Chapter 5

Lab 1: Introductory Example

We now begin working on the first few interactive lab programs. They will allow you to experiment with optimal control problems and see the solutions. Most of the labs are based on current applied mathematical research, dealing with an array of biological problems. The first is the problem from the preceding chapter, and the code used is exactly what we developed there. Before preceding, however, we need to clarify a few details about the programs and MATLAB.

First, while MATLAB is needed to run the provided programs, it is certainly not needed to solve optimal control problems in general. Any mathematical programming language, such as FORTRAN or C++, is capable of the calculations needed. MATLAB was chosen for this text because, in the opinion of the authors, it is easily accessible and has superior graphing tools.

On that note, however, the programs used in this workbook are designed so that no knowledge of MATLAB is required. For each problem, there is a user-friendly interface that will guide you through. Each lab consists of two different MATLAB programs, *lab_.m* and *code_.m.* For example, there are two programs associated with Lab 1, *lab1.m* and *code1.m.* Here, **.m* is the extension given to all files intended for use in MATLAB. The file *code1.m* is the Runge-Kutta based, forward-backward sweep solver we built in the previous chapter. It takes as input the values of the various parameters in the problem and outputs the solution to the optimality system. The file *lab1.m* is the user-friendly interface. It will ask you to enter the values of the parameters one by one, compile *code1.m* with these values, and plot the resulting solutions. All the files must be in the directory that MATLAB treats as the home directory. This is usually the *Work* directory.

If you have experience with MATLAB, you may wish to not use the interface and instead use only the actual codes. They operate as standard MATLAB function files, with the parameters entered as input. This will allow you a little more freedom than the interface. However, the interface, especially when going through the labs, is very convenient and will most likely save time. If you do choose to use only the *code_* files, you will need to run the interface a few times before starting the labs in order to see exactly what they do, so that you can emulate them on your own.

If you are not a seasoned MATLAB veteran, do not worry. This book is written with you in mind. However, we do need to cover just a few basic things about MATLAB. When you open MATLAB, there will be several different windows or portals. There will be one, most likely the largest and most likely on the right side, called the Command Window. Everything we do in this book will take place here. In the Command Window, there will be a prompt. This is where you will type your commands. For example, to open the interface for Lab 1, simply type *lab1* at the prompt and press enter.

One of the more important commands to know is the stop command. Any time you wish MATLAB to stop what it is doing, simply hit *Ctrl-c*. This will kill the current application and return the prompt. It will also report to you exactly what it was doing when you gave the stop order, but this will rarely be of interest to you. The command *Ctrl-c* may be useful when you enter certain parameters. Ill-conditioned problems or problems with invalid parameter values will not necessarily converge. This will not stop MATLAB from trying, however. It will continue to sweep forward and backward until it is stopped. All the data provided in the labs is taken from the research, so convergence always occurs. However, when you supply your own data, which you are highly encouraged to do, you have no such guarantee. Unless otherwise specified in the lab, convergence should take no longer than 30 seconds. If it has failed to do so by then, stop the application and try different numbers.

The interface should be self-explanatory. Once opened, it will ask you to enter a value for the first parameter. Type a number and press enter. If you fail to type a number, or enter a number which is not of the right type, you will receive an error message and be asked for the parameter again. If you accidentally enter the wrong value and press enter, simply hit *Crtl-c* and begin the *lab_* program again. Once all the parameters are entered, it will display "One moment please ..." as it compiles the solutions using the *code_* program. After the solutions have been found, it will ask if you would like to vary any parameters. If you respond negatively, it will automatically plot the optimal solutions in labeled graphs. The graphs may appear too small to view. However, if you expand the window, the graphs will enlarge appropriately. If you reply positively about varying the parameters, it will ask which parameter to vary. You will then be asked to enter a second value for this parameter. It will compile a second set of solutions. This set represents solutions for the problem with the same parameters as before, except the chosen parameter will be changed to the new value. Then, both solutions will be plotted together, with the original solutions plotted in blue, and the second set plotted in green. This will allow you to evaluate how each variable affects the optimal system. When you are done studying the graph, simply go back to the prompt and retype *lab_* to start the interface over.

Finally, you are encouraged to compute the optimality system for each problem by hand. Then, open the *code_.m* files to see that system translated to MATLAB code. All the programs work more or less the same way and are written in a uniform manner that should make them easy to read. To open an m-file, select the OPEN option under FILE in the upper left-hand corner. There may also be a standard open folder icon at the top of your screen, depending on how your MATLAB interface is organized. This will open a window showing all the m-files in your *Work* folder.

If you have any questions about MATLAB, you can refer to the MATLAB manual or to any of the numerous MATLAB guides available. However, the MATLAB Help menu within the program is one of the best resources of information.

This first lab will utilize the code developed in Chapter 4 in order to solve the following optimal control problem.

$$\max_u \int_0^1 Ax(t) - Bu^2(t)\, dt$$

$$\begin{aligned} \text{subject to} \quad & x'(t) = -\frac{1}{2}x^2(t) + Cu(t),\; x(0) = x_0 > -2, \\ & A \geq 0, B > 0. \end{aligned}$$

To begin the program, open MATLAB. At the prompt, type *lab1* and press enter. To become acquainted with the program, perform a few test runs. Enter values for the constants A, B, C, and x_0. At first, do not vary any parameters. The graphs of the resulting optimal solutions, i.e., the adjoint and the optimal control and state, will automatically appear. Run the program again, enter different values, and vary one of the parameters. Once you feel comfortable with the structure of the program, begin working through the lab exercises below.

This lab will focus on using the program to characterize the optimal control and resulting state and to ascertain how each parameter affects the solution. First, let us consider the goal of the problem. On one hand, we want to use the control u to maximize the integral of x. On the other hand, we also want to maximize the negative squared value of u. This, of course, is equivalent to minimizing the squared value of u. Thus, we must find the right balance of increasing x and keeping u as small as possible. Enter the values

$$\boxed{A = 1 \quad B = 1 \quad C = 4 \quad x_0 = 1} \tag{5.1}$$

and do not vary any parameters, then look at the solutions. Your output should look something like Figure 5.1. We see u begins strongly, pushing x up but steadily decreasing to 0. This makes logical sense when we consider the differential equation of x. Undisturbed by u, the state x will decrease monotonically. So, we want to push x up early in the time period, so that the natural decay will be less significant. As we only care about minimizing the integral of u, and the distribution is irrelevant, the control should be highest early on. We see this is exactly what the optimal control is. Also, note that x begins to decrease at the end of the interval, as the control approaches zero.

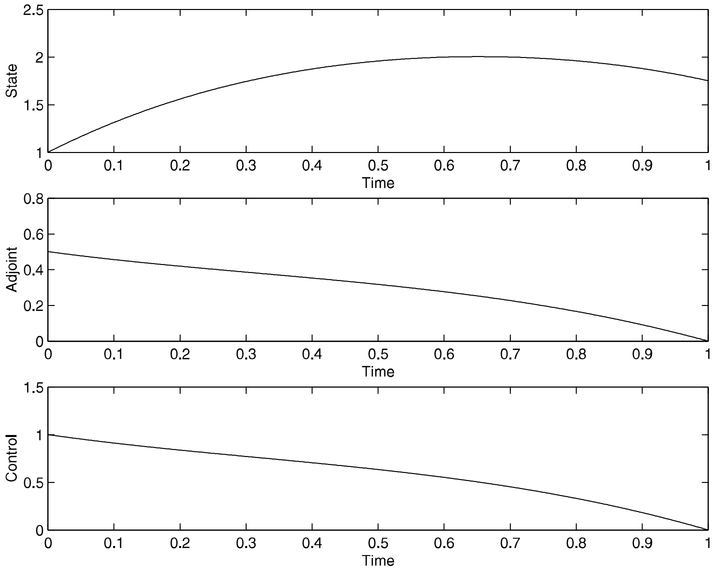

FIGURE 5.1: The optimal state, adjoint, and control for the values (5.1).

Reenter the values in (5.1) and then vary the initial condition with $x_0 = 2$. As the second state begins higher, less control is needed to achieve a similar effect. Notice that the second control begins lower than the first, but they quickly approach each other and are almost identical by $t = 0.6$. This causes the two states to move towards each other as well, although they never actually meet. Now use $x_0 = -1$. This time, x begins below zero, so a greater control is needed to push the state up more quickly. Notice, however, we see the same effect as before, where the two controls eventually merge, although, much later than in the previous simulation. We mention here why the requirement $x_0 > -2$ is imposed. If you were to solve the state equation without u (i.e., $C = 0$), you would find $x_0 > -2$ is required, or division by 0 will occur and the state will blow-up in finite time. However, we know u will be used to increase x, so this condition is sufficient to give a finite state solution with the control.

Use the (5.1) values, varying C with $C = 1$. We have decreased the effect u has on the growth of the state. The optimal control in the second system is less than in the first. It is worth using a greater control in the first system, as it is more effective. Also, the second state, unlike the others we have seen, is decreasing over the whole interval. What little control is used does not increase the state, but only neutralizes some of the natural decay. It would now take far too much control to increase the state. Enter the same parameter values, this time varying with $C = 8$. The results are as you might expect. The second optimal control, now more effective, is greater than the first. The second state increases far more than the first, but still decreases as its control approaches zero. Finally, note that when C is varied, we do not have the two controls merging together.

Enter (5.1) and vary with $C = -4$. The control now has the opposite effect on the growth of the state. We see the control for the second state is merely the first control reflected across the x-axis, while the state and adjoint are the same. Try $C = 0$. Here, the control has no effect on x, so the optimal control is $u \equiv 0$, regardless of A, B, or x_0.

Reenter (5.1). Choose to vary A. Specifically, try $A = 4$ as your second value. In the second system, $A = 4B$, so maximizing $x(t)$ is four times as important as minimizing u^2. We see this playing out in the solutions. A greater u is used so that x can be increased appropriately. Conversely, enter (5.1) varying with $B = 4$. In this case, minimizing $u(t)^2$ is more important. We see on the graph, $u(t)$ is pulled closer to zero, even though this causes $x(t)$ to increase much less at the beginning. The constants A and B are called *weight parameters*, as they determine the importance or weight of variables in the objective functional.

If you were to compare the graphs of the optimal solutions for

$$\boxed{A = 1 \quad B = 2 \quad C = 4 \quad x_0 = 1} \tag{5.2}$$

to the solutions for

$$\boxed{A = 2 \quad B = 4 \quad C = 4 \quad x_0 = 1} \tag{5.3}$$

you would notice they were exactly the same. This is because the system is only influenced by the ratio of the constants A and B, not the actual values. We know $B \neq 0$, so we could divide it out of the integral. This would make our objective function

$$B \int_0^1 \frac{A}{B} x(t) - u(t)^2 \, dt.$$

Of course, the constant B in front of the integral is irrelevant, so we ignore it. Thus, the only constant of significance in the integrand is $\frac{A}{B}$. In all future labs, one term of the integrand will have no weight parameter, as it has been divided out.

Before finishing, we look at a few special cases. Try $A = 0$. This will also cause the trivial solution $u^* \equiv 0$ regardless of B, C, and x_0. If we no longer care about maximizing x, then we clearly should simply pick $u \equiv 0$ and ignore x. We cannot choose $B = 0$, because we divide by B in the characterization of the control. However, a similar situation occurs as we let $B \to 0$. For instance, try $A = 1$ and $B = 0.01$. Then, compare the graphs to $A = 1$ and $B = 0.00001$. A very large u (or large negative u, if $C < 0$) is used to push x

up as quickly as possibly, because almost no importance is placed on keeping u^2 small.

Exercise 5.1 Reconsider the problem with $B = 0$

$$\max_u \int_0^1 Ax(t)\,dt$$

$$\begin{aligned} \text{subject to} \quad & x'(t) = -\frac{1}{2}x^2(t) + Cu(t),\ x(0) = x_0 > -2, \\ & A \geq 0. \end{aligned}$$

Show (analytically) that no optimal control can exist when $A > 0$.

Chapter 6

Lab 2: Mold and Fungicide

For the second lab, we will explore an optimal control problem with biological applications. Let $x(t)$ be a population concentration at time t, and suppose we wish to reduce the population over a fixed time period. We will assume x has a growth rate r and carrying capacity M. The application of a substance is known to decrease the rate of change of x, by decreasing the rate in proportion to the amount of u and x. Let $u(t)$ be the amount of this substance added at time t. For example, the population could be an infestation of an insect, or a harmful microbe in the body. Here we view $x(t)$ as the concentration of a mold and $u(t)$ a fungicide known to kill it. The differential equation representing the mold is given by

$$x'(t) = r(M - x(t)) - u(t)x(t), \, x(0) = x_0,$$

where $x_0 > 0$ is the given initial population size. Note the term $u(t)x(t)$ pulls down the rate of growth of the mold. The effects of both the mold and fungicide are negative for individuals around them, so we wish to minimize both. Further, while a small amount of either is acceptable, we wish to penalize for amounts too large, so quadratic terms for both will be analyzed. Hence, our problem is as follows

$$\min_u \int_0^T Ax(t)^2 + u(t)^2 \, dt$$

$$\text{subject to} \quad x'(t) = r(M - x(t)) - u(t)x(t), \, x(0) = x_0.$$

The coefficient A is the weight parameter, balancing the relative importance of the two terms in the objective functional. As we saw in the last lab, one weight term can be divided out, so only the A parameter is needed here. The other parameter in front of the u^2 is taken to be 1. To begin, type *lab2* and press enter. Enter the values

$$\boxed{r = 0.3 \quad M = 10 \quad A = 1 \quad x_0 = 1 \quad T = 5}\,. \tag{6.1}$$

Do not vary any parameters for now. The control initially increases, then levels off to become constant. The state is also constant here; we say the control and state are in *equilibrium*, meaning both stay at constant values. The control eventually begins decreasing again, going all the way to 0. The state never decreases, with heavy growth at the beginning and end of the interval and constant in the middle. In application, though, we wanted to eliminate the state, or at least decrease it. Note, we entered the value $A = 1$, meaning lowering the level of mold is as important as keeping the levels of fungicide down. This generally would not be the case, however. We are much more interested in removing the mold. Therefore, we should use a higher weight parameter.

Enter the values

$$\boxed{r = 0.3 \quad M = 10 \quad A = 10 \quad x_0 = 1 \quad T = 5}\,. \tag{6.2}$$

Here, the level of fungicide used is much higher. Notice that the state and control still experience the long period of equilibrium. The control begins at its greatest point, decreasing slightly before becoming constant, then decreasing to 0. As desired, the state decreases from its initial amount to about 0.95 and becomes constant. However, at the end of the interval, when the fungicide use decreases, the level of mold rapidly increases. Seemingly, the best course of action would be to begin another 5-day regimen of a second fungicide on about day 4. For comparison, see Figure 6.1, which shows the optimal state with these values, versus a mold population where no fungicide is used ($u \equiv 0$).

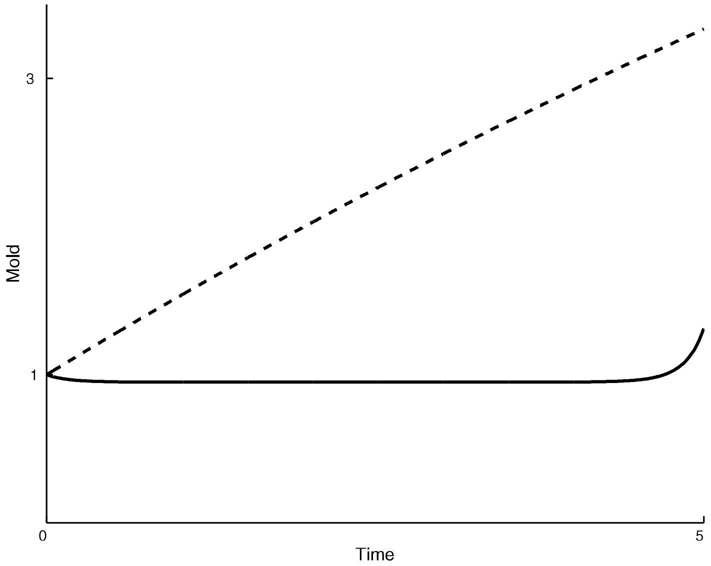

FIGURE 6.1: The optimal mold population for (6.2), in solid, increases at the end of the interval, but is held much lower overall than if no fungicide was used (dashed).

Now try varying two larger values of A. Enter the values (6.2), varying with $A = 15$. The two systems have similar dynamics. A stronger regimen of fungicide is used in the second system; hence, the state is driven lower before becoming constant. Both controls decrease to 0, so that both states experience rapid growth at the end of the period. The second state does not grow as large as the first, though, because it begins at a lower equilibrium point and is still receiving slightly more fungicide.

Enter (6.2), varying with $r = 0.1$. The mold in the second system has slow natural growth. Much less fungicide is used, but the mold in the second system still decreases more than the first. Also, at the end of the interval when both controls decrease, the increase of the first state is a great deal sharper or more rapid than the second. We also mention that the second state is the first we have seen where the amount of mold is less at the end of the interval than at the beginning. Now vary with $r = 0.5$. Here, the state is everywhere increasing, even though a stronger control is used. We are not able to find an acceptable schedule of fungicide strong enough to overcome the natural growth rate of the mold. It is worth noting the two controls begin at the same value.

Examine the carrying capacity M. Enter (6.2), varying with $M = 12$. Clearly, the higher carrying capacity will cause the mold in the second system to naturally increase more rapidly. Much like we saw with the growth rate, the best strategy is to balance this effect with the control and state. Namely, a stronger control is used and a less desirable state is achieved. Note, the controls begin at the same point here as well.

So far, we have only looked at systems where the mold concentration begins at a fraction of carrying capacity. Look at a simulation where x_0 is close to M. Then, try $x_0 = M$. For instance, run

$$\boxed{r = 0.6 \quad M = 5 \quad A = 10 \quad x_0 = 5 \quad T = 5}\,. \tag{6.3}$$

You will notice a change in the overall behavior of the control. Here, the control begins with an extremely strong dose of fungicide, with $u(0) \approx 5 \times u(1)$. It then quickly returns to the levels we have seen, becoming constant. However, the state, near carrying capacity, will experience little initial natural growth. So, the initial blitz of fungicide is devastating to the population. This allows the large decrease in fungicide use before the equilibrium period, despite the high growth rate.

Now try varying the initial concentration. Enter (6.2), varying with $x_0 = 2$. In this simulation, a much stronger control is used initially in the second system, pushing the second state closer to the first. At approximately $t = 0.75$, the states become identical, as do the controls. Now vary with $x_0 = 3$. Now with $x_0 = 0.5$. We see the same behavior occurs, always in about the same amount of time. So, initial concentration affects only the initial dose of the fungicide regimen. Afterwards, a uniform schedule is used, based on the other parameters. We will see that this phenomena occurs in several of the later labs as well.

Finally, vary the length of the time interval. Enter (6.2), varying with $T = 2.5$. The initial dynamics are identical. The systems differ only after the second system exits the equilibrium state. Notice, the decrease of the control and increase of the state, which occur at the end of the interval, is the same in both systems, only occurring at different times. In fact, if you continue to experiment, you will see altering T only changes the length of the equilibrium period. On that note, if you make T small enough, say $T = 0.5$, you will eliminate the equilibrium period entirely, and the dynamics will be noticeably altered.

Before finishing, we bring up an example of what can go wrong with numerical solutions. MATLAB, due to the amount of information it is able to store, actually has a "largest number." If we enter values which cause x to grow too fast, the state can actually reach this limit. When this happens, MATLAB will simply stop calculating. Enter the values

$$\boxed{r = 1 \quad M = 20 \quad A = 10 \quad x_0 = 1 \quad T = 5}\ . \tag{6.4}$$

You see the graph goes straight up, then stops at about $t = 1.75$. This behavior also occurs in the next lab.

Chapter 7

Lab 3: Bacteria

Suppose a certain bacteria is grown in a lab, perhaps for medical use. Left alone, the bacteria population will grow exponentially, with growth rate r. A chemical nutrient is known to speed the reproduction process of the bacteria when added. However, use of the chemical by the bacteria creates a second chemical byproduct, which hinders growth. It is also known that the level of hinderance is related to the size of the bacteria population. Namely, the larger the bacteria population is, the smaller the effect this byproduct will have. It is believed this relation is roughly exponential. Therefore, if $x(t)$ is the bacteria concentration at time t, then the growth is given by

$$x'(t) = rx(t) + Au(t)x(t) - Bu(t)^2 e^{-x(t)},$$

where $u(t)$ is the amount of the chemical being added at time t, A is the relative strength of the chemical nutrient increasing growth, and B is the strength of the byproduct. Let $x_0 > 0$ be the given initial concentration. We will consider growth and supplementation over the normalized time interval $[0, 1]$. We wish to maximize x at the end of this interval while simultaneously minimizing the amount of chemical agent used. Thus, our problem can be stated

$$\max_u \; Cx(1) - \int_0^1 u(t)^2 \, dt$$

$$\begin{aligned} \text{subject to} \quad & x'(t) = rx(t) + Au(t)x(t) - Bu(t)^2 e^{-x(t)},\ x(0) = x_0, \\ & A, B, C \geq 0. \end{aligned}$$

Before beginning, we make two short notes. First, it is easily shown from the adjoint and transversality conditions that $\lambda(t) > 0$ for all t. Thus, we can get the characterization of the control as usual (see Exercise 7.1). Second, unlike the previous labs, there is a payoff term. Here, $\phi(x) = Cx$ and $\phi' = C$. So, the adjoint is not zero at the end of the interval, but $\lambda(1) = C$. In previous MATLAB codes, we had set the variable λ equal to a vector of zeros, to declare its size. We were also inserting the transversality condition: the adjoint is zero at the final time. Here, we must set $\lambda(1)$ equal to the constant C. You can see in the file *code3.m* this is precisely what is done.

To begin the program, type *lab3* and press enter. Enter the values

$$\boxed{r = 1 \quad A = 1 \quad B = 12 \quad C = 1 \quad x_0 = 1}\,. \tag{7.1}$$

For now, do not vary any parameters. We see the chemical injection is concentrated at the end of the time interval. This is due to the decreasing effect of the byproduct. As x becomes larger, the e^{-x} term decreases, and the byproduct has less of a hindering effect. Consequently, the level of chemical added starts fairly low and steadily increases, with noticeably higher rates of increase around $t = 0.6$ and $t = 0.8$. As such, the bacteria growth is approximately exponential early in the time interval, but begins to increase more and more rapidly.

Enter (7.1), varying with $x_0 = 0.9$. The two solutions begin very close to each other. However, the chemical use in the first system increases slightly faster than the second, leading to a significant difference by the end of the interval. As the bacteria concentration in the first system becomes larger than the second, the effect of the byproduct becomes less significant, and more chemical can be used. Now try varying with $x_0 = 1.1$. The differences in this simulation are more pronounced than before. As x_0 is increased, more chemical can be used earlier. Now try $x_0 = 1.1$ vs. $x_0 = 1.1495$. This small change almost doubles the final bacteria population. As x_0 inches up, the bacteria population will explode. In fact, entering only $x_0 = 1.16$ will cause the population to grow beyond MATLAB's "largest number," as in the last lab.

Try a small initial population, such as

$$\boxed{r = 1 \quad A = 1 \quad B = 12 \quad C = 1 \quad x_0 = 0.1} \tag{7.2}$$

without varying any parameters. You see very little chemical is used. Due to the small initial count, the population never gets large enough for the byproduct to be as insignificant as before. Now try $x_0 = 0.0001$. Virtually no chemical is used.

Now examine the role of A. Enter (7.1) again, varying with $A = 1.1$. The chemical now aids the growth more. As expected, more chemical is used. However, almost the same amount is used in both systems until about $t = 0.4$. At this point, the byproduct's effect has apparently reached a threshold where the rate of chemical use should be increased more quickly. The positive effect of the chemical is greater in the second system, so more chemical is used there. Similar to what we saw above, as A is increased, the bacteria population will explode. Only $A = 1.4$ is needed to breach MATLAB's limit.

If we decrease A, the chemical will have less positive effect, and less will be used. With $A = 0.4$, a moderate amount is used, while almost none is used when $A = 0.01$. If $A = 0$, then the chemical has no positive effect at all. It is broken down by the bacteria, with no benefit, to create a harmful byproduct. Not surprisingly, no chemical is used when $A = 0$, regardless of the other parameters.

Now enter (7.1), varying with $B = 20$. As expected, with more harmful byproduct, less chemical is used. It is worth noting that when A was adjusted, the use in chemical would stay almost the same for much of the beginning of the time interval, before increasing rapidly. Here, the chemical uses in the two systems begin apart, and the difference between them steadily increases. Like before, if we continue to increase B, less and less chemical will be used.

Of interest is the behavior of the control as B decreases. Try

$$\boxed{r = 1 \quad A = 1 \quad B = 0.1 \quad C = 1 \quad x_0 = 0.1} \, . \tag{7.3}$$

The control is now concave, where as in most of the other simulations we have done it has been convex. For smaller B, with appropriate x_0 so that MATLAB can handle the numbers, we see the control has less variation. The extreme of this occurs when $B = 0$. Enter

$$\boxed{r = 1 \quad A = 1 \quad B = 0 \quad C = 1 \quad x_0 = 0.01} \, . \tag{7.4}$$

Here, the control appears constant. As there is no negative byproduct, the optimal control is one of almost constant chemical injection. (Note: The control is most likely not exactly constant. The overall change is simply so small that MATLAB's graphing tools cannot display it. You may have been given an error message to this effect.)

Examine the role of the growth rate. Enter (7.1), varying with $r = 1.1$. In the second system, the bacteria has a higher natural growth rate. Therefore, the byproduct becomes less harmful more quickly, and more chemical can be used. Notice that the optimal control, and thus state, are virtually the same until about $t = 0.4$. Now compare $r = 1.1$ to $r = 1.2$. As r is increased, the control becomes more varied and reaches higher maximum levels. Conversely, as r is decreased, the control has less variation. Try $r = 0.8$. The control here experiences far less rapid growth at the end of the interval. Now try $r = 0.1$. The control is almost linear.

Finally, experiment with the weight parameter C. So far, we have used $C = 1$, meaning maximizing the final bacterial concentration and minimizing total chemical usage are of equal importance. Suppose the chemical is cheap and plentiful, and we are not very concerned with how much we use. Compare $C = 1$ vs. $C = 5$, using (7.1). More chemical is used to drive the bacterial concentration higher. The two systems differ, though, only after about $t = 0.7$. On the other hand, suppose the chemical is very expensive, and we are only willing to use a little to adjust the bacterial growth. Try $C = 1$ against $C = 0.2$. As expected, less chemical agent is used, although the effect of the weaker chemical schedule does not become apparent immediately.

Exercise 7.1 Calculate the necessary conditions for this lab problem. Using the adjoint equation and transversality condition, show $\lambda(t) > 0$ for all t, so that the characterization of the optimal control

$$u^* = \frac{Ax\lambda}{2(1 + B\lambda e^{-x})}$$

is well-defined.