VARIABLES, VECTORS, MATRICES, AND ARRAYS

Variables and Arrays

- A **variable** is a symbolic name associated with a value.
- Variables can reference different data types
  - scalars, vectors, arrays, matrices, strings etc....
- **Arrays** are lists of numbers or expressions arranged in horizontal rows and vertical columns.
- A single row, or single column array is called a **vector**.

Arrays and Vectors

- An array with m rows and n columns is called a matrix of size m x n.

```
>> x = [1 2 3]
x =
    1 2 3
>> y = [4; 5; 6]
y =
    4
    5
    6
```

Square brackets [ ] are used to denote a vector or matrix.

Spaces are used to denote columns.

The semicolon (;) operator is used to separate columns

The transpose operator (')

```
>> x'
ans =
     1
     2
     3

>> y'
ans =
     4 5 6
```

Creating vectors using Range

- A **range** can be created using the **colon operator**
  - **8:1:10** means create a range that starts at 8 and goes up in steps of size 1 until 10.
- A **range** can also be created using the **linspace** function
  - **linspace(0,10,5)** means create a range between 0 and 10 with 5 linearly spaced elements.

```
>> z = 8:1:10
z =
    8 9 10
```

```
>> v = linspace(0,10,5)
v =
 0 2.5000 5.0000 7.5000 10.0000
```

The Dot Operator

```
>> clear
>> a = [2 3; 5 1]
a =
     2     3
     5     1

>> b = [4 7; 9 6]
b =
     4     7
     9     6
```

The Dot Operator

```
>> a*b
ans =
      35    32
      29    41
```

```
a =
     2     3
     5     1

b =
     4     7
     9     6
```

Where did those numbers come from?

The Dot Operator

- Now, try the **Dot operator**

```
>> a.*b
ans =
     8    21
    45     6
```

```
a =
     2     3
     5     1

b =
     4     7
     9     6
```

Where did those numbers come from?

What is the difference?

The Dot Operator

- The dot operator signifies an element-by-element operation.
- The dot can be used for multiplication .*, division ./, or exponentiation .^ of elements of vectors that are the same size.
- Omitting the dot before an arithmetic operator means MATLAB performs the matrix version of the operation.

Exercise 2, Q5, page 16

Create a vector ***t*** that ranges **from 1 to 10 in steps of 1**, and a vector *theta* that ranges from **0 to π and contains 32 elements**. Now compute the following:

$$x = 2\sin(\theta)$$

$$y = \frac{t-1}{t+1}$$

$$z = \frac{\sin(\theta^2)}{\theta^2}$$

Accessing Elements of Matrices

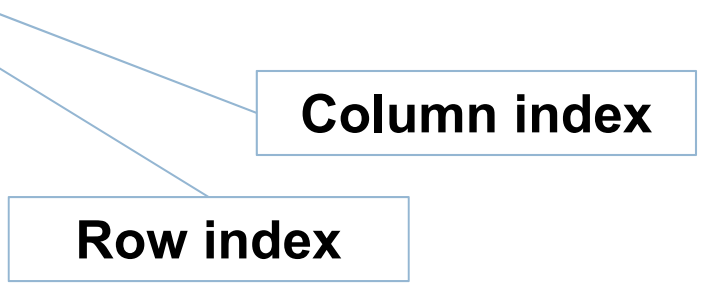
```
>> w = [1 2 3 4; 5 6 7 8; 9 10 11 12]
w =
     1     2     3     4
     5     6     7     8
     9    10    11    12

>> size(w)
ans =
     3     4
```

- The `size` command returns the number of rows and columns in the matrix.

Accessing Elements of Matrices

```
>> w(3,1)
ans =
     9
```



- when accessing an individual element in a matrix, the first number after the round bracket refers to the row number (**row index**), and second number refers to the column number (**column index**).

```
w =
     1     2     3     4
     5     6     7     8
     9    10    11    12
```

Accessing Elements of Matrices

```
>> w(3,:)
ans =
     9    10    11    12
```

- Colon operator (:) is used to denote all of the columns, i.e. all the columns in the third row are selected.
- Colon operator can also be used as a row index to denote all rows. Try:

```
>> w(:,3)
```

```
w =
     1     2     3     4
     5     6     7     8
     9    10    11    12
```

Accessing Elements of Matrices

```
>> w(2,4) = 13
w =
     1     2     3     4
     5     6     7    13
     9    10    11    12
```

- Here, we are accessing a single element in the matrix w to **change its value** (from 8 to 13)

```
w =
     1     2     3     4
     5     6     7     8
     9    10    11    12
```

Accessing Elements of Matrices

```
>> v = w(1:2,2:3)
v =
     2     3
     6     7
```

```
w =
     1     2     3     4
     5     6     7    13
     9    10    11    12
```

- a ***New matrix*** v is created as a ***sub-matrix*** of w.

```
>> z = w([2,3],[2,4])
z =
     6    13
    10    12
```

- **Square brackets** are used within the round brackets to enclose the list of row and column numbers.

Exercise

- The following matrix is defined:

```
M =
     6    9   12   15   18   21
     4    4    4    4    4    4
     2    1    0   -1   -2   -3
    -6   -4   -2    0    2    4
```

- Evaluate the following expressions:
  a) `A = M([1,3],[2,4])`
  b) `B = M(:,[1,4:6])`
  c) `C = M([2,3],:)`

TEXT STRINGS, INPUT, AND DISPLAY FUNCTIONS

Text Strings and `Display` Functions

Text strings are entered into MATLAB surrounded by **single quotes**. For example,

```
>> s = 'This is a test'
```

assigns the given text string to the variable **s**.

Text strings can be displayed with the function **`disp`**.

For example:

```
>> disp('this message is hereby displayed')
```

returns

```
this message is hereby displayed
```

The `blanks` command

- The **`blanks`** command enters a string of blanks, and is used with the **`disp`** command; **`blanks(n)`** is a string of **`n`** blanks. For example:

```
>> disp(['xxx' blanks(20) 'yyy'])
```

will give:

`xxx                    yyy`

(Notice the 20 blank spaces between xxx and yyy)

The `blanks` command

Typing `>> disp( blanks(n)' )`

moves the cursor down n lines. For example:

```
>> A=5
>> disp( blanks(3)')
>> M=10
```

returns:

```
A =
     5


M =
    10
```

The `input` Function

Used to request user input and assign it to a variable.

Example:

```
>> x = input('Enter a number: ');
```

This will display the text “Enter a number:” in the Command Window and then wait until the user enters something to be assigned to the variable x.