Revised 5/25/00

LAB 1: Introduction to MATLAB

In this lab you will learn how to use MATLAB to create and operate on matrices and vectors. The commands needed to do this are short and easy to remember, because MATLAB specializes in matrix computations and uses standard linear algebra notation.

Reading from Textbook: Before beginning the Lab, read through Sections 1.1, 1.2, 2.1 and 2.4 of the text and work the suggested problems for each section.

Demonstration of MATLAB: The basic mode of MATLAB is interactive. After you start the MATLAB program and obtain the prompt (`>>` or `EDU>>`), you type commands that MATLAB then executes when you press the `Enter` key. If you have never used MATLAB before, we suggest you type `demo` at the MATLAB prompt now. Use the mouse to move the selection bars to `Matrices` and `Basic matrix operations` on the menus that are displayed, and then click on `Run basic matrix` to get a slide show that demonstrates some of MATLAB's power. Also run the demonstration `Matrix manipulation`.

Diary File: Now that you have seen a demonstration of MATLAB, you can begin to work through this assignment. You will need to record the results of your MATLAB session to generate your lab report. Put a formatted floppy disk in the computer and type

```
diary a:\lab1.doc
```

followed by the `Enter` key (this assumes that the floppy drive in your computer is called drive `a`; change this as needed). Now each computation you make in MATLAB will be saved on your floppy disk in a text file named `lab1.doc`. At any point in your session you may turn off the recording by typing `diary off` (*don't do this now*).

If you want to stop your MATLAB session before completing a lab assignment, you can reopen the diary file the next time you start MATLAB. If you use the same file name, the results of your new MATLAB session will be written at the end of the old diary file. You may prefer to use different names (such as `lab1b.doc, lab1c.doc`) for each session on an assignment. Of course, for the other labs you will change the filename to `lab2.doc`, and so forth.

Lab Write-up: With your diary file open, type the comment line

```
% Math 250 MATLAB Lab Assignment #1
```

at the MATLAB prompt. Type `format compact` so that your diary file will not have unnecessary spaces. Put labels to mark the beginning of your work on each part of each question. For example,

```
% Question 1 (a) ...
      ⋮
% Question 1 (b) ...
```

and so on.

Be sure to answer all the questions in the lab assignment. Insert comments in your diary file as you work through the assignment. After you have worked through all the parts of a lab, you will need to edit your diary file. Remove all errors and other material that is not directly related to the questions.

Preview the document before printing and remove unnecessary page breaks and blank space. Put your name, section number, and student ID number on each page. (If you have difficulty doing this using your text editor, you can write this information by hand after printing the report.)

Important: An unedited diary file without comments will get a GRADE OF ZERO as a lab writeup.

Question 1. Creating Matrices and Vectors

The most direct way to create a matrix in MATLAB is to type the entries in the matrix between square brackets, one row at a time. To separate the entries in the same row, type a comma or press the space bar. To indicate the beginning of a new row, type a semicolon or press the `Enter` key. Try this by typing

```
A = [1 2; 3 4; 5 6]
```

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

$$A = \begin{matrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{matrix}$$

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

(a) Use MATLAB to create the following matrix, row vector and column vector:

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad x = \begin{bmatrix} 4 & 3 & 2 \end{bmatrix} \qquad X = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

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

(b) The MATLAB `size` command determines the number of rows and columns in a matrix. Every MATLAB command is documented in a `help` file, which you can access during a MATLAB session. Type `help size` now to get information about this command. Then use the `size` command to create a 4×2 matrix whose rows are the sizes of A, B, X, x (in that order), by typing

```
[size(A); size(B); size(X); size(x)]
```

Give this matrix the name S by typing `S = ans`. Note that all matrices which occur in MATLAB must have names; if a matrix is unnamed, then it gets assigned the temporary name of `ans` at the moment that it is created.

(c) To access a given entry in a matrix, put the row and column number in parentheses following the matrix name. Type `a32 = A(3,2)` and check that $a32$ is the $(3,2)$ entry in A defined above. Observe that the equal sign = in MATLAB (as in other programming languages) executes a *substitution* : the current value of the variable on the right side of the equal sign is placed into the location whose name is on the left side. Type `A(3,2) = 7` and check that the $(3,2)$ entry of A is now 7. Now change the $(3,2)$ entry of A back to 6. One way to do this is to type `A(3,2) = 6`. Another way that you should try for future use is the *up-arrow* key $\uparrow$. This lets you cycle through the commands that you have already typed. When you get to the command that generated A, press `Enter`. If you go too far with the up-arrow, you can use the down-arrow key $\downarrow$.

(d) To access a whole row or column of a matrix, use the colon operator. For example, $A(:,2)$ is the second column of A, while $B(1,:)$ is the first row of B. Type `C(:,1) = B(:,1), C(:,2)= B(:,3)` to create a 3×2 matrix C whose first column is the first column of B and whose second column is the third column of B. Then use the colon operator to create a 2×3 matrix D whose first row is the first row of B and whose second row is the third row of B. Use MATLAB to display the matrices C and D by typing

```
C, D   <Enter> .
```

More about MATLAB:

Editing: Because MATLAB is a command-line program, you cannot use the mouse to do on-screen editing, as you can in MAPLE. Instead, you must type some MATLAB command that causes the quantity to be changed as needed. This also means that you must edit your diary file after your MATLAB session to remove unnecessary or incorrect lines. You can use the mouse in the editing state, of course. You can also use the mouse to position the cursor before typing MATLAB commands, and you can use the right scroll bar to see the results of your current MATLAB session.

Saving Workspace: At this point your MATLAB workspace contains several matrices. It is a good idea to save the workspace every few minutes, in case your session is accidentally terminated or if you want to close MATLAB and resume work later. To do this, first choose **Set Path** from the **File** menu to set the path to drive `a:` (change the drive or subdirectory name as needed). Then type

```
save lab01
```

This will create a binary file on your diskette with the name `lab01.mat` that contains the data in your current workspace. You can restore the workspace by typing

```
load lab01
```

Try saving your workspace to diskette now. Then type `clear` to erase your workspace. Be careful; MATLAB does not give a warning message when the `clear` command is used, so you will lose your work if the `save` command was not executed. Now type `A` (this will give an error message). Then type `load lab01` and type `whos` to check that your workspace

has been restored. In working subsequent labs, you will change the filename as needed, of course.

Important: When you edit your lab write-up, remove from your diary file any of the results generated by the `load, save, clear` commands that you have just been trying. Your write-up should only contain the answers to the questions.

Question 2. Block Matrices and Special matrices

With MATLAB it is easy to form new matrices from those that are already in the workspace and to create matrices of special types.

(a) You can create *block matrices* by putting two matrices side by side (if they have the same number of rows), or one on top of the other (if they have the same number of columns). Use the matrices A, B, C, D, X created in Question 1 to make the following block matrices (the semicolon means that the matrices are stacked one on top of the other). Before typing the MATLAB commands, insert a comment line that lists which of the matrix combinations in this list fit together. Then use MATLAB (you will get error messages when the matrix sizes are not compatible).

$$[A\ X] \qquad [B\ C] \qquad [C\ D] \qquad [C; B] \qquad [B; D]$$

(b) Type each of the following commands that generate special matrices.

```
eye(4)   zeros(3)   zeros(3,5)   ones(2,3)
diag([4  5  6  7])   diag([4  5  6  7 ], -1)   diag([4  5  6  7 ], 2)
C   diag(C)   diag(diag(C))
```

Insert a comment that describes what the parameters -1 and 2 do in the second and third matrices. Insert a comment that describes how the matrices `diag(C)` and `diag(diag(C))` are formed from the elements of `C`. Then use these commands to create the following matrices:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 5 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 3 \end{bmatrix} \qquad \begin{bmatrix} 0 & 5 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{bmatrix}$$

(c) You can use the colon operator to create vectors and matrices with evenly spaced entries. Try typing `E = [1:5; 2:6]` and `F = [5:-1:1; 2:.5:4]` to see what you get. Then use the colon operator to create the following matrix.

$$G = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 7 & 5 & 3 & 1 \end{bmatrix}$$

(d) The MATLAB command `rand(m, n)` creates and $m \times n$ matrix whose entries are random numbers between 0 and 1. Try this by typing `R = rand(2, 3)`. Then use the up-arrow key to generate two more samples of the random matrix R. Notice how the entries in R change each time the command is executed.

More about MATLAB:

Suppressing Displays: When you place a semicolon at the end of a command, the command will be executed but the result will not be displayed on the screen. This is very useful when you are creating big matrices. Try typing `z = 1:.2:3;` to create a long row vector (note the semicolon at the end). Then type `z` to display the vector `z`.

If a command statement does not fit on one line, use three periods ... followed by the `Enter` key to indicate that the statement continues on the next line. Try this with

$$\begin{aligned} S \;\; = \;\; & 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 \; \ldots \\ & -1/8 + 1/9 - 1/10 + 1/11 - 1/12; \end{aligned}$$

(be sure to insert a space before ...; note the semicolon at the end of the second line). Then type `S` to display the numerical value of S (a very slowly-converging approximation to $\ln 2$).

Format Commands: You can control how numbers are displayed on the screen (this does not affect the internal arithmetic). Type

```
y = [4/3  1.2345e-6];
```

Then observe what display you get when you type the following.

```
format short, y
format short e, y
format long, y
format long e, y
format rat, y
```

Important: When you edit your lab write-up, remove from your diary file any of the results generated using the `format` and other MATLAB commands that you have just been trying.

Question 3. Matrix Addition and Multiplication

(a) To obtain a linear combination $aU + bV$ of the matrices U and V (where a and b are real numbers) using MATLAB, you must type `a*U + b*V`. This is only defined when U and V are the same size. Calculate the following using MATLAB.

$$A + C, \quad C + A, \quad 6C, \quad 2(3C), \quad 6A + 15C, \quad 3(2A + 5C)$$

Insert comment lines that explain the properties of matrix addition and scalar multiplication that these calculations illustrate.

(b) To obtain the product UV of the matrices U and V using MATLAB, you must type `U*V`. Remember that the product is only defined when the number of columns of U is the same as the number of rows of V. Also $UV \neq VU$ in general, even when both products are defined. Before typing the MATLAB commands, insert a comment line that lists which of the matrix products in the following list are defined, and the size of the product matrix. Then use MATLAB to calculate the products (you will get error messages when the matrix sizes are not compatible).:

$$AB, \quad BA, \quad AX, \quad XA, \quad BX, \quad XB, \quad Ax, \quad xA, \quad xB, \quad Bx, \quad Xx, \quad xX$$

(c) Calculate the following products using MATLAB.

$$H = BC, \quad J = HD, \quad K = CD, \quad L = BK$$

Note that $J = L$. Insert a comment line to explain why the properties of matrix multiplication predict this.

(d) The powers $B^2, B^3, \ldots$ of a square matrix B are obtained in MATLAB by typing

```
B^2,   B^3, ....
```

Calculate $M = B^2$, $N = B^3$, $P = M^3$, and $Q = N^2$. Insert a comment line to explain (using the properties of matrix multiplication) why $P - Q = 0$. Then use MATLAB to check this.

Question 4. Symbolic Matrix Calculations

Important: This question requires the MATLAB symbolic toolbox.

MATLAB can deal with matrices whose entries are *symbols* (ordinary variables as in calculus, such as x, y, z, or numbers such as $1/3$, π and $\sqrt{2}$ that need an infinite number of decimals to be expressed exactly). This is done using the Symbolic Toolbox (based on MAPLE). To get a quick online introduction to the Symbolic Math Toolbox, first save your current workspace (this is important, otherwise your work will be lost at the end of the demo). Also, close your diary file. Then type `demos` at the MATLAB command line. In the dialog box that is displayed, select `Symbolic Math` (in the left list box) and then `Introduction` (in the right list box). At the end of the demo, you will see `echo off`. At this point, quit MATLAB and then restart it and reopen your diary file and reload your workspace.

(a) Generate a symbolic diagonal matrix by

```
syms a  b  c
Ds = diag([a b c])
```

Then use MATLAB to calculate $(Ds)^2$ and $(Ds)^3$. Insert a comment to explain why the results are predicted by the way diagonal matrices multiply.

(b) Generate the 4×4 *Hilbert matrix*

```
H = hilb(4)
```

Then make H into a symbolic matrix by the command `Hs = sym(H)`. Notice that the entries in Hs are rational numbers that follow a simple pattern.

By the rules of matrix algebra, what should you get for the product $H^5 * H^{-5}$? Try to use MATLAB to calculate this product. What happens? Now use MATLAB to calculate the product of the symbolic matrices $(Hs)^5 * (Hs)^{-5}$. Do you get the correct answer?