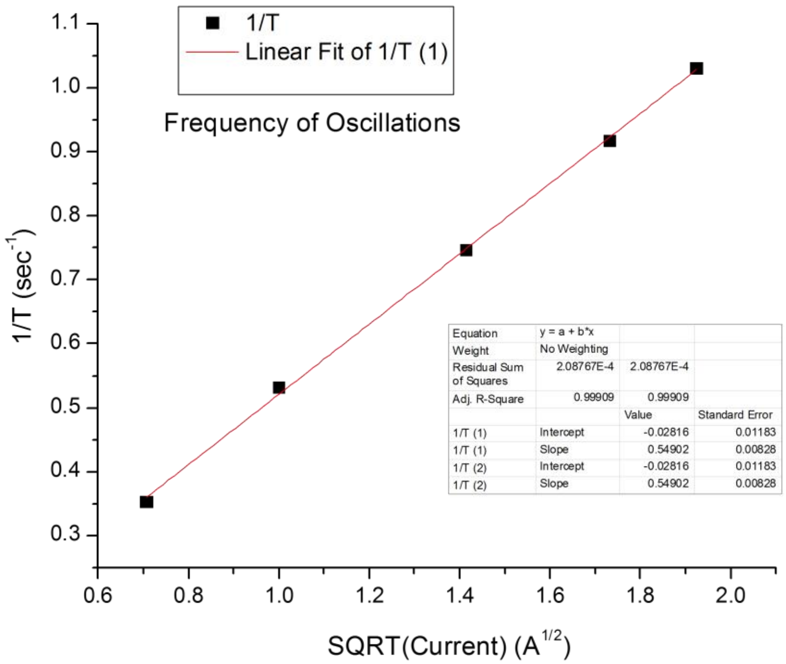




The line on the ball is used in the video to determine the spin rate of the ball. The post that you grab to spin the ball is partially obscured by the tape.

This string is helpful when timing precession cycles.

I need sample data for this lab. Please send it to me.



These are sample data for 4.2.7 (the last part of the lab.)

Gravity and Kepler's Laws

Gravity Simulation

Purpose:

The purpose of this exercise is to create a simulation of two objects orbiting each other due to gravity. Use the simulation to verify Kepler's laws.

Materials:

You will need a computer with *Python* and *V-Python* installed. These and documentation can be found at www.python.org and http://vpython.org, respectively.

Obviously, you must be completely familiar with *Python and V-Python* before the students do this laboratory. You have many weeks to prepare, and so there are no excuses. The lab itself is a simple exercise in numerical integration.

I do not know when Sean is going to "upgrade" the 1401 computers, but double-check with him to make sure everything is installed. Go around and make sure the software runs properly on each machine.

Basics – Programming in Python:

This exercise requires only some very basic programming skills and then learning a couple of very easy commands. Python is a fairly popular object oriented language, but most likely you have exposure to C/C++, FORTRAN or Java. The syntax of Python is very similar to C/C++ and FORTRAN, but if you know any programming language, this should find this straight-forward.

We can't go through a complete tutorial of the Python language. If you are interested, you can pick up any "Learn C [or C++ or FORTRAN] in 21 Days" text, or stop by the book store and pick up one of the texts in the CSci section. Any of these choices are great as far as learning how the syntax works together. Take a look at the tutorials on www.python.org, or check out the documentation on special functions at http://vpython.org/webdoc/visual/index.html. There are also several free tutorials and introductory books available at www.freeprogrammingresources.com. Still, this tutorial should be enough to get by with, and we give you all of the code to make a proper program.

Starting a Program

Begin each program with a header. This header should load any necessary library files to the compiler so that your program can be correctly interpreted. The header that we will use is to load the V-Python libraries, and is given below.

```
from visual import *
from visual.graph import *
```

There are ways to make comments and leave notes for yourself or other programmers. Begin the comment with the "#" symbol. These are very useful and you will notice that we make prolific use of them in the examples to follow. An example of a comment is given on the next page.

```
# This is a commented line.  Only text on this line will be ignored by the
# compiler.
However, text on this line will produce an error because it is not commented.
```

You will need objects to work with in the program, and V-Python is useful for this. You can declare objects such as spheres that have attributes such as position (pos), velocity (vel), radius, color and mass. For example, we can make a spherical object "satellite1".

```
Satellite1 = sphere(pos(-5,0,0),radius=0.1,color=color.blue,mass=1)
```

We will discuss how to manipulate these objects a little later on.

Next is the body of the program. This is where you give commands for the program to do. The compiler reads everything from top to bottom and left to right, so the order that you place certain functions will be important. We will discuss how to visualize or output data from our program, but first let's just talk about making it do some calculations.

Doing Math in Python

Math in Python looks just like it may on paper. You can add, subtract, multiply and divide as usual, and Python follows the usual order of operations.

There are a couple of things that you should be aware of about performing calculations:

A single equals sign (=) is the assignment operator. The variable to the left of it is the variable being assigned the argument on the right. Consider the quick following example.

```
radius = circumference / (2*3.14)
```

The variable "radius" is being assigned the value from the calculation on the right.

Two equal signs (==) is an evaluation of the two adjacent arguments. For example, the statement "radius==circumference" evaluates whether the two values are equal or not. If the equality is true, then the function will return the value of "1", and "0" if untrue. Similarly you can evaluate inequalities (<,>,<=,>=). Usually you will use these in loop functions or gates.

Python reads everything from left to right. Don't worry about your parentheses – again order of operation still holds true here, and it will do the calculations correctly. However, you cannot assign the value 12 to the variable "x" written as the statement below.

12 = x

This simply won't work. The variable being assigned must be on the right, and the value on the right. A correct statement should be like that below.

x = 12

Another set of common operators that you may use are the power, square root and exponential operators. Trigonometry operators such as the sine, cosine and tangent functions are the same as you would write on paper, and their inverse functions are simply noted by putting an "a" in front of the original operator. Examples of these are shown in the table on the next page.

Function	Code	Output
Exponential	exp(x)	e^x
Logarithm	log(x,base)	$Log_{base}(x)$
Power	pow(x,y)	x^y
Square root	sqrt(x)	$\sqrt{x}$
Sine	sin(x)	Sin(x)
Arcsine	asin(x)	$Sin^{-1}(x)$

V-Python has made a very useful set of objects, operators and functions that make a lot of the tedious programming routines easier, primarily for vectors.

To create a vector you need a name for the vectors (say v1 and v2), and then some components (x,y,z). The statement below shows how you can initialize or assign two vectors.

```
v1 = vector(1,2,3)
v2 = vector(4,5,6)
```

These vectors obey normal scalar addition, subtraction and scalar multiplication with the usual symbols (+,-, and *). Below are examples of the cross- and dot- product operations.

```
cross(v1,v2)
dot(v1,v2)
```

A couple of other useful operations are ones that give you the magnitude (length) of the vector (v1 in the example below), and another that gives the angle between two vectors (v1 and v2), shown respectively.

```
mag(v1)
v1.diff_angle(v2)
```

That's it. This is really all the *math* that you will need to complete the program. Now we will discuss the theory we need to implement, and then how to go about programming it.

Theory:

The programs that you and your students will create will demonstrate the motions of objects in *elliptical* orbit over time.

We will consider two objects with masses *M* and *m*. Suppose that the mass M is fixed in space, and the second mass, m, is initially a distance *R* away from the first. In order for the smaller mass to not fall into the larger one, we must give it some initial velocity v_o. For simplicity, we'll make this velocity perpendicular to the line that passes through the two masses.

You may wish to have your students start with circular orbits – they should be quite accustomed to these by now, and it will make the programming process go a lot easier. The condition for the circular orbit is that the centripetal force experienced by the orbiting object is equal and opposite to the gravitational force.

$$-\frac{GMm}{R^2} = m\frac{v_o^2}{R},$$

where *G* is the gravitational constant. Solving for v_o, and ignoring the minus sign, we see that

$$v_o = \sqrt{\frac{GM}{R}}.$$

This situation is fairly trivial for students at this level, and we don't want them to crutch on this too much. We are trying to simulate actual *elliptical* orbits so that they can confirm Kepler's laws for in a more general and sophisticated way.

Of course we already know that to control our virtual massive objects we will have to use forces and kinematics in our program, but to make elliptical orbits it's best to use energy to describe what is really going on.

The potential energy, *U*, of the smaller object, *m*, at a distance *R* from the larger object is given by

$$U = -\frac{GMm}{R}.$$

The negative sign is a consequence of the gravitational force being attractive and that we chose the potential energy to go to zero as mass *m* moves infinitely far away. The total energy of the system, *E*, is the potential energy plus the kinetic energy:

$$E = \tfrac{1}{2}mv^2 - \frac{GMm}{R}. \qquad \text{(a)}$$

One can now see that for circular orbits, the total energy is

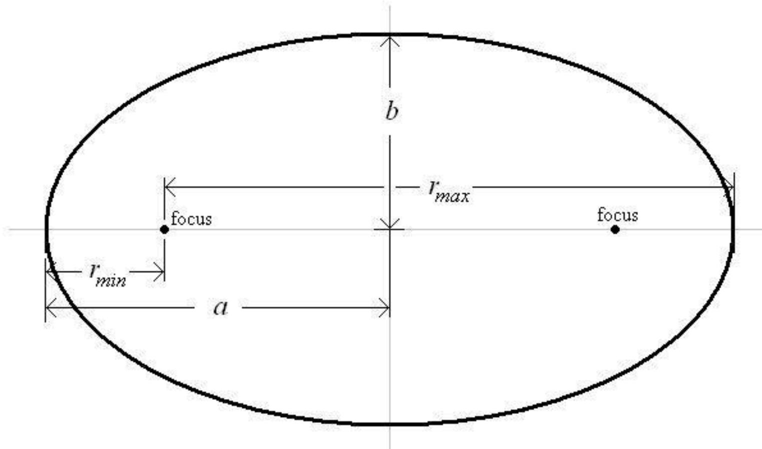
$$E_{circular} = -\frac{GMm}{2R}.$$

Again, we are more interested in *elliptical* orbits. The condition for making an elliptical orbit is that the total energy must be greater than that for a circular orbit, but less than zero:

$$-\frac{GMm}{2R} < E < 0.$$

If we substitute equation (a) for *E*, and solve for the velocity squared, we find that

$$\frac{GMm}{R} < v^2 < 2\frac{GMm}{R}. \qquad \text{(b)}$$



This is all fine and dandy, but this is not the only condition – we still need to know about the initial conditions and parameters of our elliptical orbits. Consider the diagram to the right. The ellipse has two axes – the major axis with a length 2*a*, and the minor axis with a length 2*b*. There are also two points labeled as the foci. Two distances that will be important to us are the maximum distance or apogee (r_{max}) and minimum distance or perigee (r_{min}) from one focus.

A characteristic value of an ellipse is its eccentricity, *e*. This is given by

$$e = \frac{r_{max} - r_{min}}{2a}.$$

With a little bit of algebra, and the fact that $r_{max} + r_{min} = 2a$, then

$$e = 1 - \frac{r_{min}}{a}. \qquad \text{(c)}$$

The eccentricity can be used to describe circles, ellipses and parabolas; however, for an ellipse, the eccentricity always must have a value between zero and one:

$$0 < e < 1. \qquad \text{(d)}$$

Finally we are ready to make our elliptical orbit. If we begin our simulation with the sun at a focus of the ellipse and the smaller mass at the perigee of the orbit, then using equations (b) and (c) we find that the velocity, v, of the orbiting object must meet the condition that

$$\frac{GM}{a(1-e)} < v^2 < 2\frac{GM}{a(1-e)}. \qquad \text{(e)}$$

Therefore, to make the elliptical orbit, you will only need to *specify* a few things before you begin running your program:

- the masses of your two objects (m and M),
- the initial distance between the two objects (r_{min}),
- and the eccentricity of the orbit (e).

You can use equation (c) to determine the value of a, and be sure that you still satisfy the condition in equation (d).

Implementation:

Now we can actually start writing our program.

Start out with the header discussed earlier. Then you will need vectors to describe your object's positions, velocities, and accelerations. You may even want to add in angular momentum for each of the objects, since you will need them in the near future. A vector for the center of mass may be helpful, too. You will have to determine many of these values and vectors beforehand using equation (e) and the list of specifications on the previous page.

You will need scalar variables such as those that describe distances, mass, time or any constants. The example below was made to give you an idea of how to go abut doing this – it is not complete for the program.

```
# Begin with the header
from visual import *
from visual.graph import *

# Now declare the two objects for the satellite and sun
satellite = sphere(pos(-100,0,0), radius=1, color=color.blue, mass=1)
sun = sphere(pos(0,0,0), radius=100, color=color.red, mass = 10000)

# Give the satellite attributes like velocity and acceleration and distance from
# the sun
satellite.vel = vector(0,0,0)
satellite.acc = vector(0,0,0)
distance = sun.pos-satellite.pos

# Initialize scalar variables and constants for the time steps (dt), number of
# steps (maxsteps), the current step, (tstep), total time that you can count up
# (tTime), and G.
dt = 0.01
maxSteps = 6000
tTime = 0
tStep = 1
G = 6.67
```

```
# Make any blank lists that you need by making an array – specify this with "[]"
posArray = []
```

It's best to declare and initialize all of your variables at the beginning of your program. There will be some variables that you will have to specify in order for the simulation to work, such as the initial separation of the objects, velocities, and masses.

Next you will have to calculate the acceleration of the satellite, and its displacement in the small time, dt, as in the example above. These calculations can actually be quite simple. Note that you don't need to keep writing "vector" each time. Once a vector is initialized, only need to refer to it by name. If you want a specific attribute of an object, add a period and name of the attribute after the object's name. For instance, to recall the position of the satellite, simply type "satellite.pos". See the example syntax below.

```
# Calculate the acceleration of the satellite
satellite.acc = (G*sun.mass)*distance/pow(mag(distance),3)

# Now use kinematics to determine new position of the satellite after time dt
satellite.pos = satellite.pos + satellite.vel*dt + 0.5*satellite.acc*dt*dt
satellite.vel = satellite.vel + satellite.acc*dt
distance = satellite.pos-sun.pos
tStep = tStep + 1
tTime = tTime + dt
```

It is important to remember that for every time that you make a calculation, you will have to update all of your variables that change with time.

To do these steps over and over, you will need the help of loops. There are a couple of choices of the kind of loop you may use, but to keep things simple we will use the "for" loop. This command loops through everything indented below it for as many times that you specify. In the example syntax below, we repeat the commands described above as long as the variable "tStep" is between the numbers 1 and 6000 (specified by "maxSteps").

```
for tStep in range(1,maxSteps):
        # Calculate the acceleration of the satellite
        satellite.acc = (G*sun.mass)*distance/pow(mag(distance),3)

        # Now use kinematics to determine new position of the satellite after
        # time "dt"
        satellite.pos = satellite.pos + satellite.vel*dt + 0.5*satellite.acc*dt*dt
        satellite.vel = satellite.vel + satellite.acc*dt
        distance = satellite.pos-sun.pos
        tStep = tStep + 1
        tTime = tTime + dt
```

```
    #You can add data to an array, such as the x-position
    posArray.append(satellite.pos.x)

# The next comman or any below it will not be repeated in the loop because it is
# not indented.  Only the commands above will be repeated.

print posArray
print "completed"
```

The last command here will print the values in the array to another screen. The arrays are useful for filling with data you will want later on, and even groups of numbers can be stored as one unit if you put then inside another set of parentheses, such as (12,552), for instance. Other things can be printed in the window as well, such as text that is enclosed in quotations, such as the word "completed" above.

Visualization:

The "print" command is one way to generate an output from the program, but it is a bit limited. There are a number of things that will help see what is going on.

For one, you can make a trailing curve that will draw out the path of an object in the 3-D output screen. The attribute for the object is called "trail", and you simply need to give it a color and update it as you do other variables.

```
# This is how you will initialize the trail array
satellite.trail = curve(color=satellite.color)


# To see the trail, the trail variable must contain all of the previous positions of
# the object.  Just tack on the new position using the "append" command (the
# trail variable stores this in its own "pos" attribute)
satellite.trail.append(pos = satellite.pos)
```

Its nice to see the orbs go around in the animation, but we are really interested in the data. To make a graph of the position of your satellite, you need to initiate a plot, preferably right after you initialize your variables near the beginning of the program. The following syntax will do the trick.

```
# Setup graph (axes, size, labels, ranges and colors)
trajplot = gdisplay(x=0, y=0, width=800, height=400,
                    title="y vs x", xtitle="x position",
                    ytitle="yposition", xmin=-5.5,
                    xmax=5, ymin=0, ymax=25,
                    foreground=color.black, background=color.white)
```

```
# At the end of the program, when you are ready to plot your points, such as
# those in the array "posArray", type in the following command.
xyplot = gdots(pos=posArray, color=color.blue)
```

For a plot with two axes, the data in the array must be grouped by parentheses. You can easily group data, such as the x- and y- positions of the satellite, in the array as shown in the following.

```
posArray.append((satellite.pos.x,satellite.pos.y)
```

Data and Analysis:

From your simulations you will need to verify Kepler's laws of orbits, areas and periods.

The Law of Orbits

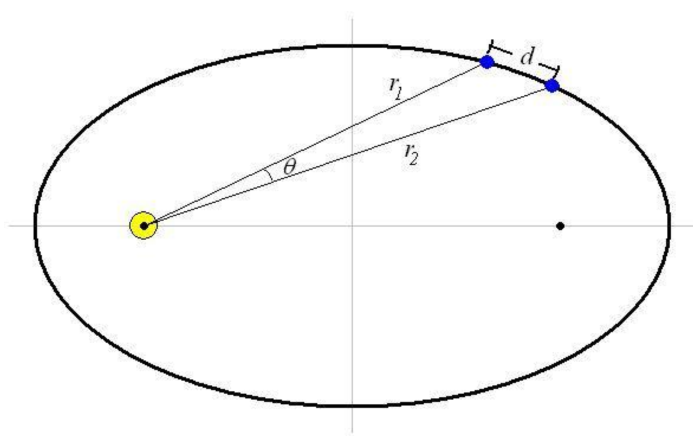
Have the students mess around with the initial velocities and positions in their programs. Use equation (b) and see what happens when you don't satisfy the inequality or the condition for circular motion. Have them note that the only stable orbits draw out an ellipse or a circle. This is a simple qualitative establishment. The trick is finding quantitative data to verify their predictions.

One of the easiest ways to do this is to find the satellite's distance from each focus of their ellipse (if that is indeed what the orbit is). The sum of these two distances should be equal 2*a*. You may even have them determine the values of *a* and *b* in the equation for an ellipse:

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1.$$

The Law of Areas

This section is tricky because the students need to calculate the area for each displacement, and some of them may be fixated with the arched ends of each section of the orbit as shown in the figure to the right. We can simplify the problem by choosing small displacements (which means small measures in time).



To determine the area of each small triangle swept out in a small time, *dt*, we will have to assume that the distance the satellite has moved is in a straight line, and not an arc. The displacement can be found from your usual kinematic equations, but the easiest way to find the area is to compute the cross product of the two vectors, r_1 and r_2. If you wish, you can find the angle, θ, between the two distances as shown above. The area, A, of the small triangle is

$$A = \tfrac{1}{2}\left\| \vec{r_1} \times \vec{r_2} \right\| = \tfrac{1}{2} \cdot r_1 \cdot r_2 \cdot \sin(\theta).$$

In theory, if you look at the conservation of angular momentum you should be able to show that the area swept out, dA, in a small period of time, dt, is equal to the angular momentum of the satellite, L, divided by twice its mass, m:

$$dA = \frac{L}{2m} dt.$$

This is the value you should try to confirm with your program.

The Law of Periods

This section may take some time because it involves running several simulations. Each simulation should consider a new orbit – different mean distances from the sun and/or different speeds at the perihelion. Have the program print out the time to complete one revolution around the sun for each of the simulations. You should also record the length of the semimajor axis, a, of each ellipse that you draw out. Make a plot of the period squared against a cubed. This should be a linear relationship.

Angular Momentum

To confirm the conservation of angular momentum, have the students store the satellite's angular momentum over each iteration in their programs along with the associated times. After they run the simulation, have them plot the angular momentum as a function of time. It should be a reasonably flat line. Of course the law of areas *is* conservation of angular momentum.

Precession of the perihelion

My advice is for them to take their working program and just add a small ϵ/r^3 term to the force. They will clearly see the precession of the perihelion. This is a good one to do.

Binary stars

Of course they can just change coordinates and replace the mass with reduced mass and claim they are done! The intent is for them to program it without changing coordinates and then see that both orbit the center of mass.

Tides

I am not sure anyone has gotten this far….

Oscillations

Damped Free Oscillations

Purpose:

This lab explores damped simple harmonic motion. The students can investigate the dependence of the frequency on mass, measure damped oscillations, and measure the relationship between the frequency and the damping constant. The damping constant is varied by changing the number of magnets on the bottom of the cart.

Materials:

The following materials may be used for this lab:

- 1 Pasco® aluminum track (2 m in length),
- a Pasco® cart,
- 1 to 4 small NdFeB magnets,
- a power supply (up to 12V),
- 2 springs,
- a Vernier Sonic Ranger® motion detector and a piece of stiff cardboard to mount on the cart (to reflect sound)
- a Vernier SensorDAQ® or LabPro® interface,
- string,
- tape, and
- a computer with Logger Pro®.

Setup:

Set up the cart between the two end stops, with springs attached to both sides of the cart. **Check the motion sensor at each table before the students arrive. As always, if there are problems check the data collection rate. Also, remember that the sound needs to be reflected off of a card mounted on the cart. Sometimes the sensor picks up a reflection from something on the next table.**

One of the problems with this lab is that students (and TA's who take sample data) do not record enough cycles of motion. Try to obtain data over two time constants. They need to be able to make a log plot of amplitude vs. time.

Note that the magnets provide a nearly perfect linear damping mechanism due to eddy currents.

Procedure:

1. First, verify that the resonance period depends on mass in the correct way and that the damping is small. You should be able to observe many

oscillations. Although not mentioned in the instructions, the students should vary the mass. Remember to check for bad carts.

2. Mount **one** of the magnets on the bottom of a cart by placing it over a metal screw. Pull the cart back some distance (about 30 to 40 cm), and begin collecting data with motion detector and Logger Pro®. Release the cart and observe the free decay of oscillations of the cart. **You will need to tape a piece of cardboard or a playing card onto the cart for the motion detector to work properly.**

Save your data to an Excel file, and determine the frequency of the oscillations.

Determine the value of *b*, the characteristic value of the damping constantby making a log plot of the amplitude as a function of time. The slope will be (-*b*/ 2*m*), where *m* is the mass of the cart. In principle, you can also determine the shift in the resonant frequency

$$\omega = 2\pi\nu = \sqrt{\frac{k}{m} - \left(\frac{b}{2m}\right)^2} = \sqrt{\omega_0^2 - \left(\frac{b}{2m}\right)^2},$$

where ω_o is the natural frequency of the spring-cart system. In practice, shifts in the resonant frequency due to damping will be small.

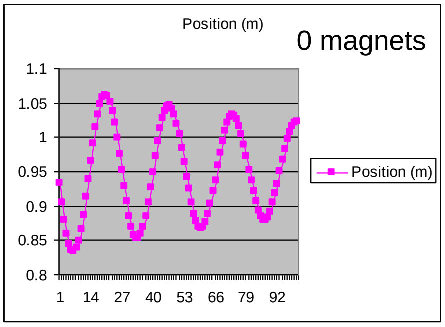
Have students add more magnets and predict what the new frequency should be. You should also encourage them to vary the mass of the cart for a fixed number of magnets.

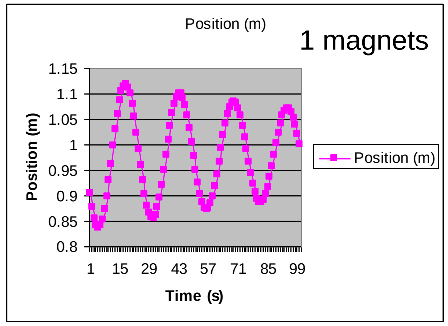
Data and analysis:

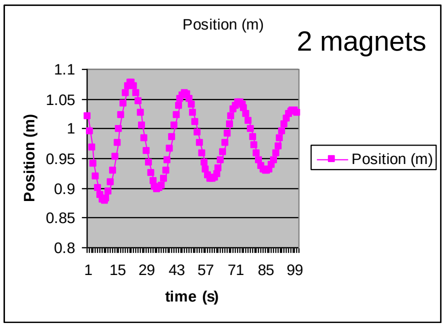
With the data students have already, they can make a plot of the frequency against the number of magnets on the cart, or against the mass of the cart for a fixed number of magnets. They should also plot their predictions alongside of the recorded values.

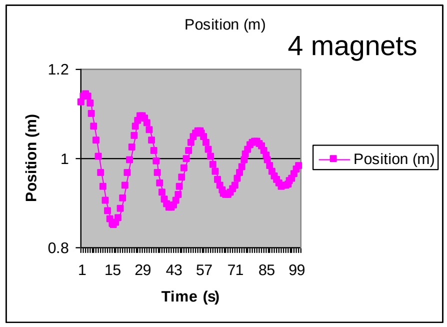
They should determine the characteristic damping time for each of the trials **This may be their first experience with a logarithmic plot!** They can then plot the damping time as a function of the number of magnets.

Some data. Note that this TA should have set up Logger Pro to record many more cycles – at least one 1/e time worth of data.









Notes and Comments:

This lab could be useful for practicein error analysis. Some of the students who are more on top of things should be able to determine the standard deviations in their data, and possibly come up with a decent value of *b* and an uncertainty.

<u>Driven simple harmonic motion</u>

At a minimum, students should plot the resonance curve as a function of frequency for at least two values of damping. With a little bit of thought, they can also measure the phase.

Note that the instructions for measuring the phase here are different than those in the student lab manual. In fact, the scheme suggested in the student manual is not very practical.

It is best to start with some damping and a reasonable mass on the cart. This keeps the resonant amplitude from getting too large and the damping time from being too long.

Materials:

The following materials may be used for this lab:

- 1 Pasco® aluminum track (2 m in length),
- a Pasco® cart,
- about 4 Pasco® 250 g bricks for the cart,
- a Pasco® mechanical oscillator/driver & mount
- a power supply (up to 12V),
- 2 springs,
- a few NdFeB magnets,
- a Vernier photogate,
- a Vernier Sonic Ranger® motion detector,
- a Vernier SensorDAQ® or LabPro® interface,
- string,
- a ruler,
- tape, and
- a computer with Logger Pro®.

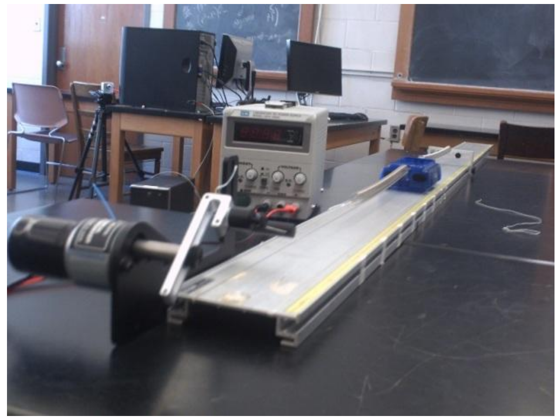
Setup:

Begin by placing two end-stops on the track about 150 to 180 centimeters apart. Place a cart between the end-stops on the track, and attach each end to an end-stop with the springs. You should add a stiff piece of cardboard onto the cart to reflect the ultrasound back into the motion detector.

Although one can map out the resonance without any damping, the Q is rather high and hence the ringdown time is long. **Add one or two magnets to the bottom of the cart.**

The students should determine the natural frequency by looking at the free oscillations (see last week's lab). They can add mass to bring the frequency down into a convenient range.

We have traditionally used motors to drive the system (set up as shown below). It is easy to adjust the amplitude at low frequencies.



Once you are done with the procedure above, remove one of the end-stops and replace it with the mechanical oscillator/driver. The motor set should be placed at the end of the track so that it can freely move. Thread a piece of string through the hole in a guide cylinder on the oscillator's mount and tie it to the arm of a linear DC motor. Tie the other end of the string to the spring connected to the cart. Be sure that the spring is stretched to approximately the same length as it was with the end-stop. The total setup should look like that in the figure above.

The oscillators should be driven with our standard 18V DC power supplies, but the **motors are only designed or 12V DC**! Make sure students do not exceed this voltage. The frequency increases with increased voltage, and the amplitude is set by the (adjustable) arm.

The mechanical drivers act as the *forcing* mechanism for our system. The strength of this force is directly proportional to the length of the arm on the motor. Usually an arm length of 2 cm works out well, small amplitudes work best.

Place the Sonic Ranger at the opposite end of the track and plug it into the interface. To measure the phase, you should place a photogate behind the motor. Position this so that the arm of the motor passes through the beam of the photogate. You may have to play around with getting the cords to the interface.

Procedure:

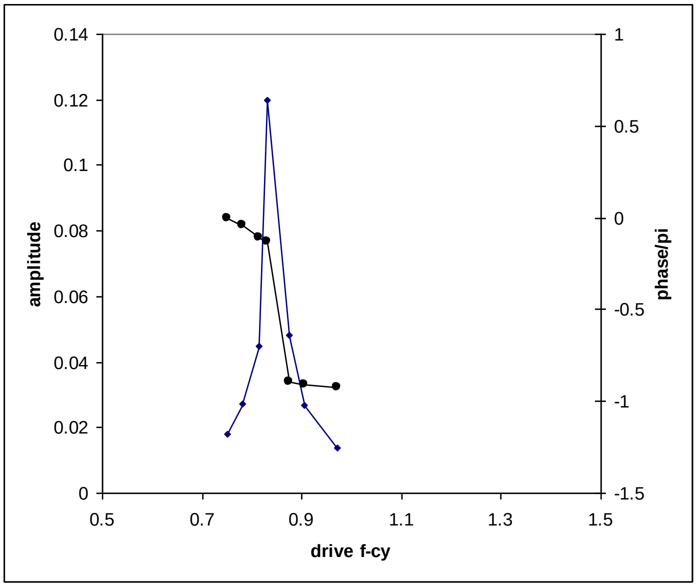
The lab consists of mapping the resonance curve. Have students change the frequency of the driving motor. Allow about 20 seconds (i.e. a few time constants) for the cart to settle at each frequency. Obviously the most interesting data for this lab will be near the resonant frequency.

Record the motion of *both the rotating arm and the cart.* You should know how long the arm is and what frequency it rotates at. Logger Pro® should be able to record data from both the photogate and motion detector simultaneously. Analyzing motion of the arm gives the drive frequency and phase. There will of course be an arbitrary phase shift set by the position of the arm, but have them focus on how the phase changes as a function of frequency. Analyzing motion of the cart gives the oscillator frequency, phase and amplitude**. However, it is usually simpler to measure amplitude using the scale on the side of the track**. *If they are having difficulty, do not let them get too hung up on trying to measure the phase. They can just observe what it is at three points (below resonance, on resonance, above resonance).*

Repeat this procedure for several frequencies. Be sure to cover the three primary cases: $f<<f_o$, $f=f_o$, and $f>>f_o$. It works best to begin at lower frequencies and steadily increase. You should be taking more data points when closer to the resonant frequency.

Data and Analysis:

Map all three of the oscillator (cart) frequency, amplitude and phase as a function of the drive-force frequency. Of course the frequency is trivial. The amplitude should go through a well defined resonance curve. The (relative) phase exhibits a well defined slip. This is one of the rare labs where phase relation between the drive/oscillator at resonance can be made obvious. Having points far from resonance is important in order to observe the full zero-to pi swing.



This data is from a simple run using video analysis, with fits done by eye. It shows amplitude and relative phase as the system goes through resonance. This one had very little damping; Q is somewhat larger than 15. **2008: With the new system, you can take much more data – and at different damping. Add magnets until there is enough damping to make a measurement without the cart flying off the track at resonance. Note that the fastest way to read the amplitude is to use the scale on the side of the track.**

At least a few cycles have to be analyzed to extract the frequency and phase. **I am looking for some good sample data, and so make sure to take some and send it to me. Logger Pro should show both the SHM of the cart and the photogate output. The phase is determined by**

measuring zero crossings of the SHM relative to an "edge" of the photogate signal.