ENGR 695 Advanced Topics in Engineering Mathematics Fall 2024

Lab #5: Solving Initial Value Problems in *Matlab*

Introduction

This lab exercise explores the solution of initial value problems in *Matlab*. The `ode45` command solves coupled sets of first-order ordinary differential equations (ODEs) of the form

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}\left(t, \mathbf{y}\right), \quad \mathbf{y}\left(t_0\right) = \mathbf{y}_0,$$

where **y** and **f** are N-element vectors of dependent variables and functions and N is the number of equations in the system. For first-order ODEs, $N = 1$, and for second-order ODEs, $N = 2$. The vector $\mathbf{y}(t_0)$ contains the initial condition(s). Although the system above is formulated using t as the independent variable, there is no reason why x cannot be substituted for t.

Second and higher-order ODEs can be solved using `ode45` by expressing the original ODE as a system of N first-order equations, where N is the order of the original ODE. Although we have not covered how to do this in class, the examples below should give you the general idea.

Before beginning, download the *Matlab* script `Lab5start.m`, which is available at the course Moodle site in the "Lab Materials" section. You should set up a separate folder on your own computer and/or in your Bucknell private Netspace for your ENGR 695 lab activities. You might also want to locate and keep handy the last page of the Lab #1 handout entitled "Important *Matlab* Commands for Linear Algebra."

Procedure

To help you become familiar with the syntax of the `ode45` command and to demonstrate how to solve common ODEs, we will work through two examples together. You will then attempt to solve a slightly more complicated IVP on your own.

We start with the first-order IVP given below:

$$y' + 0.3y = 0, \quad y\left(0\right) = 2.$$

1. The first step is to express the problem in a form suitable for `ode45`. The command wants to see a set of equations with first derivatives on the left-hand sides and a function of the dependent variables (and also possibly independent variables, but that is a more advanced topic) on the right-hand sides. In this case, we have a first-order equation, so the derivative "vector" and function "vector" have one element each; that is, they are actually scalars. Thus, for this first-order problem, we have

$$y' = -0.3y, \quad y\left(0\right) = 2.$$

2. Examine the function `fun_exp` in the script `Lab5start.m` that defines the right-hand side, paying particular attention to the syntax to see how this is done. Note that `dydt` is the value returned by the function. The value is defined only within the function code and not in the main program (the section that calls `fun_exp`). That is, `dydt` is a "local" variable known only to the function `fun_exp` and not a "global" variable known to the main routine as well.
3. Look carefully at the `ode45` command in the main part of the script. The first argument, `@fun_exp`, indicates that `ode45` should call the function `fun_exp` at the bottom of the script. (The "`@`" symbol appended to `fun_exp` indicates that it is a *function handle*, which is a variable that contains the address in computer memory where the function begins.) The second argument, `[0 ; 10]`, indicates that the solution should be calculated over the range of *x* values from 0 to 10. The third, argument, `2`, is the initial condition.
4. Run the script and observe the results. If any errors appear, work through them until the code runs successfully. The generated plot should display a decaying exponential that has the values $y = 2$ at $x = 0$ and $y = 0.0996$ at $x = 10$. The `ode45` command should have generated `x` and `y` vectors that you can display yourself (by typing '`x`' or '`y`' in the *Matlab* command window) to verify that the proper values have been calculated. If you want to display the values of *x* and *y* side-by-side, type `[x y]` at the prompt.

We will next solve the second-order IVP

$$y'' + 4\pi^2 y = 0, \quad y(0) = 20 \ \text{ and } \ y'(0) = 0 .$$

1. The `ode45` command wants to see a set of equations with first derivatives on the left-hand sides and a function of the dependent variables on the right-hand sides, but this time we have a second derivative. We can turn a second-order equation into a system of first-order equations by defining a new dependent variable y_2. The original variable *y* becomes y_1. Then, we set $y_2 = y'_1$. The second-order problem transforms into the first-order system

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= -4\pi^2 y_1 \end{aligned} \quad \text{with} \quad \begin{aligned} y_1(0) &= 20 \\ y_2(0) &= 0 \end{aligned}$$

   Note that the boundary condition on the derivative of the original dependent variable y' has become a boundary condition on the initial value of the new dependent variable y_2.
2. Modify the *Matlab* function `fun_fourier` that defines the right-hand side. Some of the code that you need has been provided in `Lab5start.m`. The variable `dydt` must now be a two-element vector that returns the right-hand side values y_2 and $-4\pi^2 y_1$.
3. Comment out the `ode45` command that operates on `@fun_exp`, and write a new `ode45` command (or modify the existing one) that evaluates the equation over the interval $x = 0$ to $x = 2$. (Commenting the old command saves it so that you can either run it again later or use it as an example.) Also provide a two-element initial conditions vector that has appropriate initial values for the variables y_1 and y_2. The help documentation for `ode45` states that the initial condition vector should be a column vector, but row vectors seem to work as well. Try it both ways!

4. The `ode45` command will return a column vector for x and a two-column matrix `Y`. The first column of `Y` contains y_1, and the second column contains y_2. The two columns of the `Y` matrix have the same length as the vector `x`. Set the variable `y` (lower-case) equal to the appropriate column vector of matrix `Y` (upper-case) that contains the solution to the problem.
5. Run the m-file and observe the results. Work through any errors until the code seems to be working. The generated plot should display a couple of cycles of a sinusoid with $y(0) = 20$ (one of the initial conditions). Solve the IVP manually (i.e., using a "pencil-and-paper" method) and compare your solution to the plot generated by the script. Does the plot look correct? If not, think about what you have entered for the initial conditions, which column you selected for the solution, and/or how you defined the right-hand-side function. Make any necessary corrections.

Finally, manually solve the second-order IVP below and then obtain a solution using the `ode45` command. Compare your manual solution to the plot generated by the script. The solution should be a concave-up, monotonically increasing curve that has the values $y = 5$ at $x = 0$ and $y \approx 7.9$ at $x = 0.5$. Save the modified version of `Lab5start.m` that represents your solution.

$$y'' - 2y' - 3y = 0\,, \quad y(0) = 5 \;\text{ and }\; y'(0) = 0 \quad \text{for} \quad 0 \le x \le 0.5\,.$$

Assistance will be provided as needed, but try to deduce on your own how to complete as much of the work as possible.

After you have completed the lab activities, e-mail to me the final version of your *Matlab* script that has been modified to solve the last problem described above. Make sure that your script generates the correct plot. Change the file name to `LName_Lab5_fa24.m`, where `LName` is your last name.

Lab Scoring and Submission Deadline

Your score will be based primarily on your submitted *Matlab* script and will be determined according to the rubric posted on the Laboratory page at the course web site.

If you do not complete the exercises during the lab session, then you may submit your documentation as late as 11:59 pm on Friday, October 11. If the documentation is submitted after the deadline, a 10% score deduction will be applied for every 24 hours or portion thereof that the item is late (not including weekend or fall break days) unless extenuating circumstances apply. No credit will be given five or more days after the deadline.