Lecture 3

Mr. Edem K. Bankas

Variables in m-file Environments

- A variable is a named storage location that can be set for a particular value which can be used in subsequent computations

- For example, we can store a value 12 in the variable x with the statement x = 12. This value remains in x until we store a different value (e.g., typing the command x = 200) or we clear x using the command clear x

Examples

- E.g.1

  Suppose we wish to compute the circumference of a circle of diameter 6 units using the formula $C = \pi d$. We could first of all set the variable *d* to 6, and so on .

Variables

- Variable names must begin with an upper –or- lower- case letter. They may contain letters, digits, and underscores; they may not contain spaces or punctuation characters
- Variable names are case sensitive. E.g., **A** and **a** are different variables

- Variables can contain several types of numerical values, namely:

Variables cont’

- Scalar - single value
- Vector – ordered series of numbers
- Matrices – a rectangular array of numbers
- Strings – Variables may contain string characters

Basics of files

- To save the expressions entered in the command window,
  - Click on file drop down menu
  - Select save workspace As…
  - Type in a file name
  - Click on save button

  This method creates a MATLAB file which has a .MAT file extension in windows

Basics of files cont’

- There are at times when long sequences of operations can be typed and stored in a file so as to execute with a single command in the command window

- This is done by creating a script file known as MATLAB program and is saved in a file format with a .m extension. This makes to be called m-files.

Example

- Create a script file that will compute e^x for a few values of x.

Solution

Open the MATLAB editor.

```
% script file example1.m to compute
%exponential of a set of numbers
x = [1:2:3:4];
y = exp(x)
```

Basics of files cont'

- We can use m-files to create and store data. For e.g.,

  Examscore = [64, 28, 18, 32, 16]

  We will store this as score.m

  We can then use this data by doing some manipulation e.g.,

  >> score

  SeventyPercent = 0.7*Examscores

Assignment 1

1. Use MATLAB to compute the sine of $\frac{\pi}{3}$ expressed as a rational number.

2. Create a MATLAB m-file to display the results of $Sin\left(\frac{\pi}{4}\right), Sin\left(\frac{\pi}{3}\right)$ and $Sin\left(\frac{\pi}{2}\right)$ as rational numbers.

Vectors

- MATLAB is exceptionally capable of handling array of numbers.

- A vector is a one dimensional array of numbers. With MATLAB, one can create column and row vectors

Column Vectors

- A column vector can be created by enclosing a set of semicolon delimited numbers in square brackets.

Example:

```
>> a = [3; 6; 2]

a =

     3

     6

     2
```

Column Vectors cont'

- Basic operations on column vectors can be executed by referencing the variable name used to create them

Scalar Multiplication

Given that >>a = [3; 6; 2];

>> b = 3

>> c = 3 * a

Row Vectors

- A row vector can be created by enclosing a set of numbers in square brackets by using space or comma to delimit the numbers.

- E.g., >> V = [6, 8, -2] or V = [6 8 -2]

Transpose Operation

- Column vectors can be turned into row vectors and vice versa

Definition:

Suppose that we have a column vector with **n** elements denoted by

$$V = \begin{pmatrix} v_1 \\ v_2 \\ . \\ . \\ . \\ v_n \end{pmatrix}$$

Then the transpose is given by

$$v^T = \begin{bmatrix} v_1 & v_2 & . & . & . & v_n \end{bmatrix}$$

Transpose Operation cont’

- In MATLAB, we represent the transpose operation with a single quote. Taking the transpose of a column vector produces a row vector

```
>>  a = [2; 6; 7];
>>  y = a’
```

Exercise

- Transpose a row vector to produce a column vector

```
>> Q = [12, 6, 5];
>> A = Q'
    A =
          12
           6
           5
```

Addition and Subtraction of Vectors

- In order to perform this operation, the vectors must both be of the same type and the same length, so we can add two column vectors to produce a new column vector or add two row vectors to produce a new row vector

Example 1

```
>>  A = [1; 4; 5];
>>  B = [2; 3; 3];
>>  C = A + B
     C =
           3
           7
           8
```

Example 2

```
>>  M = [2, 4, 7];
>>  N = [1, 2, 5];
>>  O = M - N
      O =
            1  2  2
```

Creating large Vectors from Existing Variables

- In MATLAB, we can append vectors together to create new ones.
- Let **u** and **v** be two column vectors with *m* and *n* elements respectively that have been created in MATLAB.
- A third vector **w** whose first elements are the elements of **u** and next *n* elements are the elements of **v**.
- This makes the newly created column vector has (*m* + *n*) elements

Syntax

- This is done by writing

  w = [u ; v]

Example 1:

```
>> A = [2; 4 ;6];
>> B = [6; 8; 5];
>> C = [A;B]
        C =
             2
             4
             6
             6
             8
             5
```

- Example 2

  The same can be done using row vectors

  P = [ ];

  Q = [ ];

  R = [ ]

  R =

  …….

Creating Vectors with uniform spaced elements

- To create a vector **x** with uniformly spaced elements, where x_i is the first element and x_n is the final element. The increment is q ($q \in \Re$)

  Syntax:

$$x = \left[x_i : q : x_n \right]$$

Example 1

- Create a list of even numbers from 0 to 10

Solution

```
>> x = [0 : 2 : 10]

    x =

        0  2  4  6  8  10
```

Example 2

1. What is the output of A = [0 : 0.1 : 1]
2. Create a list of numbers from 100 to 80 decreasing by 5

Characterizing a vector

- The length command returns the number of elements that a vector contains.

Example 1

```
>> A = [2 ; 3 ; 3 ; 4 ; 5];
>> length (A)
   ans =
            5
```

Characterizing a vector

- Example 2

```
>> B = [2; 4; 6];
>> length (B)
     ans =
          2
```

Characterizing a vector

- We can also find the largest and smallest elements in a vector using **max** and **min** commands

Example

```
>> A = [6 8 9 12 13 6 3 2 ];
>> max (A)
      .....
>> min (A)
```

Vector Dot and Cross Product

Definition

- The dot product of two vectors $A = (a_1, a_2, \ldots a_n)$ and $B = (b_1, b_2, \ldots b_n)$ is given by

$$A \cdot B = \sum_i^n a_i b_i$$

In MATLAB, the dot product of two vectors **a** and **b** can be calculated by using the **dot(a,b)** command

Dot Product

- Example1

```
>> a = [2, 3, 4];
>> b = [2, 6, 3];
>> c = dot(a,b)
        c =
             34
```

Cross Product

Example

```
>> A = [1, 2, 3];
>> B = [2, 3, 4];
>> C = cross(A, B)
    C =
```

Referencing Vector Components

- We can reference one or more components of a vector.
- The ith component of a vector can be referenced by writing v(i). E.g.,

```
>> A = [12; 8; 6; 0; 4; 6; 13; 28];
>> A (3)
     ans =
               6
```

Referencing Vector Components

- Referencing the vector with a colon, lists all the elements of the vector. E.g.,

  A (:)

To reference some components in a vector. E.g

v = A (2: 3)

Assignment 2

1. (a) Given that the vector **a** = [a_1 ; a_2; …a_n], Demonstrate in two different ways how the magnitude of **a** is computed in MATLAB

   (b) Hence, find the magnitude of a vector **a** = [2 ; 5; 7; 8]

2.