Revised 1/10/07

LAB 1: Matrix and Vector Computations in MATLAB

In this lab you will use MATLAB to study the following topics:

- How to create matrices and vectors in MATLAB.
- How to manipulate matrices in MATLAB and create matrices of special types.
- How to add matrices and multiply a vector by a matrix by MATLAB.
- How to find the reduced row echelon form of a matrix by MATLAB.
- The use of matrices in the Leontief input-output economic model.

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

Reading from Textbook: Before beginning the Lab, read through Sections 1.1, 1.2, 1.3 and 1.4 of the textbook and work the assigned homework exercises for these sections to get familiar with matrices and vectors.

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 Environment` 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. Create a directory (folder) on your computer disk to save your MATLAB work in. Then use the `Current Directory` field in the desktop toolbar to change the directory to this folder. Now type

```
diary lab1.txt
```

followed by the `Enter` key. Now each computation you make in MATLAB will be saved in your directory in a text file named `lab1.txt`. You can then edit this file using your favorite text editor. When you have finished your MATLAB session you can turn off the recording by typing `diary off` at the MATLAB prompt (*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 `lab1a.txt, lab1b.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 `lab2.txt`, 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, so that your edited lab write-up has the format

```
% 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.

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. This will ensure that you generate your own particular random vectors and matrices.

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

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.

Question 1. Creating Matrices and Vectors

(a) 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.

Now 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 that are in your 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 ↑. 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 ↓.

(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])
```

(c) The MATLAB command `rand(m, n)` creates an $m \times n$ 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`.

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 the values you get for y when you type the following:

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

Before continuing with the lab, type `format short`

Question 3. Matrix Addition and Matrix-Vector Multiplication

Be sure that you have set the seed of the random number generator as described at the beginning of this assignment. Generate vectors **u** and **v** and matrices A and B by typing

u = fix(10*rand(3,1)), v = fix(10*rand(3,1)), A = fix(10*rand(2,3)), B = fix(10*rand(2,3))

These matrices have entries randomly selected from the numbers $0, 1, \ldots, 9$.

(a) To obtain a linear combination $sA + tB$ of the matrices A and B (where s and t are real numbers) using MATLAB, you must type `s*A + t*B`. This is only defined when A and B are the same size. The transpose A^T of A is obtained by typing A'. Calculate the following using MATLAB.

$$A + B, \quad B + A, \quad 6B, \quad 2(3B), \quad 6A + 15B, \quad 3(2A + 5B), \quad 3A, \quad (3A^T)^T$$

Insert comment lines in your diary file that explain the properties of matrix addition and scalar multiplication that these calculations illustrate (see Theorem 1.1 on page 5 of the text).

(b) To obtain the matrix-vector product $A\mathbf{u}$ using MATLAB, you must type `A*u` (remember that a vector is just a matrix with one column). Calculate the following using MATLAB.

$$A\mathbf{u} + A\mathbf{v}, \quad A(\mathbf{u} + \mathbf{v}), \quad (A + B)\mathbf{u}, \quad A\mathbf{u} + B\mathbf{u}, \quad A(3\mathbf{u}), \quad 3A(\mathbf{u})$$

Insert comment lines that explain the properties of matrix-vector products that these calculations illustrate (see Theorem 1.2 on page 20 of the text).

Question 4. Gaussian Elimination and Reduced Row-Echelon Form

Now that you know how to do matrix calculations with MATLAB, you can easily carry out the steps in Gaussian elimination. Before starting work on this question, type `rrefmovie` at the MATLAB prompt. You will see a step-by-step example of the row operations that transform a matrix `A` into its reduced row echelon form `R = rref(A)`. In this demonstration, each pivot is chosen to be the *largest* in its column (for numerical stability), so extra row interchanges are used. Since `rref(A)` is *uniquely* determined by A, this does not affect the final answer. (Do not include this part in your lab write-up.)

(a) Generate a 3×4 matrix A with random integer entries by the command

A = fix(10*rand(3,4))

To transform A into $R =$ rref(A), start with $R = A$. Normalize the first row of R to get $R(1, 1) = 1$:

R = A; R(1,:) = R(1,:)/R(1,1)

(note the use of the colon to operate on whole rows of the matrix). If your random matrix happens to have $A(1, 1) = 0$, then interchange rows to get a nonzero entry in the $(1, 1)$ position before doing the calculation above. Now subtract a multiple of the first row of R from the second row to make $R(2, 1) = 0$:

R(2,:) = R(2,:) - R(2,1)*R(1,:)

Repeat this procedure to make $R(3, 1) = 0$:

R(3,:) = R(3,:) - R(3,1)*R(1,:)

(you can minimize typing by using the up-arrow key and editing the command line). The first column of your matrix R should now be the same as the first column of ref(A).

(b) Operate on R by the same method as in **(a)** to obtain the second column of rref(A). First normalize row 2 of R, then subtract multiples of row 2 from rows 1 and 3 to put zeros in the $(1, 2)$ and $(3, 2)$ positions (you can minimize typing by using the up-arrow key and editing the commands used in part **(a)**).

(c) Operate on R by the same method as in **(b)** to obtain the third column of `rref`(A). First normalize row 3 of R, then subtract multiples of row 3 from rows 1 and 2 to put zeros in the $(1, 3)$ and $(2, 3)$ positions.

(d) Your matrix R should now be transformed into `rref(A)` (since A is a random 3×4 matrix, `rref(A)` is (almost) sure to have rank 3). Check your answer by MATLAB with the command `rref(A)`. If the MATLAB answer is not the same as the current value of your R, go back and redo your calculations.

Question 5. Leontief Input-Output Model

Read pages 50-53 of **Section 1.5** in the text and study Practice Problem 1 on page 56 and its solution on page 59.

(a) Use MATLAB to create the input-output matrix C for the economy describe in Exercise 16 on page 57, as follows: Create **column** vectors $\mathbf{a}$, $\mathbf{m}$, and $\mathbf{s}$ that correspond to the agricultural, manufacturing, and services sectors, respectively. Then $C = [\mathbf{a}, \quad \mathbf{m}, \quad \mathbf{s}]$. Now use the *gross production vector* $\mathbf{x}$ in part **(a)** of Exercise 16 and the matrix C to calculate the *net production vector*.

(b) Find the gross production vector $\mathbf{x}$ that satisfies the conditions in part (b) of Exercise 16. This requires solving an inhomogeneous system of equations with the demand vector $\mathbf{d}$ on the right-hand side. Use the MATLAB `rref` command on a suitable augmented matrix to carry out the calculations.

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 results of the MATLAB calculations and the answers to the questions that you have written. 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 is not acceptable.