Lab 3 Directions and Problems

1. The instructor will present a brief overview of this week's new concepts and Maple features, and provide commentary on the lab required activities. (20-25 minutes)
2. The staff will move around the room to grade Lab 2 Part 2. However, while they are doing this you should begin work on Lab 3, since the grading of each group will use only a few minutes.

Notes

3. With your lab partner(s), work on part 1, below. *(40 minutes)*
   • Form a group. Group members should introduce themselves to each other if they haven't already met.
   • All of the partners should log onto a computer.
   • Work read and attempt the problems individually, but stop frequently to compare results and help each other out. Eventually, the graders will be looking for a single collection of answers from the group that everyone in the group can explain. If you are in a room with computer projectors, then please prepare a single computer which can be used to present the group's results to the grader by projection. In most rooms, it's difficult for the grader to stare over your shoulder at results on a screen. Furthermore, the other members of the group also have to stare over a shoulder to see something that they may be asked to talk about.
   • Write down the selected answers on the verification sheet. When you are finished with part 1, have a staff member come over to sign the verification sheet for you. 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 at this point.

Introduction

A model of a Damped Propulsion System

*(From Introduction to Engineering, Modeling and Problem solving, JB. Brockman, 2009 John Wiley and Sons, ch. 12.)*
Problem Statement

An automobile propulsion system can be modeled as a dynamic system with a parameter $u(t)$, a function representing the pedal position of the accelerator, and an output $v(t)$ (representing vehicle velocity) via the formula

$$ \frac{\Delta v(t)}{\Delta t} + p \cdot v(t) = b \cdot u(t) \quad (*) $$

where $p$ is called the damping factor and $b$ is a constant.

In this system, the independent variable in the formula is $t$, representing time. Recall from calculus that the expression $\Delta t$ is typically used to mean "some small value representing the difference of two values of $t$ that are close to each other". $\Delta v$ is shorthand for $v(t+\Delta t) - v(t)$ -- the difference between the values of the velocity function corresponding the two values of $t$.

As you may recall from calculus, the quantity $\lim_{\Delta t \to 0} \frac{\Delta v(t)}{\Delta t} = \frac{d}{dt} v(t) = a(t)$, the instantaneous acceleration at time $t$. Thus the formula $(*)$ is talking about how acceleration and velocity are related through damping factors and the position of the accelerator pedal.

Let's list the symbols and their meanings again:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>time (in this system, measured in seconds)</td>
</tr>
<tr>
<td>$v(t)$</td>
<td>velocity of the automobile at time $t$. Measured in feet/second in this system.</td>
</tr>
<tr>
<td>$v0$</td>
<td>initial velocity at $t=0$. Usually specified as an input to the system if this information is needed for a calculation.</td>
</tr>
<tr>
<td>$a(t)$</td>
<td>acceleration of the automobile at time $t$. Measured in $\frac{ft}{sec^2}$.</td>
</tr>
<tr>
<td>$x(t)$</td>
<td>position of the automobile at time $t$. $x(0)=0$ by assumption. Measured in feet (from the origin).</td>
</tr>
<tr>
<td>$u(t)$</td>
<td>The position of the accelerator pedal at time $t$. Measured in &quot;pedal position units&quot; which are part of the automobile design, not measured in standard metric or FPS units. In the original version of the simulation, $u(t)$ is the constant function, always returning 1. This corresponds to an accelerator that is always in the &quot;unit position&quot;. Later on we will use other, varying for $u(t)$.</td>
</tr>
<tr>
<td>$p$</td>
<td>The &quot;damping factor&quot; in the formula $(<em>)$. As used in $(</em>)$, it specifies that there are forces proportional to velocity that act to slow the car down. This might be due to air resistance, or friction due to moving mechanical parts. For the sake of the lab exercises, we will just specify what value you should use, e.g.</td>
</tr>
</tbody>
</table>
$p = 3.8$. Assuming that this formula is a realistic model of a car, we could determine the real-life value of $p$ through experimental measurement in a lab.

| $b$ | A constant. It represents how the pedal position value is translated into the units used by the other terms in (*). For the sake of the lab exercises, we will just specify what value you should use, e.g. $b = 100$. |

The Maple script we will use and study in this lab calculates values for $a$, $v$, and $x$ at various times from their values at previous times using a loop. It is similar in style to the chemical reaction and predator/prey calculations that we did in labs and quizzes in CS 122. However, in this lab we will explore how the script can be used and modified to answer questions about the automotive system.

In the predator-prey problem, the time interval between each value computed was one year. In this situation, the time interval will be a parameter, $dt$ that can be set initially. Typically $dt$ is set to a small number such as .01 or .001.

The Maple code contained in Lab3Drive1.mw. Open the file and run the simulation now. Note that it chooses $dt = 0.01$. The script prints out a plot of velocity versus time over 2.0 seconds of automobile operation when the vehicle begins at rest ($v[1] = 0$) and the input is a "unit step" -- i.e., the accelerator is depressed 1.0 unit starting at time zero and is left in that position during the entire simulation.

The script uses several variables to store information.

<table>
<thead>
<tr>
<th>Lab3Drive1 variables and their purpose</th>
<th>Values of the constants used in the automobile formula (*) (see previous table)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p, b, v0$</td>
<td>A table whose i'th entry contains the value of t corresponding to the (i-1)st time step. Thus timetbl[1] is 0, timetbl[2] is $dt$, timetbl[3] is $2*dt$, etc.</td>
</tr>
<tr>
<td>timetbl</td>
<td>Tables corresponding with information about the approximate position, velocity, and acceleration of the car whose i'th entry contains the value of $t$ corresponding to the (i-1)st time step. For example, $v[10]$ will contain the velocity information for time $t = 9<em>dt$, $a[52]$ will contain the acceleration information for time $t = 51</em>dt$, etc.</td>
</tr>
<tr>
<td>$x, v, a$</td>
<td></td>
</tr>
</tbody>
</table>
A table corresponding to information about the value of $u(t)$. As with the other tables, the $i$'th entry contains the value of $u$ corresponding to the $(i-1)$st time step. In the initial version of the simulation, since $u(t)$ is constant, all the entries of the table have the same value.

There are $size+1$ elements in each table after the script has finished executing. This is because the loop index $i$ goes from 1 to $size$ but there are assignments such as $\text{timetbl}[i+1] := .. \quad \text{v}[i+1] := ....$ etc.

### Part 1

If you haven't run the simulation yet, go back to the introduction and read the directions for how to run it. Then run it and see its behavior. You'll need this understanding in order to begin work on this part.

#### Problem 1.1

Add to the script more plots to get complete information about how things are changing over time.

a) acceleration versus time  
b) position $x$ versus time  
c) $u$ versus time

Note that the graph for velocity of the original simulation includes a caption that describes the value of the constants $p$ and $b$. As we will be varying $p$ and $b$ later on, you should put similar captions into your additional plots.

#### Problem 1.2

To answer these questions, you to look over the code and combine it with a little math to understand it a bit better. You shouldn't need to run the script to answer the questions except to experiment to provide some evidence for your hunches.

(a) If we want the simulation to go on for 4 seconds with $dt=.001$, what should $size$ be? What about three seconds with $dt=.1$? What about $K$ seconds with $dt=\delta$? Be prepared to explain how you got the answer.
(b) What value would timetbl[1000] have in it when size=2001 and dt=.05? What table entry would have the estimated value velocity at time t=3.4 when dt=.001? What table entry would have the estimated position of the car at time t=T, when dt=δ?

**Problem 1.3**

Create a version of the script that outputs a plot of velocity in miles per hour as well as feet per second. In order to do this, you should use another table vmph. vmph[i] can be the conversion of v[i] into miles per hour, using the convert function as described in Section 19.3..

**Problem 1.4**

The next thing is to understand how to invoke the simulation so that we can explore additional situations.

Determine how to invoke the simulation so that you can vary how many time steps the simulation runs, but using the same values of \( dt, p, b, u0, \) and \( v0 \) as before.

Create copies of your work from problem 1.1 plus the insight you have into the simulation so that you can answer the following questions:

(a) According to the simulation, approximately how far does the car travel in four seconds? In order to answer this question you will need to modify the \( \text{size} \) parameter so that the simulation covers more time steps.
(b) Approximately long does it take the car to travel a mile? How fast is it going at that point, in feet per second? Get the units calculator to express the terminal velocity in miles per hour instead of feet per second.
(c) Approximately how long does it take the car to travel 4 kilometers? How fast is it going at that point, in kilometers per hour?

Hint: you can get rough approximations just by reading the graphs. Clicking on the graph and then positioning the mouse over the curve will produce a reading of the (x,y) coordinate where the mouse is positioned, as first described in section 2.5.3. You can do conversions to metric using the units calculator built into Maple, as explained in section XX.

**Part 2**

**Problem 2.1**
Turn the script into a Maple procedure called \( drive2 := \text{proc}(dt, p, b, size, u0, v0) \ldots \). The function should have as input parameters as indicated. The procedure should print out its plots as a side effect and return NULL as its final result.

You can use the same process described in Section 16.2 to convert the script into a procedure: You will need to identify which variables are local, and where the return NULL; statement should be added.

(a) Test the your function with \( drive2(0.01, 3.8, 100, 201, 1.0, 0.0) \); It should give the same output as the original script. Then test your function with a four second run and verify that it agrees with your answer in Problem 1.2.

Once you have found that your procedure passes these tests, answer the following questions:

(b) What is the terminal velocity reached by the car if it starts with an initial velocity of 10 feet per second, instead of 0? What about 15? What about 30? Compare these results you got in the original version of the problem. Be prepared to describe in your own words the relationship between initial velocity, final velocity, and the way things change between the two.

(c) Vary the damping factor \( p \) and the constant \( b \) and observe the effects on the final velocity and on how quickly the velocity changes. Contrast \((p=3.8, b=100)\) with \((p=38, b=1000)\). What stays the same, and what changes? Make a prediction of what \((p=.38, b=10)\) would be like.

(d) Find values of \( p \) and \( b \) such that the terminal velocity is (roughly) 65mph, and the time to achieve it is somewhere between 5 and 10 seconds. Be prepared to describe what you tried in order to find these values.

**Problem 2.2**

Now create a different function called \( drive3 \). Instead of taking a parameter \( u0 \) that specifies a constant operation of the accelerator, change things so that \( drive2 \) takes a parameter \( ufunc \). \( ufunc \) should be a function (not an expression) that describes the accelerator position as a function of time.

In order to create \( drive3 \), you can do the following:

1. Create a new document. Copy the code region containing \( drive2 \) into the document. Rename the procedure \( drive3 \). Rename the \( u0 \) parameter to \( ufunc \).

2. Look at the lines where \( u \) is calculated, and used. We still want to use the \( u \) table, but now
we will fill it up by invoking \texttt{ufunc} at the appropriate value of time.

Now, answer the following questions:

(a) For a test assume that someone has defined

\[
\textit{uVersion1} := t \rightarrow 1 \\
\textit{t} \rightarrow 1
\]  

(1.4.2.1)

Then \texttt{drive3(.01, 3.8, 100, 201, uVersion1, 0.0)};

should produce the same results as the original script execution from Problem 1.1.

(b) You are considering an alternative accelerator use because it saves gasoline. However, you are concerned that while energy-saving, it customers will complain that the car is not very zippy.

The accelerator you are considering works as follows:

\[
\textit{uVersion2} := t \rightarrow \begin{cases} 1.0 & 0 \leq t \leq 1 \\ 0.5 & 1 < t \\ \end{cases} \\
t \rightarrow \text{piecewise}(0 \leq t \text{ and } t \leq 1, 1.0, 1 < t, 0.5)
\]  

(1.4.2.2)

That is, it's in position 1 for the first second, but position 0.5 after that.

Run the simulation using \textit{uVersion2} and satisfy yourself that it is producing correctly calculated results. Then answer the following question:

Suppose we have two cars. Car A uses \textit{uVersion1}, Car B uses \textit{uVersion2}. In all other ways, they are the same. We start both cars moving at the same time. Approximately how long does it take Car B to travel the same distance that Car A reached in two seconds? Will Car B ever be ahead of Car A at some point in time? How does its terminal velocity compare?

\section*{Problem 2.3}

The business moguls of the automotive company we work for have told us that it will be too expensive to build a car that always delivers a full unit of acceleration. However, they say that the company can make money off of any car whose accelerator function is of the form
\[ t \rightarrow \begin{cases} 
A1 & 0 \leq t \leq 1 \\
A2 & 1 < t \leq 2 
\end{cases} \text{ for any combination of A1 and A2 such that A1+A2=1.5.} \\
0.9 & 2 < t
\]

In order to impress the car magazine people, they want to build a car that has the best 0-55 mph time subject to this constraint.

You've already found values of \( p \) and \( b \) that allow the car to car travel at a peak speed of 65mph in five or six seconds with a pedal position of 1. Using those values of \( p \) and \( b \), figure out which of the following reaches 55 mph most quickly.

\[ t \rightarrow \text{piecewise}(0 \leq t \text{ and } t \leq 1, 0.9, 1 < t \text{ and } t \leq 2, 0.6, 2 < t, 0.9) \] (1.4.3.1)

\[ C1 := t \rightarrow \begin{cases} 
.9 & 0 \leq t \leq 1 \\
.6 & 1 < t \leq 2 \\
0.9 & 2 < t
\end{cases} \]

\[ t \rightarrow \text{piecewise}(0 \leq t \text{ and } t \leq 1, 0.8, 1 < t \text{ and } t \leq 2, 0.7, 2 < t, 0.9) \] (1.4.3.2)

\[ C2 := t \rightarrow \begin{cases} 
.8 & 0 \leq t \leq 1 \\
.7 & 1 < t \leq 2 \\
0.9 & 2 < t
\end{cases} \]

\[ t \rightarrow \text{piecewise}(0 \leq t \text{ and } t \leq 1, 0.75, 1 < t \text{ and } t \leq 2, 0.75, 2 < t, 0.9) \] (1.4.3.3)

\[ C3 := t \rightarrow \begin{cases} 
.75 & 0 \leq t \leq 1 \\
.75 & 1 < t \leq 2 \\
0.9 & 2 < t
\end{cases} \]

\[ t \rightarrow \text{piecewise}(0 \leq t \text{ and } t \leq 1, 0.7, 1 < t \text{ and } t \leq 2, 0.8, 2 < t, 0.9) \] (1.4.3.4)

\[ C4 := t \rightarrow \begin{cases} 
.7 & 0 \leq t \leq 1 \\
.8 & 1 < t \leq 2 \\
0.9 & 2 < t
\end{cases} \]

\[ t \rightarrow \text{piecewise}(0 \leq t \text{ and } t \leq 1, 0.6, 1 < t \text{ and } t \leq 2, 0.9, 2 < t, 0.9) \] (1.4.3.5)

\[ C5 := t \rightarrow \begin{cases} 
.6 & 0 \leq t \leq 1 \\
.9 & 1 < t \leq 2 \\
0.9 & 2 < t
\end{cases} \]
Problem EC.1

Let's explore the effect of making dt larger and smaller. Modify drive3 so that instead of display plots and returning NULL, it doesn't plot but returns a list of two lists. The first item of the result returned is the list of time values. The second item in the result is the list of x values. Call this procedure drive4.

Now invoke drive4 with the original value of $p=38$, $b=100$, $v_0=0$, $ufunc=uVersion2$. Run drive4 so that you see the results of of the x versus time for two seconds. Vary dt and size so that you see the results for $dt = .1, .09, .08, ..., .01, .009, .008, ..., .001]$. Use the view option to plot so that the vertical axis for 0-10 is displayed. Alter the title option to plot so that it displays the value of dt being used.

For example, we might see for $dt = .09$ the following graph:
but when \( dt = .009 \) we might see:

### x position with \( dt = 9.000000e-03 \)

Once you have gotten these pictures to appear successfully, create an animation that plays all of them in sequence. Then answer the following questions:

What values of \( dt \) appear to work well? Given the smaller values of \( dt \) take more time for the computer to process because there are more time steps, value of \( dt \) would you choose for this problem? Be prepared to discuss and justify your conclusions on this.

Now try to do this using \( uVersion1 \). Do the same conclusions hold?

**Afterword: Choice of \( dt \) and accuracy of result**

What you've found is that the choice of \( dt \) does matter. The discrepancies you see in the results are not due to an error in the programming, in that the computer is carrying out the calculations as you have specified. Even if we used exact rather than the limited-precision
numbers in the calculation, we would see similar results. We just have an approximation technique that does terribly when dt is inappropriate.

Experts in computation, called numerical analysts have during the past few centuries, invented many kinds of approximation techniques. Some of the better software packages will try to find appropriate values of dt automatically, so the user is freed from the need to find an appropriate value of dt themselves. You should learn more about these methods in your differential equations course.

➤ Wrap-Up

4. Save your worksheets with a name to indicate that they are no longer the "starter" versions: MyLab3Drive3.mw, etc. and upload them to a safe place. Hand in the signed verification sheet. You don't get credit for the lab without doing this. The lab staff will give you permission to leave the lab after you do this. You can get partial credit for the lab if a portion of your work is verified.

➤ Notes