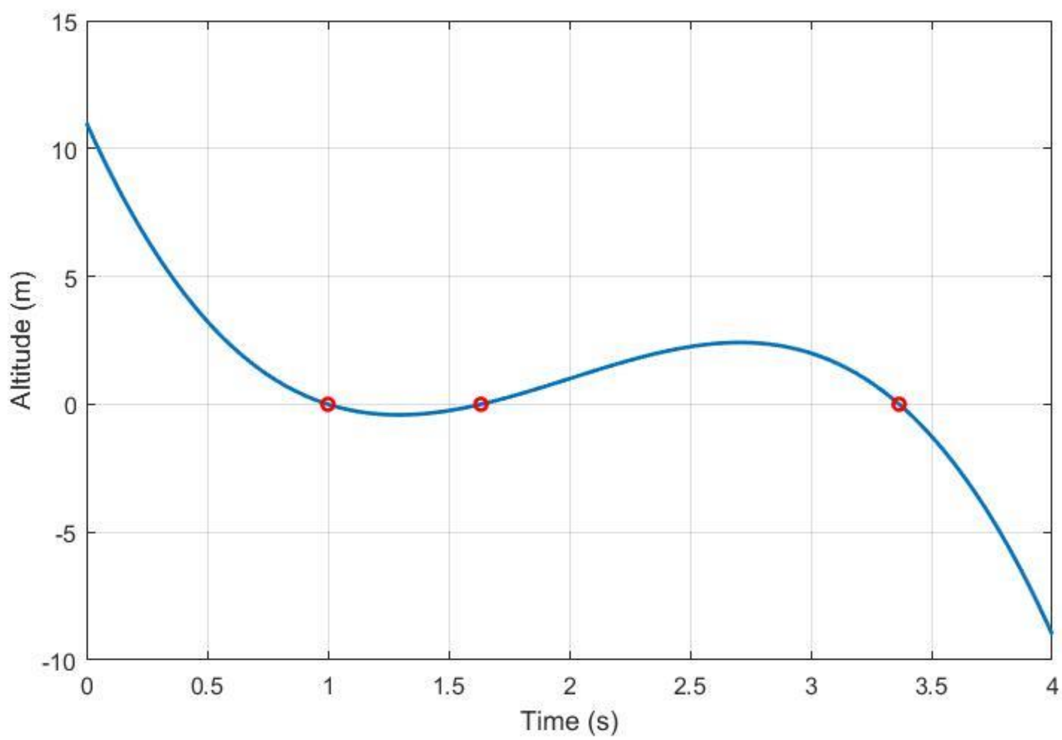
UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING
First Year Program – Core 8 and TrackOne

FIRST YEAR PROGRAM
ENGINEERING PROBLEM SOLVING LABS

MAT188: Laboratory #2
Scripts and Visualization



SCRIPTS AND VISUALIZATION

This lab will introduce you to creating and running scripts, which are useful for organizing and saving your MATLAB commands. You will also be introduced to the concept of indexing of vectors and basic two-dimensional plotting techniques. Through these you will develop your ability to visualize functions and use these plots to find graphical solutions to a problem. Both of these are important elements of the **Engineers' Toolbox**.

Learning Outcomes

By the end of this lab you will be able to…
1) Create simple scripts in MATLAB,
2) Use indexing techniques to access particular elements of a vector,
3) Create two-dimensional plots with proper labels, and
4) Understand how engineers use visualization and graphical techniques to solve problems.

Preparation *(Required to do* ***before*** *you come to the laboratory session)*

1. Complete Sections 4 through 11 of **MATLAB Onramp** online exercise at https://matlabacademy.mathworks.com (i.e., *Section 4: Importing Data* to *Section 11: MATLAB Scripts*). This should take you about an hour. Remember your login information for the Mathworks account you create to complete the Onramp, since you will be asked to log in during the lab to show your Lab TA your Progress Report at the start of the lab session. **Make sure you are completing the MATLAB Onramp using version R2017a.**

MATLAB Skills and Knowledge: *Scripts and Indexing of Vectors*

Making a Simple Script

So far you have learned how to use the command line in the Command Window to execute individual commands one at a time. A ***script*** (or m-file) is a file with a `.m` extension that contains multiple lines of code to be run at the same time (in order). Scripts are useful to organize your code and save it for later use and modification.

Go to the **HOME** ribbon at the top of the MATLAB Command Window and click *New Script*. The Editor window will open, similar to Figure 1. You can adjust the sizes of the Editor and Command Window by moving the line between them. It is beneficial to have both open while developing your script. Use the Command Window for quick testing of commands and output, but develop your actual code in the Editor.

In your script, type commands to plot the function $f(x) = e^{-x}\sin(4\pi x)$ using 101 points in the interval $[0, 5]$. You can refer to the **MATLAB Summary** document or Lab 1 to review basic plotting.

In the **EDITOR** ribbon at the top, click the *Save* button and name your script `basic_plot.m`. **You must always save your scripts in the Current Folder (shown to the left of the script editor) with an `.m` extension to be able to use them.**

In the **EDITOR** ribbon at the top, click the *Run* button to execute your script. Alternatively, you can type the script name (without `.m`) in the Command Window to run it. If there is an error, it appears in red in the Command Window. You can click the link in the error message to navigate to the line that caused the error.

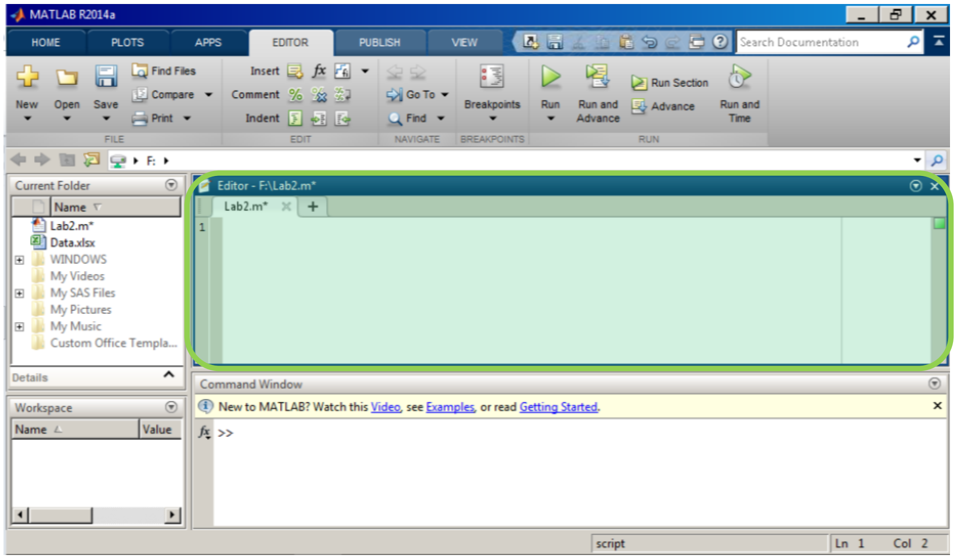


Figure 1. The MATLAB Editor when you create a script.

A ***comment*** is a line of text that starts with `%` and is ignored by MATLAB when executing commands. Comments are useful for inserting explanations or reminders to the person reading the script. Go back to the Editor and add this comment to the beginning of your script:

```
% Plots a decaying sine function
```

The `%` sign can also be used to “comment out” specific lines of code that you do not want to run but want to keep for later.

Save your script. If you run it again, it will produce the exact same result.

Exercise: Modify your `basic_plot.m` script so that it plots the function $g(x) = x^2 \sin(4\pi x)$ in the interval $[-8, 8]$. Use the `grid on` command to add a grid to your plot.

Indexing Vectors

Since a vector is a sequence of numbers, each value (or element) is referred to with an index, where **index 1 is the first element**.

For example, say we have the following vector:

```
>> x=[3 7 -4 1 2 11]
```

To access the 4^{th} element of the vector you would enter:

```
>> x(4)

= 1
```

You can use `end` to refer to the last index. For example, to access the last element of the vector:

```
>> x(end)
```

You can also use the equals sign (=) to assign a new value to a particular element of a vector. To change the 2nd element:

```
>> x(2)=99
```

If you try to use an element that doesn't exist, MATLAB will produce an error:

```
>> x(10)
```

You can use the colon operator to access a range of elements. For example, to access the first four elements:

```
>> x(1:4)
```

To access every element except the last:

```
>> x(1:end-1)
```

Exercise: Return to your `basic_plot.m` script and use vector indexing

The Engineers Toolbox: Visualization and Graphical Solutions

Engineering Problem Solving Process

One of the most important skills that you will learn in your undergraduate engineering program is the ability to solve problems in a careful and methodical manner. One of the primary purposes of these MAT188 laboratory exercises is to give you the opportunity to practice this process through the use of numeric computation (i.e., MATLAB).

Following George Pólya's Problem Solving Process, we can summarize this approach as:

Understanding the Problem

- What is the unknown? (GOAL)
- What relevant information is provided? (KNOWNS)
- What is the condition? What fundamental principles are related? (FOUNDATION)
- Draw a diagram. (VISUALIZATION)

Devising a Plan

- Consider the unknown. What are the possible connections between the information provided and the unknown.
- Can the problem be restated in a more useful manner?
- Do you know a related problem? Can you use its answer or its method?

Carrying Out the Plan

- Check each step as you carry out the plan. Is it leading you in the right direction?

Looking Back

- Can you check the result? How about the solution?
- Is the answer complete? Have you solved the problem asked?

Problem: *Analysis of Drone Flight Data*

During the initial design of a control system for a flying drone, measurements of the altitude of the drone during a test flight were recorded. After analyzing the data, it was determined that this data could be reasonably represented over the interval $t = 0$ s to $t = 4$ s by the polynomial:

$$h(t) = -2(t-2)^3 + 3(t-2) + 1.$$

Figure 1: U of T Engineering Professor Angela Schoellig and her graduate students demonstrating their drones at UTIAS[1].

The control system was supposed maintain the drone at a "hover height" of 9 m, which in this case is defined by $h = 0$. You are asked to use MATLAB to determine the following three things:

1) *Level 1 (Beginner)*
   At what times does the drone reach the hover height exactly?
   Create a properly labeled plot that shows these times (use the `xlabel`, `ylabel`, and `legend` commands). Your plot might look similar to that which is on the first page of this lab handout. *Hint*: You may find the `hold` command useful. **Each student will need to submit their plot through the MAT188 PRA Page in Blackboard using the submission instructions below.**

2) *Level 2 (Advanced)*
   How long does the drone stay within 30% of the targeted hover height? *Hint*: The `find` function may be useful.

3) *Level 3 (Expert)*
   Estimate the velocity with which the drone hits the ground. How does this compare to the analytic solution?

For each of these levels, team up with the person next to you to work through the Engineering Problem Solving Process. We suggest you each write a script for your solution so that you can easily edit or adapt your solution approach.

Understanding the Problem

What information do you know?
What is the question actually asking you for? What will a good answer look like?

Devising a Plan

How could you use your current understanding of MATLAB to solve this problem, or answer this question?

Carrying Out the Plan

Execute your plan. Assess your strategy as you progress, and refine as needed.

Looking Back

Did you properly answer the question?
Is your answer/solution presented correctly and reasonably (i.e., correct number of significant digits, etc.)

Bonus: How might you use the `roots` function to answer the *Level 1* question?

---

[1] Picture source: http://news.engineering.utoronto.ca/drone-researchers-industry-experts-talk-autonomous-flight-international-symposium/

Lab #2: Analysis of Drone Flight Data

Level 1 Exercise Submission Details

As part of this lab, each student is required to submit two things:

1) A PDF file of their properly labeled plot that they create for the *Level 1* exercise (you can save your plot in MATLAB directly to this format).

2) A brief description of how you applied the Engineering Problem Solving Process to solve this Level 1 problem? In particular, describe at least two ways that you considered approaching the problem during the Devising a Plan stage.

You will submit these two items through the MAT188 PRA Portal page:
Fall-2017-MAT188H1-F-PRA0108.PRA0109.PRA0113.PRA0 (Fall 2017 MAT188 Linear Algebra Labs)

Note, this is NOT the main MAT188 lecture Portal page.

At that page you will find the MAT188 Engineering Problem Solving Lab #2 - Analysis of Drone Flight Data Submission assignment in the Laboratory Assignments folder. There you can upload your PDF or JPEG file, and submit a brief "Text Submission" by clicking the **Write Submission** button.

Ideally, you will submit this assignment by the end of your Lab #2 session, but at the very latest you can submit this by Friday, September 22nd at 11:59PM.