GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 2026 Spring 2025
Lab #0: Introduction to MATLAB

Date: 06 – 10 Jan. 2025

Important Note: This preparation lab is intended to guide you through the MATLAB tool for the subsequent assignments. If you have previous exposure to MATLAB follow this lab to refresh your MATLAB skills. If you are new to MATLAB study the two tutorials listed in the Lab Page to acquire necessary MATLAB skills and use this lab as an exercise. There is also a guide to install MATLAB on your computer. DSP-First MATLAB files can also be downloaded from Lab Page or the Course Home Page.

This initial lab is different from the rest of the labs as it does not require you to submit a completion report. It does include the verification step (see below), which will provide examples for future labs. In subsequent assignments, each lab will guide you through three stages: Pre-Lab, Exercise and Verification of Completion. The Pre-Lab stage requires you to read the assignment and familiarize yourself with what the lab is about and what additional tools, equipment or apps you may need in the main exercises. In the Exercise stage, you need to follow the instructions to complete a number of small tasks. Specific steps are to be taken to record the result when you complete a task. The results are then collated electronically into a Completion Report in the Verification of Completion stage for submission and grading. The completion report page at the end of each lab assignment enlists what needs to be answered, recorded and/or attached for submission. These include screen shots or files that show the output of the assigned exercise. All exercises will be done on your own computer. Instructors will launch online BlueJeans interactive sessions during your original assigned lab time slots to assist you.

The labs will be held in room 2440 of the Klaus building. Your GT login will work, if you specify the "Windows domain" to be AD. **Special Lab Instructions in Spring 2025:** The labs will be held in-person or remotely through Zoom. Students registered in a particular lab session are required to be present at the designated times. Attendances will be taken before each class. A total of six labs will be conducted in spring 2024. Each lab will typically last two weeks (except three weeks for Lab 6. The first week is devoted to students' **Q&As** and the second is for students' **demos** to the instructors for codes and lab results. Students will be grouped into teams of 2 or 3 by instructors before starting Lab 1. For each lab session, there will be two instructors administrating all activities. Each team will be given a breakout period of 15 minutes in each session for Q&As and verifications. Students are encouraged to discuss lab contents in teamwork. At the end of each lab, each student are required to turn in an individual lab report in a single pdf fill, containing answers to all lab questions, including codes and plots. Georgia Tech's **Honor Code** will be strictly enforced. See CANVAS Assignments for submission instructions.

Forgeries and plagiarism are a violation of the honor code and will be referred to the Dean of Students for disciplinary action. You are allowed to discuss lab exercises with other students, but you cannot give or receive any written material or electronic files. In addition, you are not allowed to use or copy material from old lab reports from previous semesters. Your submitted work must be your own original work.

---

1 Pre-Lab

1.1 Overview

MATLAB will be used extensively in all the labs. The primary goal of this lab is to familiarize yourself with using MATLAB. Please read Appendix B: *M-file Programming in* MATLAB for an overview. Here are three specific goals for this lab:

1. Learn basic MATLAB commands and syntax, including the `help` system.
2. Write and edit your own script files in MATLAB, and run them as commands.
3. Learn a little about advanced programming techniques for MATLAB, i.e., vectorization.

1.2 Movies: MATLAB Tutorials

There are a large number of online Real-Media movies on basic topics in MATLAB, e.g., colon operator, indexing, functions, etc.

1.3 Getting Started

After logging in, you can start MATLAB by double-clicking on a MATLAB icon, typing `matlab` in a terminal window, or by selecting MATLAB from a menu such as the `START` menu under Windows. The following steps will introduce you to MATLAB.

(a) Run the MATLAB help desk by typing `helpdesk`. The help desk provides a hypertext interface to the MATLAB documentation. Two links of interest are **Getting Help** and **Getting Started with MATLAB**.

(b) One of the tabs in the Help Desk is called Demos, where you can see information on getting started and some other basics of MATLAB.

(c) Explore the MATLAB help capability available at the command line. Try the following:

```
help
help edit
help plot
help colon        %<--- a VERY IMPORTANT notation
help ops
help zeros
help ones
lookfor filter      %<--- keyword search
```

Note: it is possible to force MATLAB to display only one screen-full of information at once by issuing the command `more on`).

(d) **control-C** will stop the execution of any MATLAB command. For example, using **ctl-C** while `lookfor` is running will force it to stop and print out all the results it has found so far.

1.4 Calculate and Plot

(a) Use MATLAB as a calculator. Try the following:

```
pi*pi - 10
sin(pi/4)
ans ^ 2       %<--- "ans" holds the last result
```

(b) Do variable name assignment in MATLAB. Try the following:

```
x = sin( pi/5 );
cos( pi/5 )       %<--- assigned to what?
y = sqrt( 1 - x*x )
ans                 %<--- what value is contained in ans?
```

(c) Complex numbers are natural in MATLAB. The basic operations are supported. Try the following:

```
z = 3 + 4i, w = -3 + 4j
real(z), imag(z)
abs([z,w])      %<-- Vector (or Matrix) constructor
conj(z+w)
angle(z)
exp( j*pi )
exp(j*[ pi/4, 0, -pi/4 ])
```

(d) Plotting is easy in MATLAB for both real and complex numbers. The basic plot command will plot a vector y versus a vector x. Try the following:

```
x = [-3 -1 0 1 3 ];
y = x.*x - 3*x;
plot( x, y )
z = x + y*sqrt(-1)
plot( z )    %<---- complex values: plot imag vs. real
```

Use `help arith` to learn how the operation `xx.*xx` works when `xx` is a vector; compare array multiplication (`dot-star`) to matrix multiplication. When unsure about a command, use `help`.

2 In-Lab Exercises

For the instructor verification, you will have to demonstrate that you understand things in a given subsection by answering questions from your lab instructor (or TA). It is not necessary to do everything in the subsections, i.e., skip parts that you already know. The Instructor Verification is usually placed close to the most important item, i.e., the one most likely to generate questions from the TAs.

2.1 Intelligent Tutoring System (ITS)

During this first lab you should run ITS and answer a few questions:

1. The link to ITS is: `http://its.vip.gatech.edu`. To access ITS, login with your AD username and password (same as for `t-square`)
2. Answer the few questions in the introductory section of ITS.
3. Then show your TA that you can review your answers.

2.2 MATLAB Array Indexing

(a) Make sure that you understand the **colon** notation. In particular, explain in words what each of the following MATLAB statements will produce (array length and values):

```
jkl =  0 : 6
nnn =  2 : 4 : 17
mmm =  99 : -1 : 88
ttt =  2 : (1/9) : 4    %<--- How many elements in the ttt vector?
tpi = pi * [ 0:0.1:2 ];
```

(b) Extracting and/or inserting numbers in a vector is very easy to do. Consider the definition of `xx` in the first line:

```
xx = [ zeros(1,3), linspace(0,1,5), ones(1,4) ]
xx(4:6)
size(xx)
length(xx)
xx(2:2:length(xx))
```

Explain the results echoed from each of the last four lines of the above code.

(c) Observe the result of the following two assignments, where xx was defined in part (b):

```
xx = [ zeros(1,3), linspace(0,1,5), ones(1,4) ];
yy = xx;
yy(4:6) = exp(1)*(1:3)
```

Now write a statement that will take the vector xx and replace the even indexed elements (i.e., xx(2), xx(4), etc) with the constant π^e. *Use a vector replacement, not a loop.*

Completion of Lab Results (on separate Report page)

2.3 MATLAB Array Operations

There are two kinds of multiplication in MATLAB: matrix multiplication and *array* multiplication. Many other operations such as division and exponentiation also have this dual character. MATLAB uses a period to change the behavior from a matrix operator to an array operator, e.g., dot-star (.$*$) instead star ($*$) for multiplication.

(a) The default is matrix multiplication when the "$*$" symbol is used; execute the following:

```
AA = [ 1, 2; 3, 4; 5, 6; 7, 8]
BB = [ 1, 2, 3; 4, 5, 6]
CC = AA * BB
DD = BB * AA
```

Explain why one of the multiplications fails. In the case where the multiplication succeeds, explain what the dimensions of the resulting product matrix will be.

(b) In DSP it is often useful to perform an *element-by-element* multiplication, which we will call array multiplication. For example, to multiply one sinusoid by another we would use the ".$*$" operator:

```
nn = 1:9;
qq1 = 7*cos(0.1*pi*nn-pi/2);  qq2 = cos(pi*nn);
qq = qq1 .* qq2
zz = qq1 * qq2
```

Explain why the multiplication that attempts to create zz fails. Notice that the dimensions of the array-product matrix qq are the same as the dimensions of nn. Explain the values in the vector qq2.

Completion of Lab Results (on separate Report page)

2.4 MATLAB Script Files

(a) Experiment with vectors in MATLAB. Think of the vector as a list of numbers. Try the following:

```
xk = cos( pi*(0:11)/4 );  %<---comment: compute cosines
```

How many values of the cosine are stored in the vector xk? What is xk(1)? Is xk(0) defined?

Notes: the semicolon at the end of a statement will suppress the echo to the screen. The text following the `%` is a comment; it may be omitted.

(b) (A taste of vectorization) Loops can be written in MATLAB, but they are NOT the most efficient way to get things done. It's better to **always avoid loops** and use the colon notation instead. The following code has a loop that computes values of the cosine function. *Note:* the index of `cc()` must start at 1.

```
cc = [ ];      %<--- initialize the cc vector to be empty
for k=-50:50
   cc(k+51) = cos( pi*k/30 );
end
plot(cc)
```

Explain why it is necessary to write `cc(k+51)` inside the loop. What happens if you use `cc(k)` instead? Also, explain the labels on the x-axis of the plot.

(c) **Rewrite the computation in the previous without using the loop** by following the style in part (a). Two MATLAB statements plus the `plot` command should suffice.

(d) Use the built-in MATLAB editor to create a script file[1] called `mylab1.m` containing the following lines:

```
tt = -5 : 0.01 : 10;
xx = cos( 0.5*pi*tt );
zz = 0.6*exp(-j*pi/4)*exp(j*0.5*pi*tt);
%
%<-- plot the real part, which is a sinusoid
plot( tt, xx, 'b-', tt, real(zz), 'r--' ), grid on
title('Test Plot of a TWO sinusoids')
xlabel('Time (sec)')
```

Explain why the plot of `real(zz)` is a sinusoid. Determine its phase (φ) and amplitude (A) from its definition. Make a calculation of the phase from a time-shift measured on the plot; zoom in to measure a peak location very accurately. Compare this value to the expected value from the formula that defines `zz`.

Completion of Lab Results (on separate Report page)

(e) Run your script from MATLAB. To run the file `mylab1` that you created previously, try

```
mylab1            %<---will run the commands in the file
type mylab1       %<---will type out the contents of
                  %    mylab1.m to the screen
```

[1] *When you save* this file or any of your MATLAB files *please do not save to the local hard disk.* Your computer account contains a private networked directory where you can store your own files. Use the MATLAB command `addpath()` to allow MATLAB to "see" your personal directory (usually the `Z:` drive).

Lab #0 (This is just for practising, no need to turn in.)
ECE-2026 Spring-2025
LAB COMPLETION REPORT

This page is an example of what results would be checked by your lab instructors.

Name: Rudra Goel Date of Lab: 01/07/2025

Part 2.2 Vector replacement using the colon operator:

xx(2, 2, length(xx)) = pi^exp(1)

Part 2.3 Array multiplication versus matrix multiplication:

zz is failed to be created since only the “*” operator is used implying matrix multiplication between qq1 and qq2; however, since qq1 and qq2 are considered 1x9 matrices each, the column size of the first matrix does not equal the row size of the second matrix making the matrix multiplication fail.

qq2 is a vector oscillating between -1 and 1 since it is the cosine of consecutive multiples of pi, which are completely on the real axis of the complex plane resulting in -1 and 1.

Part 2.4(d) Use Euler's formula, $e^{j\omega t} = \cos(\omega t) + j\sin(\omega t)$, to explain why the plot of `real(zz)` is a sinusoid. Determine the amplitude and phase of the sinusoid, and write those values in the space below:

$A =$ 0.6 $\varphi =$ -π/4

real(zz) is a sinusoid because the real part to any complex number is the cosine of the angle of that number, which varies with the vector tt

In the space below, **make a calculation of the phase from time-shift measured on the plot.**