Revised 9/12/01

LAB 1: Matrix Computations in 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 1.2, 1.3 and 1.4 of the textbook and work the assigned homework exercises for these sections. You may find it helpful to consult Sections 12.1, 12.2 and 12.3 of the text.

Demonstration of MATLAB: The basic mode of MATLAB is interactive. After you start the MATLAB program and obtain the prompt `>>`, 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. Click on `Desktop Envirnment` and run the playback files (you may want to return to this demo later as you work through the lab and prepare your writeup). Then move on to the demo of `Matrices`. and Run `Basic matrix operations`. This gives a slide show that demonstrates how matrices are entered and displayed. 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:\lab01.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 `lab01b.doc, lab01c.doc`) for each session on an assignment. Of course, for the other labs you will change the filename to `lab02.doc`, and so forth.

Lab Write-up: Now that your diary file is 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.

Optional Method to Generate Lab Report: In the latest version of MATLAB, you can open a Microsoft WORD document and then insert MATLAB commands which are executed within the document. If you have used MAPLE then you are already familiar with this environment. Go to the the MATLAB help files for more information if you want to use this option. For the purposes of the lab writeups in this course, the simpler method of using a `diary` file as described above is recommended.

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

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

(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> .
```

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 *without typing any zeros:*

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

(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 = [3:0.2:5];` to create a long row vector with equally spaced entries(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

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

(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
```

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. Matrix Product in terms of Row Vectors and Column Vectors

Follow the instructions for MATLAB Exercise **ML.4** on page 32 of the text. Use MATLAB to carry out the calculations for parts **(a)**, **(b)**, and **(c)** of this exercise. Be sure to answer (in words) the questions in each part also.

Optional Extra-Credit Applied Problem

Read **Section 2.1 Graph Theory** in the text. The following questions refer to the communications network in **Exercises 2.1**, problem #6 on page 102 of the text.

(a) Draw (by hand) a digraph G that describes the communication network. Label the vertices as P_1, P_2, P_3, P_4, P_5.

(b) Determine how many different **two-step** communication paths exist from P_1 to P_4. Do this in two ways: First by using the digraph G, and then by using the matrix adjacency matrix A.

(c) Are there any **three-step** communication paths from P_3 to P_4? Answer this in two ways: First by using the digraph G, and then by using MATLAB to calculate an appropriate power of A.

(d) Find the **smallest** integer k so that every pair of individuals is connected by a communication path of length **exactly** k steps. Explain how you get your answer. (*Hint:* Use MATLAB to calculate powers of A.)

Final Editing of Lab Write-up: After you have worked through all the parts of the lab assignment, you will need to edit your diary file. Remove all errors and other material that is not directly related to the questions. In particular, remove from your diary file any of the results generated by the `load, save, clear, format` commands that you used. Your write-up should only contain the answers 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 submitted as a lab writeup will get a GRADE OF ZERO.