

Created by G. M. Wilson, revised 12/23/2015

Revised 09/05/2016

Revised 01/07/2017

MATLAB Demo

In this lab, we will learn how to use the basic features of MATLAB needed to successfully complete a lab assignment, and how to create a lab write-up by properly formatting a `diary` file.

This MATLAB Demo assignment will not be collected. **However, it is imperative for you to carefully complete it, even if you have prior experience with MATLAB, since you will need these specific instructions for Labs 1 – 6 in Math 250C.**

Allow yourself enough time to do each MATLAB assignment, in order to think about the mathematical concepts and to carry out the instructions precisely.

The MATLAB computer environment is designed for matrix computations and it uses standard linear algebra notation. This makes the MATLAB commands short and easy to remember.

Preliminaries and Getting Started with MATLAB

Reading from Textbook: Before beginning the Lab, read through Section 1.1 of the textbook and work the assigned homework exercises for these sections to get familiar with matrices and vectors.

All of the MATLAB assignments require textbook readings. Most of this material is covered in class, but in each of Labs 1 – 6, you will need to allow yourself enough time to read certain material on your own. Such material will be clearly specified.

You can look at Appendix D of the textbook to learn more about MATLAB.

Version: Our labs are written to be compatible with MATLAB versions R2014b and above.

Obtaining MATLAB: Windows, Mac, and Linux versions of MATLAB R2015b are available as free downloads from <https://software.rutgers.edu>.

You may also access MATLAB as a webapp through <http://apps.rutgers.edu>. If you have never used the Rutgers apps service before, follow the directions presented there to activate your account. Once you successfully login to the Rutgers apps service, you can access MATLAB by right clicking your mouse, and navigating to APPLICATIONS > EDUCATION > MATLAB R2014B.

The Rutgers Computing Labs have MATLAB installed on their computers. Visit <https://oit-nb.rutgers.edu/computing-labs> for the lab locations and hours.

Demonstration of MATLAB: The basic mode of MATLAB is interactive. After you start the MATLAB program and obtain the prompt `>>`, you type commands in the *Command Window* that MATLAB then executes when you press the **Enter** key. If you have never used MATLAB before, we suggest you enter `demo` at the MATLAB prompt now. This will open a new window with documentation about MATLAB. Click on *Getting Started with MATLAB* for a quick tutorial.

Directory: You will need a dedicated directory on your computer for MATLAB files and output. Somewhere on your computer, make a directory (also called a folder) titled **MatlabFiles**. On the left hand side of the MATLAB window there is a list of the files and directories on your computer under the heading *Current Folder*. Navigate through the list to select your directory **MatlabFiles**.

Tcodes: You will use some instructional MATLAB *m-files* called *Tcodes* in several of the MATLAB assignments. These files are located on the website <http://math.rutgers.edu/courses/250/Tcodes>. For this lab, we will use the *Tcode*

`rdiag.m`.

Find the link to `rdiag.m` on the website, right click it, and select *Save Link As...* Save the file in your folder **MatlabFiles**.

Diary File: In every MATLAB assignment, you are required to record your work by creating a text file called a *diary file*. In the *Command Window*, type

```
diary labdemo.txt
```

followed by the **Enter** key.

When you are done working on the assignment, enter the **diary off** command, and it will write all the commands and their output to the `labdemo.txt` file. (See “Ending the session” below.) You can then edit this file using your favorite text editor.

Important note: Until you enter **diary off**, the commands that you have entered, and their output, are not written into the diary file. But MATLAB does in fact record all of your input and output in your workspace. **Be sure to save your workspace regularly, by clicking *Save Workspace in the toolbar*.**

If you stop your MATLAB session before completing a lab assignment, it is crucial for you to save your workspace. After you open your saved workspace, you can restart the diary file with the command `diary labdemo.txt`. The results of your new MATLAB session will be written at the end of the old diary file `labdemo.txt`. You may prefer to use different names (such as `labdemo-1.txt`, `labdemo-2.txt`) for each session on an assignment, and then merge the files when you prepare your lab write-up with a text editor. Of course, for the other labs you will change the filename to `lab1.txt`, and so forth.

Lab Write-up: Now that your diary file is open, enter the command

```
format compact
```

so that when you print out your diary file it will not have unnecessary spaces. Comments can be inserted into MATLAB by typing a percentage sign `%` in front of your comment. For instance, you will start off every MATLAB assignment by entering your name, last four digits of your Rutgers ID number, your section number, and the assignment title into MATLAB as comments. Enter the following comments into MATLAB, filling in your own information where appropriate.

```
% John Doe
% 1234
% Section C2
% Math 250 MATLAB Lab Demo
```

You will also use comments to indicate the start of a new problem, and to answer short-response questions. It is important for you to answer all the questions in each lab assignment. We will indicate where to insert these comments in the tutorial below.

General guidelines: The MATLAB assignments are intended to be interactive exercises. You need to follow the instructions precisely, and carefully inspect the output of each command. Think about what the result means. In your diary file, the outputs of your commands must (and automatically will) immediately follow the commands themselves.

Random Seed: When you start your MATLAB session, initialize the random number generator by typing

```
rand('seed', abcd)
```

where *abcd* are the last four digits of your Student ID number. **You must use the last four digits of your Student ID number, or you will not receive full credit (in Labs 1 – 6)!** This will ensure that you generate your own particular random vectors and matrices.

BE SURE TO INCLUDE THIS LINE IN YOUR LAB WRITE-UP

Note: If you stop your MATLAB session before completing a lab assignment, your random seed will be saved when you save your workspace (see above). When you later continue with the assignment, using your saved workspace, do not re-enter the random seed command.

IMPORTANT: You may want to use copy/paste to transfer commands from the MATLAB assignments to your MATLAB workspace, especially when the commands are relatively long. However, sometimes a command in an assignment might have symbols such as apostrophes, as in the random seed command above, that don't copy/paste accurately, and when this happens, you'll need to adjust the copy/pasted command in your workspace. Of course, if you use copy/paste for the random seed command above, you need to use your last four digits, in addition to fixing the apostrophes if necessary. In general, if you use copy/paste to transfer commands from assignments to your workspace, make sure that you read each command and understand its meaning. Some commands will invoke various technical MATLAB features.

The lab report that you hand in must be your own work. The following problems use randomly generated matrices and vectors, so the matrices and vectors in your lab report will not be the same as those of other students doing the lab. Sharing of lab report files is not allowed in this course.

10

Question 1. Matrices and Vectors

(3)

(a) In the *Command Window*, type

```
% Question 1(a)
```

and press ENTER. This comment will be saved in your diary file, and will indicate to you which problem the work which follows is for.

You can create a matrix in MATLAB by typing the entries in the matrix between square brackets, one row at a time. To separate the entries in the same row, either type a comma or press the space bar. To indicate the beginning of a new row, either type a semicolon or press the **Enter** key. Try this by typing

```
A = [0 1 2 ; 3 4 5]
```

(followed by **Enter**). MATLAB should then display the 2×3 matrix

$$A = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}$$

(MATLAB displays matrices without braces). As mentioned above, you could also generate this matrix by pressing the **Enter** key at the end of each row, instead of typing a semicolon.

Now use MATLAB to create the column vector

$$a = \begin{bmatrix} 9 \\ -1 \end{bmatrix}$$

Next, enter in the names of each of these matrices and vectors that you have created at the MATLAB prompt. Note that A and a are different objects; MATLAB is *case sensitive*. Finally, enter in **whos** at the prompt to get a list of all the matrices and vectors that are in your current MATLAB workspace.

(1)

(b) Now enter the following comment:

```
% Question 1(b)
```

In MATLAB, the entry in the i th row and j th column of a matrix A is obtained by the expression $A(i,j)$. Type `a12 = A(1,2)` into the command window, and check that `a12` is the entry of A in the 1st row and 2nd column.

(1)

(c) Add a comment to indicate the start of Question 1(c). A diagonal matrix is an $n \times n$ matrix B for which the entry in the i th row and j th column is zero whenever $i \neq j$.

Is the matrix A a diagonal matrix? Explain.

Here you are to record your answer to this question by typing it into MATLAB as a comment.

Here is one way to answer the question:

```
% The matrix A is not a square matrix, and therefore
% not a diagonal matrix.
```

If at any point in the lab, you aren't sure how to answer a question which requires an explanation, that's OK. Once you are all finished with your lab, you can type the required explanation directly into the diary file `labdemo.txt`.

- (1) (d) *Tcode* files give you new commands to use in MATLAB. For instance, `rdiag.m` gives you a new command to produce a random $n \times n$ diagonal matrix with integer entries between 0 and 9. Type

```
D = rdiag(3)
```

at the MATLAB prompt, and then press **Enter**. A 3×3 matrix will be returned. Is the matrix `D` a diagonal matrix? Explain.

- (2) (e) In the next part of the problem, you will need to explain why the following statement is true:

If B is a diagonal matrix and c is any scalar, then cB is a diagonal matrix.

We can test if this statement is true by using MATLAB to check some special cases. After seeing what happens in enough examples, we will hopefully see why the statement is true **in general**.

- Using MATLAB, calculate $23D$, and determine if it is a diagonal matrix.
- Using MATLAB, calculate $0D$, and determine if it is a diagonal matrix.
- Using MATLAB, calculate $-17D$, and determine if it is a diagonal matrix.
- Using MATLAB, calculate $\frac{1}{10}D$, and determine if it is a diagonal matrix.

Important note: MATLAB requires that you explicitly indicate multiplications using `*`, e.g., `23*D`.

- (2) (f) Explain why the following statement is true **in general**:

If B is a diagonal matrix and c is any scalar, then cB is a diagonal matrix.

Here is one way to answer this question:

```
% If B is a diagonal matrix, then whenever i is different from j,
% B(i,j)=0. Let c be a scalar. To see that cB is a diagonal matrix,
% we must check that cB(i,j) = 0 whenever i is different from j. The
% (i,j) entry of cB is given by c*B(i,j). Hence if i is different
% from j, cB(i,j) = c*B(i,j) = c*0 = 0. Therefore c*B is also a
% diagonal matrix.
```

Note that our calculations from Question 1(e) are insufficient to answer this question. We must explain why it is true that for **any** diagonal matrix B and **any** scalar c that the matrix cB is still diagonal. We cannot check all possible cases, so we need to use the algebra of matrices to explain why the statement is true in general.

As mentioned above, comments like the one given in Questions 1(c) and 1(f) may be entered directly into the diary file later, using a text editor.

Ending the session We are now done with the MATLAB demo, and will stop the diary file by entering the following command:

```
diary off
```

You may now close out of MATLAB.

Editing your diary file to produce a lab write-up: Use your favorite text editor, like Notepad, TextEdit, Emacs, Gedit, or Leafpad to open up your diary file `labdemo.txt`. On `apps.rutgers.edu`, the default text editor is Leafpad. You can open this program by navigating to `Menu > Accessories > Leafpad`. Open up the program Leafpad, and open your diary file `labdemo.txt`.

- Remove any unnecessary blank lines in your diary file. Assuming that you ran `format compact`, as instructed above, you will not have many lines to remove.
- If you accidentally entered in a wrong command, remove it from your diary file. For example, enter the following wrong command, and then remove it, along with the error message that it generates, from your diary file:

```
A = [1 2 ; 3]
{Error using <a href="matlab:matlab.internal.language.introspective.
errorDocCallback('vertcat')" style="font-weight:bold">vertcat</a>
Dimensions of matrices being concatenated are not consistent.}
```

- Remove the commands `load`, `save`, `clear`, `format`, `help`, `diary`, **with the exception of the “format compact” command that was entered at the beginning**. Remove any output from the commands `load`, `save`, `clear`, `format`, `help`, `diary`, as well.
- Ensure that you have answered all the questions, and have labeled all questions appropriately with comments of the following form:

```
% Question 1(d)
```

- Correct any typing errors.
- Save the file as a plain text file. Leafpad will do this by default, but Notes and Microsoft Word will not!
- Here is an important piece of editing advice that you should keep in mind for all of your MATLAB write-ups: Make sure that each new line and comment in your diary file is aligned to the left. You can check if the document is aligned correctly by right clicking the file and opening it with your browser (Internet Explorer, Chrome, Firefox, etc.).

Final editing of lab write-up: *Remove all typing errors from your lab write-up. Your write-up must contain only the input commands that you typed which were required by the assignment (including `format compact` at the beginning), the output results generated by MATLAB, immediately following the corresponding input commands, your answers to the questions in the indicated places, and the indicated comments such as question numbers.*

Lab write-up submission guidelines: This MATLAB Demo assignment will not be collected for grading, but here is a preview of instructions that will appear in Labs 1 – 6: Preview the document before printing and remove unnecessary page breaks and blank space. *Put your name and last four digits of your student ID number on each page, in case your pages become separated.* (If you have difficulty doing this using your text editor, you can write this information by hand after printing the report.) You must print your lab write-up. **Staple the pages together.** *Electronic submissions will not be accepted. Late submissions will not be accepted.*

Important: The submission of an unedited diary file without comments will be penalized by the removal of a significant number of points from the score.

Check your own write-up: A sample MATLAB write-up for this assignment is available on the course webpage. Carefully check your work and formatting against this version. Your diary file must look essentially like this demo writeup. If it does not, be sure to correct your work before starting Lab 1.