MAT 343 Laboratory 1
Matrix and Vector Computations in MATLAB

MATLAB is a computer software commonly used in both education and industry to solve a wide range of problems.

This Laboratory provides a brief introduction to MATLAB, and the tools and functions that help you to work with MATLAB variables and files.

More specifically, we will learn how to

1. Create matrices and vectors.
2. Manipulate matrices and create matrices of special types
3. Add and multiply matrices

Preliminaries

The MATLAB Environment

To start MATLAB double-click on the MATLAB shortcut icon. The MATLAB desktop will open. The MATLAB default desktop consists of three windows: the command window, the Current Folder browser and the Workspace browser.

- The *command window* is where MATLAB commands are entered and executed.
- The *Current Folder* browser allows you to view MATLAB and other files and to perform file operations such as opening and editing or searching for files.
- The *Workspace* browser allows you to view and make changes to the contents of the workspace.

Note that windows within the MATLAB desktop can be resized by dragging the separator bar(s) and they can be closed by clicking on the × in the upper right corner of the window.

Basics And Help

Commands are entered in the Command Window.

★ Basic operations are +, -, *, and /. The sequence

```
>> a=2; b=3; a+b, a*b
ans =
     5
ans =
     6
```

defines variables a and b and assigns values 2 and 3, respectively, then computes the sum $a+b$ and product ab. Each command ends with `,` (output is visible) or `;` (output is suppressed). The last command on a line does not require a `,`.

★ Standard functions can be invoked using their usual mathematical notations. For example

```
>> theta=pi/5;
>> cos(theta)^2+sin(theta)^2
ans =
     1
```

verifies the trigonometric identity $\sin^2\theta + \cos^2\theta = 1$ for $\theta = \frac{\pi}{5}$. A list of elementary math functions can be obtained by typing

```
  >> help elfun
```

★ To obtain a description of the use of a particular function type help followed by the name of the function. For example

```
    >> help cosh
```

gives help on the hyperbolic cosine function.
★ To get a list of other groups of MATLAB programs already available enter `help`:

```
    >> help
```

★ If you are looking for a function, use `lookfor` *keyword* to get a list of functions with the string keyword in them. For example, typing `lookfor 'identity matrix'` lists functions (there are two of them) that create identity matrices.
★ Another way to obtain help is through the help button ? in the toolbar.

★ MATLAB is case-sensitive. For example

```
>> theta=1e-3, Theta=2e-5, ratio=theta/Theta
theta =
  1.0000e-003
Theta =
  2.0000e-005
ratio =
    50
```

★ The quantities `Inf` (∞) and `NaN` (Not a Number) also appear frequently. Compare

```
    >> c=1/0
    c =
       Inf
```

with

```
    >> d=0/0
    d =
       NaN
```

Matrices in MATLAB

Entering matrices in MATLAB is easy. For example, to enter the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

type `A=[1, 2, 3, 4; 5, 6, 7, 8; 9, 10, 11, 12; 13, 14, 15, 16]`
or the matrix could be entered one row at a time:

```
A=[ 1 2 3 4
    5 6 7 8
     9 10 11 12
     13 14 15 16]
```

Once a matrix has been entered, you can edit it. Here are some examples, which you can try in the command window:

Input	**Output**	

`A(1,3) = 5`	`A =`
	`1  2  5  4`
	`5  6  7  8`
	`9  10  11  12`
	`13  14  15  16`
		Changes the third entry in the first row of A to 5

`C = A(2:3,2:4)`	`C =`
	`6  7  8`
	`10  11  12`
		Submatrix consisting of the entries in rows 2 and 3 and columns 2 through 4

`A(:,2:3)`	`ans =`
	`2  3`
	`6  7`
	`10  11`
	`14  15`
		Submatrix of A consisting of all the elements in the second and third columns

`A(4,:)`	`ans =`
	`13  14  15  16`
		Fourth row of A

`E = A([1,3],[2,4])`	`E =`
	`2  4`
	`10  12`
		Matrix whose entries are those which appear only in the first and third rows and second and fourth column of A

Vectors

Vectors are special cases of matrices, with just one row or one column. They are entered the same way as a matrix. For example
`u = [1, 3, 9]` produces a row vector
`v = [1; 3; 9]` produces a column vector.
Row vectors of equally spaced points can be generated with MATLAB's `:` operation or using the `linspace` command.

Input	**Output**	

`x = 2:6`	`x =`
	`2  3  4  5  6`
		row vector with integer entries going from 2 to 6

`x = 1.2:0.2:2`	`x =`
	`1.2000  1.4000  1.6000  1.8000  2.0000`
		Row vector with stepsize 0.2

`x = linspace(1.2,2,5)`	`x =`
	`1.2000  1.4000  1.6000  1.8000  2.0000`
		Row vector with 5 equally spaced entries between 1.2 and 2

Generating Matrices

We can also generate matrices by using the built-in MATLAB functions. For example, the command

```
B=rand(4)
```

generates a 4×4 matrix whose entries are random numbers between 0 and 1. Here is a list of the most common built-in matrices:

`rand(m,n)`	m by n matrix with random numbers between 0 and 1
`eye(m,n)`	m by n matrix with 1's on the main diagonal
`zeros(m,n)`	m by n matrix of zeros
`ones(m,n)`	m by n matrix of ones
`triu(A)`	extracts the upper triangular part of the matrix A
`tril(A)`	extracts the lower triangular part of the matrix A
`diag(v,k)`	square matrix with the vector **v** on the kth diagonal

The first four commands above with a single argument, e.g. `ones(m)`, produce a square matrix of dimension m.

More special matrices:
There is also a set of built-in special matrices such as `magic`, `hilb`, `pascal`, `toeplitz`, and `vander`.

The matrix building commands can be used to generate block of partitioned matrices. Here is an example:

Input	**Output**

`E=[eye(2),ones(2,3);zeros(2),[1:3; 3:-1:1]]`	`E =`
	`1 0 1 1 1`
	`0 1 1 1 1`
	`0 0 1 2 3`
	`0 0 3 2 1`

Addition and Multiplication of Matrices

Matrix arithmetic in MATLAB is straightforward. We can multiply our original matrix A times B simply by typing `A*B`. The sum and difference of A and B are given by `A + B` and `A - B`, respectively. The transpose of the real matrix A is given by `A'`.

Exponentiation

Powers of matrices are easily generated. The matrix A^5 is computed in MATLAB by typing `A^5`. We can also perform operations element-wise by preceding the operand by a period.
For instance, if `V=[1,2; 3,4]`, then

Input	Output	

`V^2`	`ans =`
	`7 10`
	`15 22`
		Equivalent to `V*V`

`V.^2`	`ans =`
	`1 4`
	`9 16`
		component-wise exponentiation

Appending a row or a column

A row can be easily appended to an existing matrix provided the row has the same length of the rows of the existing matrix. The same thing goes for the columns. The command `A=[A, v]` appends the column vector **v** to the columns of A, while `A = [A; u]` appends the row vector **u** to the rows of A.

Examples: If $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $\mathbf{u} = \begin{bmatrix} 5 & 6 & 7 \end{bmatrix}$, and $\mathbf{v} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$, then

- `C = [A; u]` produces $C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 5 & 6 & 7 \end{bmatrix}$, a 4 × 3 matrix

- `D = [A, v]` produces $D = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 4 \end{bmatrix}$, a 3 × 4 matrix.

Deleting a row or column

Let `A=[1, 2, 3, 4, 5; 6, 7, 8, 9, 10; 11, 12, 13, 14, 15]`, then

Input	**Output**	

`A(2,:)  = []`	`A =`
	`1  2  3  4  5`
	`11  12  13  14  15`
		deletes the 2nd row of matrix A

`A(:,3:5) = []`	`A =`
	`1  2`
	`6  7`
	`11  12`
		deletes the 3rd through 5th columns of A

`A([1,3],:)  = []`	`A =`
	`6  7  8  9  10`
		deletes the 1st and 3rd row of A

Columnwise Array Operators

MATLAB has a number of functions that, when applied to either a row or column vector **x**, returns a single number. For example, the command `max(x)` will compute the maximum entry of **x** , and the command `sum(x)` will return the value of the sum of the entries of **x**. Other functions of this form are `min`, `prod`, `mean`. When used with a matrix argument, these functions are applied to each column vector and the results are returned as a row vector.

For example if $A = \begin{bmatrix} -3 & 2 & 5 & 4 \\ 1 & 3 & 8 & 0 \\ -6 & 3 & 1 & 3 \end{bmatrix}$, then

Input	**Output**	

`min(A)`	`ans =`
	`-6  2  1  0`
		minimum entry in each column of A

`max(A)`	`ans =`
	`1  3  8  4`
		maximum entry in each column of A

`sum(A)`	`ans =`
	`-8  8  14  7`
		sum of the entries in each column of A

`prod(A)`	`ans =`
	`18  18  40  0`
		product of the entries in each column of A

EXERCISES

Instructions

You will need to record the results of your MATLAB session to generate your lab report. Create a directory (folder) on your computer 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 save in your directory in a text file named lab1.txt. When you have finished your MATLAB session you can turn off the recording by typing `diary off` at the MATLAB prompt. You can then edit this file using your favorite text editor (e.g. MS Word).

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), and the comment line

```
% MAT 343 MATLAB Assignment # 1
```

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
        .
        .
% Question 2 (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 typing errors.
- Unless otherwise specified, your write-up should contain the MATLAB input commands, the corresponding output, and the answers to the questions that you have written.
- If the question asks you to write an M-file, copy and paste the file into your lab report in the appropriate position (after the problem number and before the output generated by the file).

- If the question asks for a graph, copy the figure and paste it into your lab report in the appropriate position. Crop and resize the figure so that it does not take too much space. Use ";" to suppress the output from the vectors used to generate the graph. Makes sure you use enough points for your graphs so that the resulting curves are nice and smooth.
- Clearly separate all questions. The questions numbers should be in a larger format and in **boldface**. Preview the document before submitting it and remove unnecessary page breaks and blank spaces.

Important: An unedited diary file without comments submitted as a lab writeup is not acceptable.

1. **Entering matrices:** Enter the following matrices:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 6 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & -3 \\ 1 & 4 \end{bmatrix}, \quad C = \begin{bmatrix} -2 & 1 \\ 3 & 2 \end{bmatrix}$$

2. **Check some linear algebra rules:**
   (a) **Is matrix addition commutative?** Compute $A + B$ and then $B + A$. Are the results the same?
   (b) **Is matrix addition associative?** Compute $(A + B) + C$ and $A + (B + C)$ in the order prescribed. Are the results the same?
   (c) **Is multiplication with a scalar distributive?** Compute $\alpha(A + B)$ and $\alpha A + \alpha B$, taking $\alpha = 6$ and show that the results are the same.
   (d) **Is multiplication with a matrix distributive?** Compute $A(B + C)$ and compare with $AB + AC$.
   (e) **Matrices are different from scalars!**
      (i) For scalars, $ab = ac$ implies that $b = c$ if $a \neq 0$. Is that true for matrices? Check by computing AB and AC for the matrices given above.
      (ii) In general, matrix products do not commute either (unlike scalar products). Check if AB and BA give different results.
   (f) Compare the result of `A*B` to:
      (i) `[A*B(:,1), A*B(:,2)]`
      (ii) `[A(1,:)*B; A(2,:)*B ]`
      (iii) Explain how the matrices in (i) and (ii) are generated.

3. **Create matrices with `zeros`, `eye`, `ones`, and `triu`:** Create the following matrices with the help of the matrix generation functions `zeros`, `eye` , `ones`, and `triu`. See the on-line help on these functions if required (i.e. `help eye`)

$$M = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad N = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad P = \begin{bmatrix} 3 & 3 & 3 \\ 0 & 3 & 3 \\ 0 & 0 & 3 \end{bmatrix}, \quad Q = \begin{bmatrix} 4 & 4 & 4 \\ 4 & 4 & 4 \end{bmatrix}.$$

4. **Create a big matrix with submatrices:** The following matrix G is created by putting matrices A, B, and C from Exercise 1, on its diagonal and inserting 2×2 zeros matrices and 2×2 identity matrices in the appropriate position. Create the matrix using submatrices A, B, C, `zeros` and `eye` (that is, you are not allowed to enter the numbers explicitly).

$$G = \begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 \\ 3 & 6 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & -3 & 1 & 0 \\ 0 & 0 & 1 & 4 & 0 & 1 \\ 1 & 0 & 0 & 0 & -2 & 1 \\ 0 & 1 & 0 & 0 & 3 & 2 \end{bmatrix}$$

5. **Manipulate a matrix:** Do the following operations on matrix G created above in Problem 4.
   (a) Extract the first 3×3 submatrix from G and store it in the matrix H, that is, create a matrix

   $$H = \begin{bmatrix} 1 & 2 & 1 \\ 3 & 6 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

   by extracting the appropriate rows and columns from the matrix G.

   (b) Create the matrix E obtained from H by replacing $E_{33} = 2$ by 5. Do not enter E explicitly. Hint: enter first `E=H;` to create copy of the matrix H and then manipulate the matrix E.

   (c) Create the matrix F obtained by deleting the second row of the matrix H. Do not enter F explicitly.

   (d) What happens if you type `G(:,:)` and hit return? Do not include the output in your lab report, but include a statement describing the output in words.
   What happens if you type `G(:)` and hit return? Do not include the output in your lab report, but include a statement describing the output in words.

   (e) What do you get if you type `G(8)` and hit return? Can you explain how MATLAB got that answer? Try `G(15)` to confirm your answer.

   (f) What happens if you type `G(10,1)` and hit return? Explain.

   (g) What happens if you type `G(G>2)` and hit return? Can you explain how MATLAB got that answer? What happens if you type `G(G>2) = 50` and hit return? Can you explain how MATLAB got that answer?

6. **See the structure of a matrix:** Create a 25×25 matrix with the command `A = ones(25);`
   Now replace the 5×5 submatrix between rows 11:15 and columns 11:15 with zeros. See the structure of the matrix (in terms of nonzero entries) with the command `spy(A)`.
   Set the 5×5 submatrices in the top left corner and bottom right corner to zeros and see the structure again.
   NOTE: Use semicolon to suppress the output for all the matrices in this problem. In your lab-write up include the pictures obtained with the `spy` command. To include the pictures, open your diary file using a word processor such as MS Word then, on the MATLAB figure, select "Edit" and "Copy Figure", and paste the picture into the Word file. Make sure you crop and resize the picture so that it does not take up too much space.

7. **Create a symmetric matrix:** Create an upper triangular matrix with the following command:

```
A = diag(1:6) + diag(7:11, 1) + diag(12:15, 2)
```

   Make sure you understand how this command works (see the on-line help on `diag`). Now use the upper off-diagonal terms of A to make A a symmetric matrix with the following command:

```
A = A + triu(A,1)'
```

   This command takes the upper triangular part of A above the main diagonal, flips it (transpose), and adds to the original matrix A, thus creating a symmetric matrix A. See the on-line help on `triu`.

8. **Perform row operations:** Enter the following matrix:

   $$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

   (a) Enter the following command in MATLAB:

```
A(2,:)=A(2,:)-4*A(1,:)
```

This command performs an elementary row operation on the matrix A. Describe the row operation.

(b) On the matrix obtained in part (a), use MATLAB to perform a row operation that subtracts 7 times the first row from the third row.

(c) On the matrix obtained in part (b), divide the second row by -3.