Laboratory Exercises for Scripting and Programming for Modeling and Simulation

Copyright © Bruce Char
2008-10
Laboratory Exercises for Scripting and Programming for Modeling and Simulation

All rights reserved. Permission is given to Staff and students of CS 121, Fall 2010 to reproduce these notes for their personal use only.
# Contents

Acknowledgments ........................................................................................................................................ v

1 Lab 1 CS 121 Computation Lab I Fall 2010 Directions and Problems ................................................................. 1
   1.1 Overview .............................................................................................................................................. 1
   1.2 Introduction to Lab 1 (20-25 minutes) ................................................................................................. 1
   1.3 Problems -- Part 1 (20 minutes) ......................................................................................................... 1
   1.4 Maple TA (20 minutes) ...................................................................................................................... 4
      Notes on Maple TA ................................................................................................................................. 4
   1.5 Problems -- Part 2 (30 minutes) ..................................................................................................... 5
   1.6 Saving your work (5 minutes) .......................................................................................................... 5
   1.7 Final actions (End of class) ............................................................................................................. 6

2 Lab 2 Cs 121 Computation Lab I Fall 2010 Directions and Problems ................................................................. 7
   2.1 Lab 2 Overview .................................................................................................................................. 7
      Overview ............................................................................................................................................... 7
      General directions for this lab ........................................................................................................... 7
   2.2 Instructor's demonstration of word processing, scripting ................................................................... 8
   2.3 Part 1 ................................................................................................................................................ 8
      Problem Description ........................................................................................................................... 8
   2.4 Part 2 .............................................................................................................................................. 12
      Part 2 Description ............................................................................................................................. 12
   2.5 Final actions (end of class) ............................................................................................................ 13
   2.6 Concluding remarks ....................................................................................................................... 13

3 Lab 3 Cs 121 Computation Lab I Fall 2010 Directions and Problems ................................................................. 15
   3.1 Lab 3 Overview .................................................................................................................................. 15
      Overview ............................................................................................................................................... 15
      Directions for this lab .......................................................................................................................... 15
   3.2 Instructor's demonstration .............................................................................................................. 16
   3.3 Part 1 ................................................................................................................................................ 16
      Part 1 Description .............................................................................................................................. 16
   3.4 Part 2 ................................................................................................................................................ 18
      Part 2 Description .............................................................................................................................. 18
   3.5 Attachment: starter script for Part 1 ................................................................................................. 20
   3.6 Final actions (end of class) ............................................................................................................ 23
   3.7 Conclusion ....................................................................................................................................... 23

4 Lab 4 Cs 121 Computation Lab I Fall 2010 Directions and Problems ................................................................. 25
   4.1 Lab 4 Overview .................................................................................................................................. 25
      Overview ............................................................................................................................................... 25
      To prepare for this lab beforehand .................................................................................................. 25
      Directions for this lab .......................................................................................................................... 25
   4.2 Instructor's demonstration of definition of functions, advanced plotting, and animation .................. 26
   4.3 Introduction to the "Human Cannonball" simulation ........................................................................ 26
   4.4 Problem 1 ......................................................................................................................................... 30
   4.5 Problem 2 ....................................................................................................................................... 30
   4.6 Problem 3 ....................................................................................................................................... 30
   4.7 Problem 4 ....................................................................................................................................... 32
   4.8 Final actions (end of class) ............................................................................................................ 33
   4.9 Conclusion ....................................................................................................................................... 34
   4.10 Acknowledgements ....................................................................................................................... 34
Acknowledgments

To our colleagues and families, who supported us in trail-breaking.
To our students, who learn how to work with the new and different.
1 Lab 1CS 121 Computation Lab IFall 2010 Directions and Problems

1.1 Overview

This lab introduces the use of Maple, the primary computer language used for this course. You will learn how to do simple arithmetic calculations, as well as annotated plots. A ecology management problem is introduced that can be solved with the calculational facilities introduced.

This lab also introduces Maple TA, the primary homework/quiz/exam site for the course. You will log onto Maple TA with your personal account, and taking a practice quiz. Starting next week, there will be required and graded work on Maple TA for you to do.

1.2 Introduction to Lab 1 (20-25 minutes).

The instructor will introduce themselves and present a brief overview of course, Maple, and the lab.

The lab staff will hand out verifications sheets along with paper copies of these directions. In later weeks, these directions will be posted on-line and can be read from your lab computer. The verification sheets will still be passed out, to be the permanent record of your attendance and accomplishments during the lab.

1.3 Problems -- Part 1 (20 minutes)

1. Sit down with your lab partner and if you haven't previously met, introduce yourself to them. Write both of your names down on the verification sheet in the space provided.

2. All of the partners should log onto a computer, following the demo given by the instructor in the introduction.

3. Do the calculations below. Everyone should try doing the computations on their own computer. To gain more confidence that you are getting the right answer, look at what your partners are getting. Get their help if they appear to be more successful than you. Sometimes just talking about what problems you are facing may produce useful insight towards overcoming them. If there is a problem that you can't collectively resolve, call the lab staff over and get some help.

4. You are to do all of the steps below. Some of the answers should be transcribed onto the verification sheet as indicated, for grading by the staff. Have a staff member come over to sign the verification sheet for part 1. Be prepared to show your work to the staff member, and to explain how you got your answers. This is also the opportunity to clear up any questions or uncertainties you may have even after doing the work.

5. When you complete part 1, get a staff member to verify your work before moving onto part 2.

1.a) Get Maple to calculate the sum of 2+2. Presumably you will be able to tell whether or not you got the right answer pretty easily.

b) What is exact fraction you get from adding together $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$? What about the sum of $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$, $\frac{3}{2}$, $\frac{4}{3}$, and $\frac{5}{4}$? Note that if you are doing a calculation that is highly similar to a previous one, cutting and pasting can save you some effort entering the second expression.

2. Use Maple to perform the following exact calculations. To enter π, you can select the letter from the Common Symbols palette on the left hand side of the Maple window (it's a few segments below the Expression palette). Note that Maple does not regard Π as the same as π. To enter e, the base of the natural logarithm, use the $e^x$ from the expression palette, or the e from the "Common Symbols" palette. Typing "e" from the keyboard unfortunately does not produce the same result -- that kind of e Maple will regard as an symbol for an algebraic unknown like x or y.

a)
b) \[
\sin\left(\frac{\pi}{3}\right) = \]

c) \[
\sqrt[\ln(e^8)]{e^8} \quad \text{(You should get 8)}
\]
d) \[
\sqrt{1 + \frac{2}{5} + \frac{3}{\left(\frac{15}{13}\right)}}
\]
(You should get \(2\))
e) \[
\log_{55}\left(\sum_{i=0}^{10} i\right)
\]
(You should get \(1\))

3. A state lottery allows you to pick six numbers from the numbers from 1 to 52 to win. Maple exact arithmetic to calculate the exact odds of winning. This can be done by using the "choose" function from the expression palette: \(\binom{a}{b}\) means "the number of ways you can choose \(b\) things from \(a\) things". For example, if the lottery asked you to pick three numbers from the numbers from 1 to 6, the chances of winning would be 1 out of \(\binom{6}{3}\)

\[
\binom{6}{3} = 20
\]

4. Calculate \(2^3\). Note that \((2^3)^4 = \)

\[
\frac{1}{2} + \frac{1}{3} \cdot \frac{47}{42} = \frac{2 \cdot 3 \cdot 5 \cdot 13}{2 \cdot 3 \cdot 5 \cdot 13}
\]
4. Why doesn't Maple give that as its answer?

5. Get Maple to reproduce this plot.

\[
\log_{10}\left(\left|\sin\left(\frac{1}{x^2 + 1}\right)\right|\right)
\]

6. You should use the right-click->plots->Plot Builder menu to specify things such as the plot range, the plot color, etc. Get Maple to reproduce this plot exactly, including the color, the line type, the proper horizontal and vertical ranges and labels, and correct title and caption.

\[
(x - 1) \cdot (x - 2)^2 \cdot (x - 5) \cdot e^{-\frac{x}{10}}
\]
1.4 Maple TA (20 minutes)

1. The instructor will give a brief demo of how to use Maple TA, including how to log in, and how to take simple quizzes. (5 minutes)

2. Take Maple TA quiz 0. (10 minutes).

Notes on Maple TA

1. Maple TA is a quiz-administration system running separately from Blackboard Vista and Drexel One. Your userid should initially be your Drexel One userid (e.g. egk23) and the password should be your Drexel student ID number (e.g. 10096739). Note that the password is probably Drexel One password. You can change your Maple TA password after you log in.

2. The address for Maple TA will be given in class. Links to it will also appear on the class web site www.cs.drexel.edu/cs121/Fall2009 as well as the class site on Blackboard Vista, under "Maple TA".

3. After logging onto Maple TA, you need to select the correct class, and then the correct test to take. Usually your choices will be limited, but the choices may change during the term depending on the need.

4. After you have finished answering all the questions, you should hit the "Grade" button so that your score is recorded. If you don't do this this, Maple TA will record your answers but you will receive no credit for your work because your recorded score will remain at 0.

5. If you encounter any technical difficulties, you should contact the course staff by visiting the Cyber Learning Center (University Crossings 147) or on-line in the Blackboard class discussion group. If you have questions about the grading of an Maple TA assignment, you should contact your section instructor (the person listed in the schedule of courses).

6. The quiz server will only handle 150 simultaneous users and will turn away the excess, so don't wait until the last moment to take the quiz. You will be given credit for only that part of the quiz that you finish before the deadline.

7. If there is a catastrophic system failure, the deadline will be adjusted. An announcement will be made on Blackboard and the course website.
1.5 Problems -- Part 2 (30 minutes)

Complete part 1 problems if you haven't finished. Then work on part 2 of Lab. Get verification.

1. Find the exact solution to \( 3 \cdot x + 5 = 0 \).

2. Find the exact solution to
   \[
   3 \cdot x^2 + 24 \cdot x + c^2 = 5
   \]
   (solve for \( x \)).

3. Given your answer to 2, determine values of \( c \) that make the solution for \( x \) a real number, not a complex quantity. This means that the solution for \( x \) won't involve any imaginary numbers. (Hint: experiment with using solve on an inequality. We haven't told you about this, but like many things in Maple, what should work often does. You can find inequality symbols under the Common Symbols palette. See if you an also figure out how to enter inequalities from the keyboard!)


A breeding group of 20 bighorn sheep is released in a protected area in Colorado. It is expected that with careful management the number of sheep, \( N \), after \( t \) years will be given by the formula:

\[
N = \frac{220}{1 + 10 \cdot (0.83)^t}
\]

and that the sheep population will be able to maintain itself without further supervision once the population reaches a size of 80.

(a) Graph \( N \) versus \( t \).
(b) How many years must the state of Colorado maintain a program to care for the sheep? Show your work.
(c) How many bighorn sheep can the environment in the protected area support? (hint: examine the graph for large values of \( t \)).

1.6 Saving your work (5 minutes)

1. The instructor will demo how to save a Maple worksheet file, and how to upload the work to Blackboard (occasionally required for some labs).

2. Save your work into a .mw file. The resulting file should show up on your Desktop, although it depends on your computer's notion of current working directory. If you have problems finding the file on the Desktop and your partners can't help you, call over a staff member.

After saving the file, upload a copy of the file to Blackboard so that you can refer to it later on. (Most public computers at Drexel automatically wipe out all files created during a student session after the student logs out.)

Ask a lab staff member for a demo of this if they haven't done it already.

You can also send yourself a copy via email as an attachment. This is good for those who want to remember how they did things, or wish to look at the worksheet again after lab. If you upload the file to Blackboard, you can download it to your home computer from there.
1.7 Final actions (End of class)

1. Before you leave, get the staff to grade, sign, and collect the verification sheet. You don't get credit for the lab unless they have a score recorded for you in a signed verification sheet that they have at the end of lab. You may leave the lab after you do this. You can get partial credit for the lab if a portion of your work is verified.

2. Final grades for the course will be curved if necessary, so don't fret excessively if you don't finish but it looks like others are in the same shape. However, you should try to learn the material you don't complete in lab so that you can pass the quizzes and be ready for the next lab. Computer work at this introductory level introduces a lot of ideas and concepts that appear pervasively in subsequent work. The plus side is that you'll probably see next time more of what you worked on this time, so you'll have another chance to practice and improve. The down side is that you can't ignore tough details and hope that they won't matter much. If you don't get it all and it still appears mysterious after you review things on your own after class, come with your questions to the Cyber Learning Center (University Crossings 147) next week during office hours and talk to the consultants there.
2 Lab 2 Cs 121 Computation Lab I Fall 2010 Directions and Problems

2.1 Lab 2 Overview

Overview

This lab practices the development of re-usable multi-step scripts to solve a problem.

Before beginning lab work, you will also see the instructor demonstrate the word processing features of Maple worksheets. You will also see a demonstration of how to assign names to parameters, and how to execute a whole script as a single block through Edit → Execute → Worksheet or Selection.

Part 1 of the lab has you apply a given script to solving several versions of a problem. You will find that constructing the first version of the script is the labor-intensive portion of the work. Solving the second and third versions of the problem should be very quick, since it involves only copying and a little editing of the numerical values used for the parameters of the problem.

Part 2 has you developing your own script and applying it. You must do three "original" things: a) figure out how to create a script that solves one version of the problem, b) identify the parameters of the problem, c) edit the script to use the parameters (if you haven't done so already), and d) apply your script to the other versions of the problem by cutting and pasting.

To develop the script for Part 2, you will need to come up with a recipe for how to get the solution as a sequence of Maple operations involving assignment, solving, plotting, and taking limits as well as work through the standard difficulties of getting the proper information and instructions into the computer. Script development is something where you should expect to succeed after trial, experimentation, and troubleshooting.

General directions for this lab

1. Form a lab team of two or three members. You should all sit on the same side of your work table. You need not use the same partner as last time. Group members should introduce themselves to each other if they haven't already met.

2. Listen to the instructor's overview of this week's new concepts and Maple features.

3. Get your copy of the verification sheet and check out which parts of the work will be verified.

4. Work on part 1 (30 minutes), and part 2 (50 minutes). We would like to see everyone end up with individual copies of the solution scripts.

5. Once you have finished your work, save a copy of your work (email it to yourself, or something similar), and get the staff to sign off on the verification sheet. Having the work verified means that you can demonstrate to the staff how to solve the lab problems and can get the work done. In general, showing up at the start of lab with a completed copy of the lab work will not result in verification, since that doesn't demonstrate that your team can do the work.
2.2 Instructor's demonstration of word processing, scripting

Scripting and script-building

After this, the instructor will quickly review the work required in the scripting portion of the lab. It is expected that you will have read Chapters 3 through 5 of the course readings before coming to lab and are already familiar with the assignment operation (what you get through "assign to name" in the clickable menu, or by entering := from the keyboard), the concept of parameters in a script. If you've worked through the examples given in chapter 5 of a script and how to build one from a problem description, you should find the work in the lab straightforward.

2.3 Part 1

In this problem, we ask you to create a worksheet to solve a version of a falling body problem. The problem gives you a formula relating elapsed time to velocity. From this formula, you can calculate information such as the terminal velocity achieved by the body, and the amount of time it takes to achieve a certain percentage of terminal velocity. The Maple worksheet you will write will set up the formula and then perform the calculations needed to provide the desired information.

Problem Description

1. We want to solve three versions of a problem.

<table>
<thead>
<tr>
<th>Version 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A package of medical supplies is dropped from a helicopter straight down by parachute into a remote area. The velocity $v$ (in feet per second) of the package $t$ seconds after it is released is given by the equation $v = 24.61 \left(1 - e^{-1.3 \cdot t}\right)$.</td>
</tr>
<tr>
<td>(a) Graph $v$ versus $t$. (b) What is the horizontal asymptote of the graph? (c) How long does it take for the package to reach 98% of its terminal velocity?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A different package (with a different aerodynamic configuration) is dropped from a helicopter. The velocity $v$ (in feet per second) of the package $t$ seconds after it is released is given by the equation $v = 27.47 \left(1 - e^{-1.1 \cdot t}\right)$.</td>
</tr>
<tr>
<td>(a) Graph $v$ versus $t$. (b) What is the horizontal asymptote of the graph? (c) How long does it take for the package to reach 87.5% of its terminal velocity?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A different package (with a different aerodynamic configuration) is dropped from a helicopter. The velocity $v$ (in feet per second) of the package $t$ seconds after it is released is given by the equation $v = 22.47 \left(1 - e^{-1.47 \cdot t}\right)$.</td>
</tr>
</tbody>
</table>
(a) Graph \( v \) versus \( t \). (b) What is the horizontal asymptote of the graph?
(c) How long does it take for the package to reach 47.3\% of its terminal velocity?

a) Do File -> New -> Document to get a fresh blank Maple worksheet. At the top of the document, insert the names of your group members, your lab section, your lab instructor's name, and the date/time. Then enter the following sequence of commands to solve version 1 of the problem. This portion of the work is "type it in and make sure that you get the same effects as the demo example shows".

For verification on this part, you should be able to identify the parameters of the problem and explain why they, and not other variables in the script are parameters.

<table>
<thead>
<tr>
<th>Lab 2, Problem 2.1, Version 1 Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 121</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>(Insert Group info, section info, date info here.)</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>Version 1</td>
</tr>
</tbody>
</table>
| A package of medical supplies is dropped from a helicopter straight down by parachute into a remote area. The velocity \( v \) (in feet per second) of the package \( t \) seconds after it is released is given by the equation \( v = 24.61 \left( 1 - e^{-1.3t} \right) \).

(a) Graph \( v \) versus \( t \). (b) What is the horizontal asymptote of the graph?
(c) How long does it take for the package to reach 98\% of its terminal velocity?

Solution
Define the basic relationship between time and velocity. We use parameters \( a \) and \( b \) to represent the coefficients in the equation.

\[ 24.61 \]
\[ \text{assign to a name} \rightarrow \]
\[ a \]
\[ 24.61 \]
\[ -1.3 \]
\[ \text{assign to a name} \rightarrow \]
\[ b \]
\[ -1.3 \]

We use the parameter \( p \) to represent the percentage of terminal velocity that we want to hit.
\[ .98 \]
\[ \text{assign to a name} \rightarrow \]
\[ p \]
Enter the expression for velocity and assign it the name *velocity*:

\[ a \cdot (1 - e^{b \cdot t}) \]

Assign to a name *velocity*.

Plot this expression to better understand it. We do this by entering the name of the expression for velocity, and then right-clicking on the result to order up a plot. Generate the plot through right-click → *Plots* → *Plot builder* so that you specify the axes labels, plot range, etc. After the plot has been generated you can change/fix any settings by right-clicking on the plot and operating the pop-up menu, which works similar to but slightly differently from the Plot Builder.

The horizontal asymptote is the limit as \( t \) goes to infinity of the right hand side of the equation. Don't worry too much if the mathematical notation for getting the asymptote seems unfamiliar; the important thing to note is that Maple can figure it out if you learn how to fill in the "lim" template from the Expression Palette.

\[
\lim_{t \to \infty} \text{velocity} = 24.61000000
\]

Assign to a name *terminalVelocity*. 

\[
p \cdot \text{terminalVelocity}
\]
Setup the target equation that equates the fractional velocity to the velocity expression, and solve it numerically. The latter "solve" is done using the right-click → Solve → Numerically Solve.

\[ fractionTerminalVelocity = velocity \]

\[ 24.11780000 = 24.61 - 24.61 \ e^{-1.3 \ t} \]

\[ \text{solve} \]

\[ 3.009248466 \]

This value (2.10) is how many seconds it takes the falling body to attain \( p \rightarrow 100 \) % of terminal velocity.

b) Once you have the script for Version 1 working, save the worksheet from part a) as yourNameLab2Part1-1.mw (recall from last week that this is by doing Save As from the File menu). Then save another copy of the work under a different name, by saving the worksheet as yourNameLab2Part1-2.mw. This allows you to do new work while retaining a copy of the original work.

Now change the parameter values in your worksheet to configure the script to solve version 2 of the problem. Edit the other textual information in the worksheet to reflect the second version.

Execute the new version. Check that it solves version 2 of the problem (how will you do that?). Save this second version in yourNameLab2Part1-2.mw.

In order to get credit for this part, you have to be able to show to the graders that all you did to the Version 1 script from part a) was to edit the value of the parameters, and then executed the whole worksheet. You shouldn't need to retype any of the formulas, the solve or plot commands, etc.

c) Send your version of the result for part b) to one of your partners. They in turn should send you a copy of their file. Since this part depends on all team members being at more or less the same place, team members should help each other out to get synchronized at this point.

Once you receive your partner's worksheet, open it up on your machine and edit it so that it will solve Version 3 of the problem. Highlight the sections that you changed or fixed in red. You may need to fix the commentary as well as the code if you find spelling or grammatical mistakes, inaccuracies, or unfinished sections.

Save this script as yourNameLab2Part1-3.mw. You can mail back the altered version to the partner you got Part1-2 from, so that they can see what changes you felt you needed to make.
The normalized amplitude $A$, of the vibration of a door panel of an automobile is found to be

$$A = \frac{1}{\sqrt{1 - \left(\frac{\omega}{\Omega_f}\right)^2 + \left(\frac{\omega}{\Omega_f}\right)^2}}$$

where $c$ is a measured constant that depends on the car, $\omega$ is the number of revolutions per second of the motor, and $\Omega_f$ is the measured frequency of vibration of the door panel in cycles per second.

Consider the following three versions of the problem.

<table>
<thead>
<tr>
<th>Version 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>We find that for a 2009 Camaro (yellow, of course), $c = 0.15$, and $\Omega_f = 20$ Hz.</td>
</tr>
<tr>
<td>(a) Display a reasonable graph of engine speed (in &quot;rpm&quot;, or revolutions per minute) versus amplitude.</td>
</tr>
<tr>
<td>(b) Calculate the speed of the engine (in rpm) for which the normalized amplitude is 2.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>We find that for a 2003 Mini Cooper, $c = 0.18$, and $\Omega_f = 25$ Hz.</td>
</tr>
<tr>
<td>(a) Display a reasonable graph of engine speed (in revolutions per minute) versus amplitude.</td>
</tr>
<tr>
<td>(b) Calculate the speed of the engine for which the normalized amplitude is 2.7.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>We find that for a 1974 Mercury Marquis (black), $c = 0.11$; and $\Omega_f = 15$ Hz.</td>
</tr>
<tr>
<td>(a) Display a reasonable graph of engine speed (in revolutions per minute) versus amplitude.</td>
</tr>
<tr>
<td>(b) Calculate the speed of the engine for which the normalized amplitude is 1.5.</td>
</tr>
</tbody>
</table>

In a fresh document enter a script similar in style to that of Problem 1, to solve Version 1 of this problem. Be sure to include your name and the names of your other group members in the worksheet that you create.

Use the := operation to handle assignment of parameters in this part. You may use the "assign to a name" operation from the clickable menu to do assignment to non-parametric results, though.

One thing that you will have to think through before you start typing and clicking is how to handle the requirement that the information you are given and wanted in revolutions per minute when the formula you are given is using revolutions per second.

Another thing you will need to work on is how to establish the plotting ranges. This will not be done automatically for you since the software is not sophisticated to know what portion of the graph you'd find interesting to look at. (In other words, they haven't invented mind-reading computers yet.) We suggest experimenting with ranges until you find something satisfactory.

Save your Version 1 as myNameLab2Part2-1.mw.
Make a copy of your working script in myNameLab2Part2-2.mw and edit it to handle Version 2 of the problem. You should find that the work involved to convert the script to handle Version 2 of the problem is by editing the values of the parameters. Execute the script and check that it solves Version 2 correctly.

When you have Version 2 working, send a copy of that file to one of your lab partners, and get a copy of their version of the script from them in return. Save this as myNameLab2Part2-3.mw, and edit it to handle Version 3. Highlight the sections that you changed or fixed in red. You may need to fix the commentary as well as the code if you find spelling or grammatical mistakes, inaccuracies, or unfinished sections. You can mail back the altered version to the partner you got Part2-2 from, so that they can see what changes you felt you needed to make.

We are told that the normal operation of car engines leads them to operate in the 1500-2000 rpm range while cruising. Do you think that the drivers of these three cars will be satisfied with the vibration properties of their cars?

Do you know what TV shows or movies were these cars seen in? (No Maple, doesn't have a button which will answer that.)

2.5 Final actions (end of class)

Upload all of your work to Blackboard, or email copies to yourself and/or your partners. Before you leave, make sure that the staff has signed and accepted the verification sheet for your group so that your work is properly credited.

2.6 Concluding remarks

In this lab, you have learned how to create scripts that combine commentary and a sequence of computational actions. Scripting allows you to easily solve second and third versions of a problem once you have done the hard work of creating the script by studying how to the first version of the problem.

Because programming is a relatively expensive activity in terms of time, reusing someone else's work is normal activity in computing. Scripts need to be written in a way that makes it easy for someone else to use it and to modify it in modest ways. Of course there will be other times where you'll have the responsibility of creating something completely on your own instead of reusing someone else's work. Just keep in mind that you're writing not only for yourself but potentially for others.
3 Lab 3 Cs 121 Computation Lab I Fall 2010 Directions and Problems

3.1 Lab 3 Overview

Overview

This lab practices more with the development of scripts. This time you develop scripts using the "textual style" for operation entry where everything is specified from the keyboard, rather than the "calculator style" where expressions are created by filling in slots from Palette options, and operations are selected by the mouse.

Maple extends the style of mathematical functions -- f(x), g(3, 5), etc. -- to specify not only mathematics, but also computations. Thus solve, and plot are written textually as functions solve(...), and plot(...).

To add to the functional frame of mind, we also explore how you can create and name your own functions in Maple.

In addition to providing practice with the textual/functional style of programming, this lab has you use additional parts of Maple: lists and strings.

The jump to textual entry of an entire Maple operation takes some getting used to, but the greater flexibility and power of expression is needed to do sophisticated technical problems.

Before beginning lab work, you will also see the instructor demonstrate how to use these elements. This should be a recapitulation and time to clarify your own explorations with these features before lab.

In Part 1 of the lab, you apply a given script (written in textual form) to solving various versions of a problem. You then are presented with a variation of the problem and asked to figure out how to modify the script to solve the variant. This will require a bit of original thought -- it can't be handled just by changing parameter values.

In Part 2, you will study how to do data fitting -- find a formula that provides a mathematical description of measurements collected. Once you have the formula, you can use it to answer additional questions and make predictions about the situation that was measured. This part also requires the use of textual operations, as data fitting is not available under the clickable menu.

Directions for this lab

1. Find a lab partner. You need not use the same partner as last time. Group members should introduce themselves to each other if they haven't already met.

2. Listen to the instructor's overview of this week's new concepts and Maple features.

3. Get your copy of the verification sheet and check out which parts of the work will be verified.

4. Work on part 1 (30 minutes), and part 2 (40 minutes). We would like to see everyone end up with individual copies of the solution scripts. However, it may be more efficient this time to work as a pair at a single computer, and distribute the scripts to both partners via email or other file exchange at the end.
3.2 Instructor's demonstration

Textual entry of Maple commands, lists, sets, and plots

The instructor will demonstrate:

How to invoke solve and plot with textual entry of the function, including the use of lists and strings. "Command completion" (entering solve and then hitting the escape key) will be demonstrated as a typing shortcut. Ways of troubleshooting your way out of problems with textual entry will also be discussed.

Use of functions and defining your own functions

The instructor will review the available functions in Maple, where to read more about them, and how to define your own. Use of a user-defined function in a script will also be shown.

Data-fitting

The instructor will demonstrate the data fitting facilities in Maple. Part 2 is basically "read the on-line documentation and experiment with the examples until you get them to work".

3.3 Part 1

Part 1 Description

From notes on Time Constants, ENGR 101 Fall 2009 (week 3)

A capacitor connected to a battery charges according to the following formula

\[ V(t) = V_i + (V_{max} - V_i) \left( 1 - e^{-kt} \right) \]

where \( V_i \) is the starting voltage, \( V_{max} \) is the final voltage, and \( k \) is the reciprocal of the time constant \( k = \frac{1}{\tau} \). The voltage is measured in volts, the time in seconds.

Here is a problem that can be solved by a Maple script:

<table>
<thead>
<tr>
<th>Problem A, Version 1</th>
</tr>
</thead>
</table>
| For a particular capacitor/battery set up, we find \( V_i = 63 \), \( V_{max} = 266 \), \( k = 0.09 \).
| (a) What is the value of \( \tau \), the time constant? |
| (b) What is the voltage after 20 seconds? |
| (c) What is the voltage after 40 seconds? |
| (d) How long does it take the voltage to reach 256 volts? Plot the voltage curve and a horizontal line whose intersection gives a visualization of when this happens |
| (e) What percentage of the difference in voltage has the capacitor been charged to after \( \tau \) seconds? |
Directions, Part 1

1. Download the file Lab3Part1Script.mw and open it.

2. Do Edit->Execute->Worksheet and execute the worksheet. Verify that it's working correctly by comparing the output to what you believe is the correct answer.

3. Modify the worksheet to run the following version of the problem:

   **Problem A, Version 2**
   For a particular capacitor/battery set up, we find $V_i=37$, $V_a=251$, $k=0.0075$.
   (a) What is the value of $\tau$, the time constant?
   (b) What is the voltage after 60 seconds?
   (c) What is the voltage after 120 seconds?
   (d) How long does it take the voltage to reach 220 volts? Plot the voltage curve and a horizontal line whose intersection gives a visualization of when this happens
   (e) What percentage of the difference in voltage has the capacitor been charged to after $\tau$ seconds?

   After you make your changes to the worksheet's parameters, you can do Edit->Execute->Worksheet to run the whole thing again. Verify that the worksheet is still working plausibly.

   **Save version 2 as YourNameLab3-A2.mw.**

4. Here is a similar but different problem. Huddle with your group and figure out how to solve the problem. Everyone in the group should then modify their previous script to solve this new problem. You will need to do more than change the parameter values, you will have to change the instructions that occur in the script. The modifications revolve around figuring out how you find $k$. That in turn will cause you to change which parameters the script has. Because of the great similarity between the problems, it's easier to edit a copy of your original script than to type things in all over again.

   This happens all the time in programming -- once you have something successful, you run into variations that require reprogramming rather than just re-execution of a fixed script.

   **Problem B, Version 1**
   For a particular capacitor/battery set up, we find $V_i=37$, $V_{max}=251$.
   (a) We find that after 25 seconds, the voltage has risen from its initial value to 72.0 volts. What is the value of $k$?
   (b) What is the value of $\tau$, the time constant?
   (c) What is the voltage after 40 seconds?
   (d) What is the voltage after 60 seconds?
   (e) How long does it take the voltage to reach 220 volts? Plot the voltage curve and a horizontal line whose intersection gives a visualization of when this happens
   (f) What percentage of the difference in voltage has the capacitor been charged to after $\tau$ seconds?

5. **Save this script as YourNameLab3B-1.mw.** Lab partners should exchange their scripts and move onto the next problem.
6. Check out your partner's script by using it to solve the problem below. Verify that they've done an adequate job of parameterizing by observing that all that is needed to solve B version 2 is to change values of some of the parameters and re-execute the script.

**Problem B, Version 2**

For a particular capacitor/battery set up, we find $V_i = 30$, $V_{max} = 274$.

(a) We find that after 30 seconds, the voltage has risen from its initial value to 101.0 volts. What is the value of $k$?

(b) What is the value of $\tau$, the time constant?

(c) What is the voltage after 40 seconds?

(d) What is the voltage after 60 seconds?

(e) How long does it take the voltage to reach 220 volts? Plot the voltage curve and a horizontal line whose intersection gives a visualization of when this happens.

(f) What percentage of the difference in voltage has the capacitor been charged to after $\tau$ seconds?

7. Save this version of the script as YourNameLab3B-2.mw. Mail all your scripts to yourself for future use.

### 3.4 Part 2

**Part 2 Description**

Sometimes rather than plotting points of a function, we are given data points taken from measurements and want to find a function that would produce them. One additional issue is that the data is typically precisely accurate, there is experimental error in making the measurements. So we are satisfied if the function we derive is "reasonably close" to the data points rather than passing exactly through them. This is called the data fitting problem.

Typically rather than searching through all possible functions to find the best fit, we look for good candidates from a particular class of functions. One class are the linear functions: all functions $g(x) = a \cdot x + b$ for some values $a$ and $b$. The data fitting problem becomes that of finding good values of $a$ and $b$.

There are several techniques for doing data fitting. One of the more popular is called least squares data fitting.

The data points $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ can of course be split into two separate lists of values, $x$ written as $[x_1, ..., x_n]$ and $y$ written as $[y_1, ..., y_n]$

**Problem C, Version 1**

(From Anton, Calculus 8th ed., p. 1007)

If a gas is cooled with its volume held constant, then it follows from the ideal gas law in physics that its pressure drops proportionally to the drop in temperature. The temperature, that, in theory, corresponds to a pressure of zero is called absolute zero. Suppose that an experiment produces the following data for pressure $P$ versus temperature $T$ with the volume held constants:

<table>
<thead>
<tr>
<th>$P$ (kilopascals)</th>
<th>134.2</th>
<th>142.5</th>
<th>155.0</th>
<th>159.8</th>
<th>171.1</th>
<th>184.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$ (deg Celsius)</td>
<td>0</td>
<td>20.1</td>
<td>39.8</td>
<td>60.0</td>
<td>79.9</td>
<td>100.3</td>
</tr>
</tbody>
</table>

We want to find values $a$ and $b$ so that the line described by $a \cdot t + b$ does a good job of representing the data. Then we will use the formula we get for $P$ to answer some questions.
1. Create a fresh Maple session through File->New->Document mode.

2. Enter two lists. Call the first list pData and assign it the numbers found in the first row of the above table: pData := [134.2, 142.5, 155.0, 159.8, 171.1, 184.2]. Similarly, create a second list and assign it to tData.

3. Produce a point plot with Maple using the techniques discussed in chapter 5 of the course readings. Make the plot blue.

4. Look up the data fitting facility in maple by starting up Maple help and looking up "least squares". Find the examples given in the documentation page on CurveFitting[LeastSquares] and see one that will help you do data fitting using pData, and tData. Produce a formula for the line.

Notes:

(a) You will have to experiment in order to get things to work. Start by copying and pasting the instructions from the examples and getting them to work as advertised in your own worksheet. Then try substituting pData and tData for the values in the example.

(b) The "with(CurveFitting):" operation needs to be done before you can do any of the other lines in the examples.

(c) You don't have to use "v" as the variable in the curve fitting formula. It makes more sense to use "T".

(d) In the work to come, it helps to give the formula produced by the curve fitting a name, through assignment.

5. Plot the line you got from (d). Make the line blue.

6. Here's a trick to do a quick multi-plot that's not documented in the chapter readings. (a) copy the point plot to the bottom of the worksheet. (b) copy the formula plot. (c) click on the copy of the point plot. (d) right-click and select "paste". It works for a one-of plot but it doesn't lend itself to scripting very easily.

The combined plot should show the line passing close by most of the data points. If it doesn't this is an indication that something is wrong.

7. Once you have gotten a least squares formula, answer the following questions:

(a) Based on the formula, get Maple to estimate the pressure when the temperature is 120 degrees Celsius by evaluating the expression at T=120. (The eval operation is handy here).

(b) Produce an estimate for absolute zero (where pressure is zero) by solving an equation involving this formula. What is your estimate? Look up the actual value of absolute zero on the Internet and compare it with your estimate from this
"virtual experiment". Include your calculated answer in a textual explanation of what you are doing, similar to the way that the target voltage was mentioned in the script for Part 1.

8. Save your worksheet for part 2 as Lab3Part2Solution.mw and mail copies of it to yourself and your lab partner. Be sure to put the names of your team on the worksheet for easy identification.

9. Re do this problem with Tools->Assistants->Curve Fitting. You will want to select "least squares" as the technique for fitting, not splines or interpolation. Be prepared to show your worksheet and the solution with the assistant to the staff for grading. Which way was easier for you to do?

3.5 Attachment: starter script for Part 1

Script for Lab 3, Part 1

CS 121 Computation Lab I

Fall 2010

This script was run by: (fill in here).

This script solves the following problem:

A capacitor connected to a battery charges according to the following formula

\[ V(t) = V_i + (V_{\text{max}} - V_i) \left( 1 - e^{-kt} \right) \]

where \( V_i \) is the starting voltage, \( V_{\text{max}} \) is the final voltage, and \( k \) is the reciprocal of the time constant \( k = \frac{1}{\tau} \). The voltage is measured in volts, the time in seconds.

Problem A, Version 1

For a particular capacitor/battery set up, we find \( V_i = 63, \) \( V_{\text{max}} = 266, \) \( k = 0.09 \).

(a) What is the value of \( \tau \), the time constant?
(b) What is the voltage after 20 seconds?
(c) What is the voltage after 40 seconds?
(d) How long does it take the voltage to reach 256 volts? Plot the voltage curve and a horizontal line whose intersection gives a visualization of when this happens
(e) What percentage of the difference in voltage has the capacitor been charged to after \( \tau \) seconds?

Start of parameters

\[ V_i := 63 \]

(3.1)

\[ V_{\text{max}} := 266; \]

(3.2)

\[ k := 0.09 \]

(3.3)
tl := 20

20

(3.4)

t2 := 40

40

(3.5)

vTarget := 256

256

(3.6)

**End of parameters**

Define a function that describes the voltage. This is not a parameter because it doesn't change between versions of the problem.

\[ V := t \mapsto (Vi + (V_{max} - Vi) \cdot (1 - \exp(-k \cdot t))) \]

\[ t \mapsto Vi + (V_{max} - Vi) \left(1 - e^{-kt}\right) \]  

(3.7)

Test the function at \( t=0 \) -- should be initial voltage.

\[ V(0) \]

63. 

(3.8)

According to the notes, the time constant is the reciprocal of \( k \).

\[ \tau := \frac{1}{k} \]

11.111111111

(3.9)

Since we have defined \( V \), all we have to do is evaluate \( V \) at \( t=20 \) and \( t=40 \).

\[ V(t1) \]

232.4443257

(3.10)

\[ V(t2) \]

260.4532844

(3.11)

To find how long it takes to reach the target voltage, we evaluate \( V \) at the symbol \( t \), and equate it to the target voltage. Solving that equation for \( t \) will give us the answer.

\[ eqn := vTarget = V(t) \]

\[ 256 = 266 - 203 e^{-0.09t} \]  

(3.12)

The target voltage is reached in

\[ solve(eqn, t) \]

= 

[Blank]
seconds.

To do a plot which gives the same kind of answer, plot V and a horizontal line (a "constant function") whose value is the target voltage. The intersection of the two lines indicates where the voltage reaches the target. For the range, we pick ten times the time constant, which should be enough to see the exponential curve go into its "almost flat" phase. After all, when t is equal to ten times the time constant, V(t) should achieve 

\[ 1 - (1 - 0.632)^{10} \]

\[ \approx 0.999544511 \]

of the final voltage Va.

\[ \text{plot}\{ \{V(t), v\text{Target}, t = 0 \ldots 10 \cdot \text{tau}, \text{labels} = \{"time (in seconds)", "voltage"\} \} \]
End of script

3.6 Final actions (end of class)

Upload all of your work to Blackboard, or email copies to yourself and/or your partners.

3.7 Conclusion

In this lab, you have gotten further practice at creating scripts in Maple. We have expanded the repertoire of objects to include lists, which allow us to maintain aggregations of values in an easy-to-access fashion. We have practiced with point plotting, and "multi-plotting" (where multiple curves appear together on a single set of axes). In preparation for programming, we have begun to practice with entry of operations using text rather than mouse clicks and menu items.
4 Lab 4 Cs 121 Computation Lab I Fall 2010 Directions and Problems

4.1 Lab 4 Overview

Overview

The lab explores the problems of finding artillery trajectories that satisfy certain requirements and constraints. These computations use a mathematical model that describe how key values and properties change over time. It uses more sophisticated plotting and animation more rapidly achieve understanding essential to design verification or modification. As you will see, animation can lead to more rapid understanding compared to large tables of numbers or pages of graphs.

Before beginning lab work, you will see the instructor demonstrate how to use function definition, display, and animate which are key operations in the day's lab. You will be asked to write scripts using only textual versions of Maple operations. You will not receive credit for answers that use the clickable interface for operations.

To prepare for this lab beforehand

1. Read Chapters 8 and 9 of the course readings. One way to better understand the material is to learn how to reproduce the examples in the text on your own computer.

2. Study these lab directions and the lecture notes for this lab, posted on the course web site.

3. Take the pre-lab quizlet.

Directions for this lab

1. Find a lab partner. You need not use the same partner as last time. Group members should introduce themselves to each other if they haven't already met.

2. Listen to the instructor's overview of this week's new concepts and Maple features.

3. Get your copy of the verification sheet and check out which parts of the work will be verified.

4. Work on the four problems of the lab (80 minutes). You should create a separate worksheet for each problem, named as described in the directions below. Your work should use only textual versions of Maple operations. You will not receive credit for answers that invoke operations through the clickable interface. You will not receive credit for solutions that are crammed together in a single worksheet.
4.2 Instructor's demonstration of definition of functions, advanced plotting, and animation

The instructor will demonstrate: function definition, function daisy chaining, display, paramplot, and animate.

4.3 Introduction to the "Human Cannonball" simulation

The following problem comes from the book Calculus: Early Transcendentals, 7th edition, by Howard Anton, Irl Bivens, and Stephen Davis, pages 462-465 (module created by John Rickert and Howard Anton): "Blammo the Human Cannonball will be fired from a cannon and hopes to land in a small net at the opposite end of the circus arena. Your job as Blammo's manager is to do the mathematical calculations that will allow Blammo to perform his death-defying act safely. The methods that you will use are from the field of ballistics (the study of projectile motion)."

![Human cannonball (from http://www.phpsolvent.com/wordpress/?p=1263)](image)

In this problem you will compute the equations of motion for Blammo traveling in the plane and use these equations to simulate the motion of Blammo flying towards the net. The equations of motion in the plane are similar to those that were derived and used in first tutorial; however, in this case you must track both the $x$ and $y$ coordinates of the object. Prior to shooting Blammo from the cannon, you will have to specify the angle of elevation of the cannon and the initial speed of Blammo exiting the cannon. Based on these parameters, and the distance between the cannon and the net, you need to determine whether the Blammo hits the net or not. We will initially assume that there is no resistance from the air as Blammo travels and that the only force acting on Blammo is gravity, which only affects the $y$ coordinate.

Consider an elevation angle of $\alpha$ degrees and an initial speed of $V_0$. In the triangle below, the cannon is located at point A, the angle of elevation is the angle CAB and the length of the side AC is equal to the initial speed $V_0$. The initial velocity in the $x$
direction is the length of the side AB and is equal to \( V_{0x} = V_0 \cos(\alpha) \). (Remember, SOHCAHTOA?, \( \cos \alpha = \frac{\text{adjacent}}{\text{hypotenuse}} \)) The initial velocity in the \( y \) direction is the length of the side BC and is equal to \( V_{0y} = V_0 \sin(\alpha) \).

![Diagram of velocity components](image)

The position of the cannonball is given by \((x(t), y(t))\), which provides the coordinates of the cannonball at time \( t \). The equations of motion, as derived in most elementary physics texts, can be found to be \( x(t) = V_{0x} t \) and \( y(t) = y_0 + V_{0y} t - \frac{1}{2} g t^2 \) where

\[
g = 32 \frac{ft}{sec^2} = 9.8 \frac{m}{sec^2},
\]

depending on the units used. \( y_0 \) is the initial position of the object (if we are launching from the ground \( y_0 = 0 \)). We will use these equations for the following problem: Let's work in the English FPS (foot/pound/second) system of units.

If we shoot Blammo off at 100 feet per second at an angle of 45 degrees, we find that

\[
v0 := 100
\]

(4.1)

We can develop a plot, using the paramplot feature of Maple's plots package:

\[
\text{alpha := convert(45 \cdot \text{degrees}, \text{radians})}
\]

(4.2)

\[
v0x := \cos(\text{alpha}) \cdot v0
\]

(4.3)

\[
v0y := \sin(\text{alpha}) \cdot v0
\]

(4.4)

\[
g := 32
\]

(4.5)
We can see that after three seconds Blammo is still in mid-flight, having already reached his apex.

We can solve an equation to find out at what times \( t \) Blammo is on the ground.

\[
\text{solve}(ypos(t) = 0, t)
\]

\[
0, \frac{25}{8} \sqrt{2}
\]

Not surprisingly, one of the times is \( t=0 \) (the start). We can get the time we want by daisy-chaining \text{solve} (which gives a sequence of two roots) and the \text{max} function.

\[
flightTime := \text{max}(\text{solve}(ypos(t) = 0, t))
\]

\[
\frac{25}{8} \sqrt{2}
\]

We daisy-chain the function that calculates \( x \) position with the approximation function \text{evalf}. By default, \text{evalf} computes ten decimal digits accuracy.
Evidently Blammo travels 312.5 feet.

Now, if we plot from $t=0$ to flightTime, we should see the whole plot:

$$\text{plot}([\text{xpos}(t), \text{ypos}(t), t = 0..\text{flightTime}], \text{labels} = \text{["feet", "feet"]})$$

If we want to produce an animation of Blammo flying through the air, we need to create a function that creates a plot with a shape located at $(\text{xpos}(t), \text{ypos}(t))$ for any given time $t$, and then gives it to the animate function for $t=0..\text{flightTime}$.

\[
\text{drawBlammo} := (t) \rightarrow \text{plot}([\text{xpos}(t), \text{ypos}(t)], \text{style} = \text{point}, \text{color} = \text{"red"})
\]

\[
\text{animate}(\text{drawBlammo}, [t], t = 0..\text{flightTime})
\]

Maple has other useful functions to help us:

$maximize$ will find the largest value that a function attains. For example,

$$\text{evalf}(\text{maximize}(\text{ypos}(t)))$$
This means that Blammo reaches a maximum height of 78 1/8 feet. We could figure this out through calculus and/or remembering enough high school analytic geometry about parabolas, but Maple knows how to do these things without further programming on our part.

### 4.4 Problem 1

Suppose we shoot Blammo out of the cannon at an initial velocity of 110 feet per second, at an angle of 50 degrees. Read in the script Lab4StarterScript. Use the computational machinery to determine how high and how far Blammo travels. Play the animation to confirm that it is consistent with the numbers computed, as well as the parameter plot.

This is a "use the script and interpret the results" problem. You have to learn how to use the features in the script, but you don't have to modify any code or write new code.

### 4.5 Problem 2

This problem requires you to modify code. To cleanly separate your work from problem 1, save a copy of the starter script in a different file, `YourNameLab4Problem2.mw`, then begin the Problem 2 worksheet to solve this problem.

We want to have a different shape for Blammo. Consult the on-line documentation for plottools and look up the `disk` and `pieslice` functions. Choose one, and replace the point plot with a red or blue object of your choice. Assume that Blammo is roughly six feet tall.

To develop your code, modify the plotting instructions so that they produce a shape in the correct position rather than a point plot. Then change the function being given to animate to draw the shape rather than the point. Produce an animation for when Blammo is launched at a speed of 50 feet/second, at an angle of 35 degrees.

You may notice that the shape looks more squashed than it ought to be. This is because `animate` is not using the same scaling for the horizontal and vertical axes. To correct this, add the option `scaling=constrained`, as illustrated in the various examples in Chapter 9 of the readings.

Save your work as `YourNameLab4Problem2.mw`, to show to the grader.

This file should contain the solution to problem 2 ONLY. Having several solutions (possibly incorrect, and possibly interfering with each other) all in the same worksheet will just create more problems for you to have to solve.

### 4.6 Problem 3

Suppose we have a tent that's 200 feet long and 50 feet high. We buy a standard explosive charge from a manufacturer that will shoot Blammo out at a velocity of 82 feet per second. Use the computational machinery to determine what angles the gun may be set up to have Blammo safely fly through the air without running into the walls of the tent.

**Directions**

Open up the starter script again and modify it so that it eliminates the animation but adds bounding lines to the parameter plot, so that you can see whether Blammo's trajectory will exceed the boundaries of the tent.

To do this, create a function that uses `display` with some green dotted lines. Plotting this will quickly establish whether Blammo exceeds the boundaries. You don't need to produce an animation.

For example,

```maple
with(plottools):
```
If we display them together we get a plot that looks like this:

```latex
\begin{verbatim}
with(plots):

display([p1,p2])
\end{verbatim}
```

What you want to do is to generate the parameter plot and assign it to `p3`. Then doing `display([p1,p2,p3])` should result in a picture that looks something like this.

```latex
\begin{verbatim}
display([p1,p2,p3])
\end{verbatim}
```
This diagram shows quickly that the flight path goes quickly beyond the boundaries of the tent when we launch Blammo at 100 feet per second at 45 degrees. But you will have different results for 10 feet per second.

Use this as an idea to modify your existing scripts to handle this problem. When you find a range of angles that works, keep the execution from the largest angle that works. Save your work as YourNameLab3Problem3.mw.

4.7 Problem 4

Exchange the script from Problem 2 with one of your partners. You will now use their script to solve another problem.

We would like to add a net into our simulation, that Blammo can land in safely. Change the script from Problem 2 to add a new function drawNet:=(d, w) -> a green line centered at (d,0) with total width w. This means that the line extends from (d-w/2,0) to (d+w/2,0).
Modify your animation function so that every frame displays not only the position of Blammo but also the net. For example, here is a frame of an animation we created that has Blammo (a red disk), flying through the air towards the net.

Through trial and error, find angles and initial velocities that solve the following problems. Try to get Blammo's center point to land as close as you can conveniently arrange to the center of the net.

(a) **Distance to net = 100 feet. Size of net = 10 feet.** Try 70 feet/second and 30 degrees initially. Team members should try various values of v0 and the angle to make Blammo land in the net. Be prepared to play the animation of the successful shot for the grader. Note that you can vary both the velocity and the angle, so it doesn't have to be just a patient variation of just velocity or just the angle to find a solution.

Once you find a solution, see if you can find another solution with an initial velocity 10% faster, and a different angle.

(b) **Distance to net = 200 feet. Size of net = 5 feet.**. Record the velocity and angle that you found that worked.

(c) **Distance to net = 500 meters. Size of net = 3 meters.** Record how many tries it took you to find a solution that worked. In order to solve this problem, you will have to figure out how to handle metric values for the distances. However you find it, at the end of the script be sure to express the initial velocity in meters/second instead of feet per second.

### 4.8 Final actions (end of class)

Email copies to yourself and/or your partners. Be sure to get credit for doing this on the verification sheet before you leave.
4.9 Conclusion

In this lab, you have used a computerized mathematical model of artillery trajectories to solve problems having to do with a human cannonball. One of the advantages of using the model is that it's not necessary to spend as much money on test firings and replacement human cannonballs while you try to find what will work safely. You have extended your repertoire to include more elaborate user-defined functions that can be used to generate animated plots. You have learned how to plot objects besides lines and points. You have gotten more practice at using and modifying scripts.

4.10 Acknowledgements

This exercise was developed with the help of Dr. Jeremy Johnson, Dr. Fred Chapman, and Mr. Ryan Walls.