LAB 0:
MATLAB TUTORIAL

ME 436 Heat Transfer - Fall 2017

Introduction

This short tutorial is designed to familiarize you with MATLAB and the general structure of the homework assignments to come. Be sure to read through this document carefully and check out the resources section at the end. Finally, if you have any questions or difficulties, be sure to contact your TA as soon as you can.

Scope

The purpose of each homework assignment is to save *you* time. Once completed, you will have a plug-and-play mathematical model ready to go. As some of you may be aware, this is a slight departure from previous labs, in which much of the analysis is generally performed after data collection. By shifting some of the weight from *post-processing* onto the *pre-lab*, this frees up precious lab time and provides a more balanced workload for everyone. More importantly, this approach allows us to dive deeper into the experiments and ultimately ask more interesting questions.

So, rather than having each group build an Excel sheet from scratch to analyze your collected data (at home, after the lab), we will provide you with a set of partially completed MATLAB scripts and sample data. Then, your homework assignment (individual) will be to complete these MATLAB scripts and answer a few short questions. To receive full credit, you must submit your responses at the *beginning* of your lab period. You will then use these mathematical models in-class to quickly process your experimental data.

Why Matlab?

From our experience, far too much time is lost slogging through the minutiae – rather than focusing on the more critical aspects. With the assistance of MATLAB, we can essentially sidestep many of these uninspiring – and often mundane – tasks that detract from the core laboratory objectives. For example, with Excel, locating cell numbers,

debugging, formatting charts, and performing repetitive calculations *(property look-up's, interpolation, unit conversions &, etc.)*, can be very mundane... Taken together, these tasks can quickly snowball into a much larger (and time-consuming!) task than initially anticipated. More importantly, however, these mundane details dilute the overall take-away message. We should be more focused on performing experiments and exploring data - rather than fighting with computer software. That being said...

Not a Matlab fan?

No problem. This isn't a MATLAB course – nor is it an Excel course – we're interested in Heat Transfer. Thus, we only expect the *bare minimum* on the programming front. Anything beyond entry-level will be explicitly provided to you.

Getting Started

As stated above, this short tutorial is designed to help familiarize you with the basic format of the homework assignments. In addition, this document has been written to provide a comprehensive explanation at each step along the way. You may want to use this as a reference throughout the semester. In general, the assignment instructions will be written more concisely.

First, you will always need to download the starter code from Blackboard (Bb), and *extract* (unzip) its contents to the directory in which you wish to complete the exercise[1]. Before attempting, make sure that you have completed all of the *prerequisites* below.

Prerequisites

Generally, in this section you will find a short list of the essentials that should be completed *before* attempting the assignment. For *example*, in Lab 1 you will see:

- Review the **textbook sections:** 2.1, 2.2 & 3.1-3.1.4,
- Review the **experiment procedures**,
- Watch the **pre-lab videos** on Blackboard (Bb), and
- Complete the **pre-lab quiz** on Bb.

Files included in this exercise:

Now we are ready to start the assignment. Once you have unzipped the contents of the starter package, you should have the following files and folders:

- ex0.m - *main file that you will work out of.*
- /lib - *functions and files that work behind the scenes. It's best not to touch them.*

[⋆] plotData.m - *function that plots your data - you <u>will</u> need to adjust this one.*

---

[1]*Note: when using the Heat Transfer Lab computers, you <u>must</u> extract your code to the local hard drive: C:/temp/, NOT to the server (ie., Desktop, Group Folder, &, etc.) See additional details at the end of this document.*

$\star$ indicates files that you will need to complete.

In general, you will be using the script titled `ex<lab no>.m`, without making any sensible changes to it. However, sometimes you will be required to comment out the `return` commands at the end of each section, to better understand each step of the analysis. More information about this will be provided below.

0.   Warm-Up & System Check

Before starting, it is often useful to understand the data by first visualizing it. In particular, for the upcoming experiment (Lab 1), we need to make sure that our collected data is at steady–state. Hence, for this first tutorial, we will load a sample dataset and then plot it using `MATLAB`. In the process, we will also make sure that our system is setup correctly.

0.1.   Load Data

Now, make sure that you have `MATLAB` started and the script titled 'ex0.m' opened up in the editor. If you have any troubles with this step, be sure to check out the resources at the end of this document or ask your TA.

Now, with `ex0.m` open, the first thing you will see is the *Documentation Header:*

```
%% ME 436L Heat Transfer
% Lab 0 | MATLAB Tutorial
%
%  INSTRUCTIONS
%  ————————————
%
%  This file contains code that will guide you through the assignment. Be
%  sure to follow the instructions provided in the attached document.
%
%  NOTE: You will need to complete the following functions:
%
%       plotData.m
%
% For this exercise, you will be required to comment out the 'return'
% command line at the end of each section, to ensure that you fully
% understand each step of the analysis.
%%
% Written By: Spencer Pfeifer | Revision: Paola G. Pittoni
% Date:   8/24/17
%
%#ok<*UNRCH>
```

This provides information as to what the script does, who wrote it, and when. Now, skipping the *Initialization* section, we get to a section titled *'Load Data'*. Here, we load a sample dataset from a text file and store it into the variable `M`:

```
% load tab separated data
M = load('data.txt');
```

Once you have pressed 'Run,' you will now have a matrix M of size 93×11 ($rows \times columns$), or $M \in \mathbb{R}^{93 \times 11}$. The columns in M are defined as follows:

1	2	3	...	9	10	11
$time(s)$	TC_1	TC_2	...	TC_8	$Voltage, V$	$Current, A$

In which TC_n refers to the thermocouple number, read in degrees Celsius. Next, you will see a line that looks similar to the following:

```
% COMMENT ME OUT!!
break_msg; dbstack; return;
```

This is what we refer to as a `return` command. In short, this line will exit the program and leave you with a message that looks similar to the following:

```
----------------------------------------------------------
                    Lab 0: PLOTTING
----------------------------------------------------------

Loading data...
Done.

Program exited. Uncomment this line to continue.
> In ex0 (line 40)
>> |
```

Figure 1: *Sample Matlab Output. Tip: You can click on the line number and it will take you to the exact line of the* `return` *function.*

You will need to *comment out* this line (place a % out front) in order to continue.

0.2. Extract & Plot Data

Next, let's separate out this data into more meaningful variables.[2]

```
t = M(:,1);          % get time vector, [s]
dat = M(:,2:9);      % get temps, [C]
V = M(:,10);         % Voltage, [V]
I = M(:,11);         % Current, [A]
```

You should take this moment to press `Run` and open the *workspace* to confirm the above variable assignments. Now, after removing the `return` command, let's plot our temperatures with time by executing the following lines:

```
figure;          % open new figure
h = plot(dat);   % plot the data
set(h, 'LineStyle', '—', 'LineWidth', 2)   % make sure the lines are readable
```

---

[2]*Note: if you are not familiar with the syntax, check out: Matrix indexing in Matlab*

```
% don't forget to add labels!!
xlabel('Time [s]');
ylabel('Temperature [C]');
title('Transient Data, T/C','FontSize',16);
grid on
```

Now, if everything was done correctly, you should obtain the Fig. 2 below.
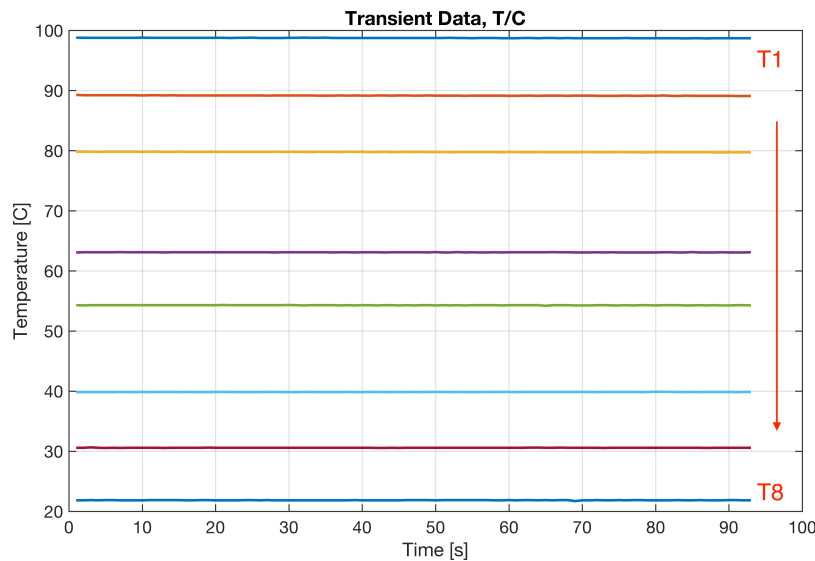
Figure 2: *Transient temperature data*

A few quick questions to ask:

> **Question Set: 1.** *Sanity Check.*
> - Does your plot match Fig. 2?
> - Is all of our data at steady-state?

- *Note: This blue box will be used to denote your deliverables for the assignment. Be sure to pay close attention.*

> *Checkpoint* - If you had any difficulties obtaining Fig. 2, **stop here** and contact your Lab Instructor. You may run into more troubles ahead.

---

1. Plot Steady-State Data

Now, let's plot the spatial temperature distribution. To do this, we simply average the transient data. This is easily done in MATLAB by the $mean()$ command:

```
% take an average of the last (~20) points
N = 20;
Tm = mean(dat(end—N:end,:));      % [C]
Vm = mean(V(end—N:end));          % [V]
Im = mean(I(end—N:end));          % [A]
```

Next, we call a `plotData.m` function to plot our data. As-it-is, this plot function *is incorrect* and will not run properly. Therefore, you will need to open `plotData.m` and adjust code accordingly in order for it to run properly. The sections that you will need to complete are outlined:

```
% plot first three T/Cs
plot(x(2:4),y(1:3),'ko—','MarkerSize', 8);

% plot middle two T/Cs
% ==================== YOUR CODE HERE ====================

% plot last three T/Cs
% ==================== YOUR CODE HERE ====================


% set labels & axis limits
% ==================== YOUR CODE HERE ====================
xlabel('');
ylabel('');
```
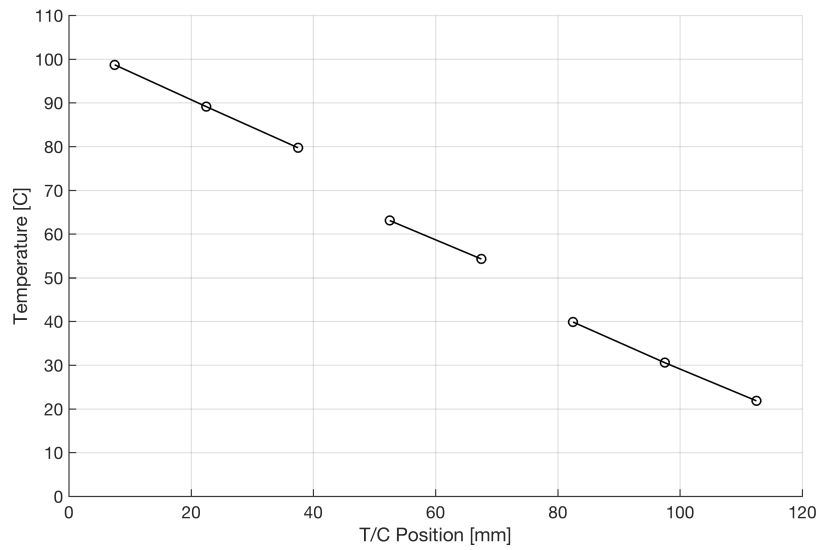
- *Note: Pay close attention to the instructions.*

When you're finished, you must **go back** to `ex0.m` before you can press `Run`. If you attempt to run the function `plotData.m`, you will receive an error. If your adjustments were correct, and `ex0.m` was properly executed, you should obtain Fig. 3.

Figure 3: *Transient temperature data*

Then, if everything was done correctly above, after commenting out the following `return` command, you will arrive at a figure that resembles Fig. 4.
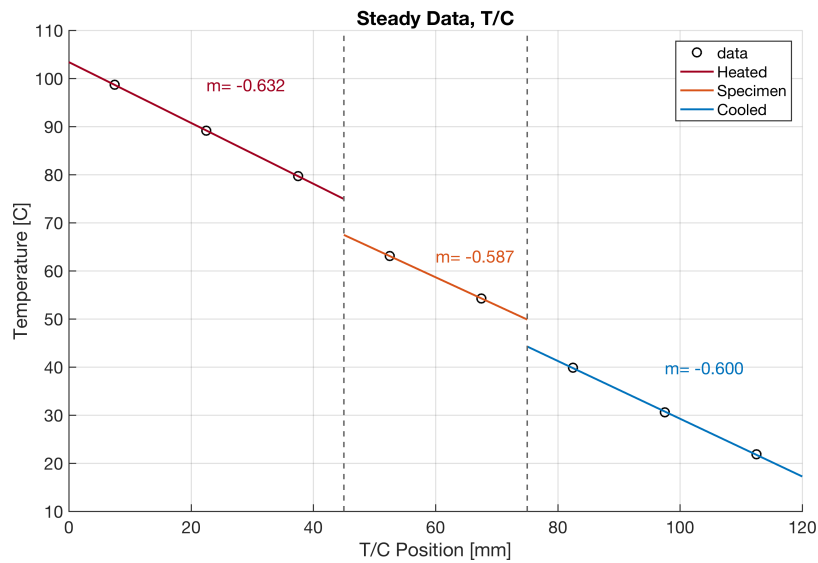
Figure 4: *Steady-state T/C data*

Here, we have added regression lines as well as some labels that will make more sense when we get to Lab 1.

The box below outlines the *deliverable* for this assignment:

> **Deliverable 1.** *SAMPLE: Steady-state temperature distribution plot* (Fig. 4). Export your figure to an image: `File » Export Setup (change the properties as desired)» Export`, and include it in your submission.

- *Note: Do not take a screenshot! Export the figure to a* `.png`.

Submission

Generally, once all of the deliverables have been completed, you will simply place the output into a single document, and hand it to your TA at the *beginning* of lab. For this exercise, print Fig. 4 for your 'participation' grade.

Notes:

Heat Transfer Lab:

- `MATLAB` must be run from the local hard drive: `C:/temp/`, **NOT** on the server *(ie., Desktop, Group Folder, &, etc.)*.

    - ***Reason:*** This will cause a chain of problems. The server directories are shared across all HT Lab computers, and synchronized every few minutes.

    - *Common scenario:* Two people unknowingly edit the same document. Once the server synchronizes, one of those two people **will lose all of their work.**

- You *must* close Matlab when you're done for the day.

    - ***Reason:*** The `addpath(`'./lib'`);` command adds your `./lib` folder to the `MATLAB` path, while the program remains open. If you do not close Matlab the next section will use your data.

General:

- *Do not rename any functions!* These are name (and case) sensitive. If any functions are renamed, your script will not find them!

- Whenever you see the dialog below, be sure to choose *Change Folder*. Otherwise you may inadvertently add unwanted files to the `MATLAB` path.
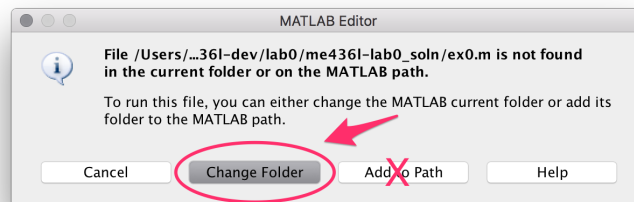
- Remember, you cannot *Run* a function by pressing the '`Run`' button. You can only do this for *scripts*. See the Resources section below for more information.

- **Do NOT** use screenshots for reporting your results.

    - ***Reason:*** The default Matlab *figure* window is designed for the screen, *not print*. This will turn out very small and blurry.

    - Simply do the following: `File » Export Setup » Apply to Figure » Export » Choose PNG Format`.

Resources

This section provides a list of helpful resources for getting started with Matlab. As always, your TA and classmates will be valuable resources as well.

- Mathworks.com: Here is a variety of resources produced by Mathworks. This covers the basics of Matlab, including a getting started tutorial.

- Lynda.com: A video-based learning platform for a variety of topics. Normally this is a paid service, but ISU provides free access. To use this resource, do the following:

  - iastate.edu » Sign Ons » More Sign Ons... »
  - Then click the Lynda.com button.
  - Once you fill in your credentials, type MATLAB into the search bar and select *'Learning Matlab'*
  - Here you will find a selections of 2-3min videos explaining specific aspects of Matlab.

- Getting Started with Matlab: A youtube playlist produced by Mathworks.

- What is Matlab and Why is it Used? - A corny video explaining what Matlab is commonly used for.

- **Finally, Ask questions!** It doesn't take much to e-mail your TA! If you don't ask, we cannot help!