Programming Language: MatLab

1st Semester 2015

Chong-Wai

W02, 21st Sep

Content

- Index of an array
- Matrix operation
  - Add
  - Subtract
  - Multiply
  - Divide
- Variable/Data type
- Data import
- Plotting a 3D figure
- Homework

Index of an array

Two methods to access the element(s) of an array:

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

$m_{i,j}$

index i index j

m(i)

Single index counting along the 1st column and then 2nd column, then …

e.g.

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

M(2,3) = 7

M(3,1) = 4

M(2:3,1:2) = $\begin{bmatrix} 3 & 5 \\ 4 & 9 \end{bmatrix}$

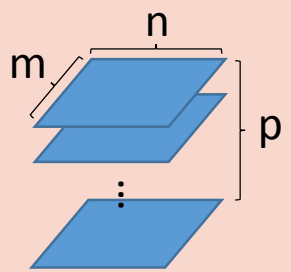
M(2) = 3

M(5) = 5

M(7) = 6

M(2:5) = [3 4 1 5]

Index of an array

	For physics	For MatLab/ programming language
$\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$	n-dimension 1st rank tensor, Vector	1-dimension array with n elements
$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & \dots & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}$	mxn-dimension 2st rank tensor, Matrix	2-dimension array with mxn elements
	mxnxp-dimension 3rd rank tensor	3-dimension array with mxnxp elements

Matrix operation

Matrix creation:

1) User define

e.g. M = [8 1 6; 3 5 7;4 9 2] or [8,1,6; 3,5,7; 4,9,2]

" " or "," => switch to next element in the same row

";" => switch to next column

(the number of elements in each row should be the same)

```
>> M = [8 1 6; 3 5 7; 4 9 2]

M =

     8     1     6
     3     5     7
     4     9     2
```

2) zeros(m,n)

```
>> zeros(3,2)

ans =

     0     0
     0     0
     0     0
```

3) ones(m,n)

```
>> ones(2,3)

ans =

     1     1     1
     1     1     1
```

4) eye(m)

```
>> eye(3)

ans =

     1     0     0
     0     1     0
     0     0     1
```

5) rand(m,n)

```
>> rand(3,2)

ans =

    0.8147    0.9134
    0.9058    0.6324
    0.1270    0.0975
```

…

Matrix operation

Add and subtract:

1) With scalar

e.g. a = 3, a+M = [a + m_{ij}]

a + M

```
>> a + M

ans =

    11     4     9
     6     8    10
     7    12     5
```

a - M

```
>> a - M

ans =

    -5     2    -3
     0    -2    -4
    -1    -6     1
```

2) With matrix

M

```
M =

     8     1     6
     3     5     7
     4     9     2
```

O

```
O =

     1     1     1
     1     1     1
     1     1     1
```

M+O

```
>> M+O

ans =

     9     2     7
     4     6     8
     5    10     3
```

M-O

```
>> M-O

ans =

     7     0     5
     2     4     6
     3     8     1
```

Elements are directly added/subtracted

(the dimension and size of two matrices should be the same)

Matrix operation

Matrix multiplication:

1) With scalar

e.g. a = 3, a*M = [a*m_{ij}]

M

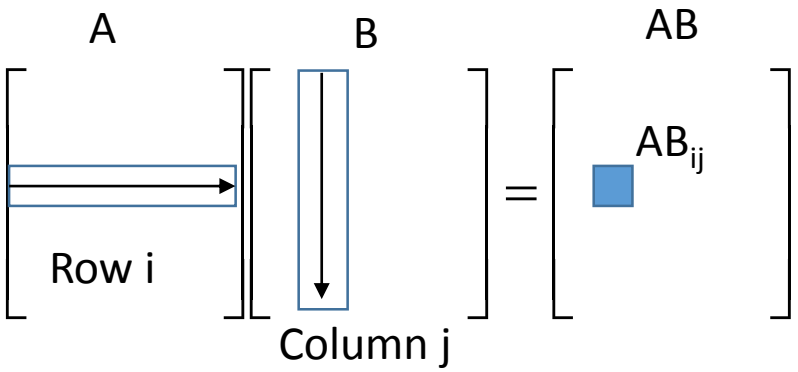
```
M =

     8     1     6
     3     5     7
     4     9     2
```

a*M

```
>> a*M

ans =

    24     3    18
     9    15    21
    12    27     6
```

2) With matrix, "*" and ".*"

For A*B: $AB_{ij} = \sum_{p} A_{ip} B_{pj}$



A

```
A =

     1     2     3
     4     5     6
```

B

```
B =

     1     0     0
     0     1     0
     0     0     1
```

A*B

```
>> A*B

ans =

     1     2     3
     4     5     6
```

(* the number of elements in row of first matrix should be the same as the number of elements in the column of the second matrix)

Matrix operation

Matrix multiplication :

1) With scalar

e.g. a = 3, a*M = [a*m_{ij}]

M

```
M =

     8     1     6
     3     5     7
     4     9     2
```

a*M

```
>> a*M

ans =

    24     3    18
     9    15    21
    12    27     6
```

2) With matrix, "*" and ".*"

For A*B: $AB_{ij} = \sum_{p} A_{ip} B_{pj}$

For A.*C: $AC_{ij} = A_{ij} C_{ij}$

A

```
A =

     1     2     3
     4     5     6
```

A*B

```
>> A*B

ans =

     1     2     3
     4     5     6
```

C

```
C =

     2     3     4
     5     6     7
```

A.*C

```
>> A.*C

ans =

     2     6    12
    20    30    42
```

B

```
B =

     1     0     0
     0     1     0
     0     0     1
```

A.*B =?

(* the number of elements in the row of first matrix should be the same as the number of elements in the column of the second matrix)
(.* the dimension of two matrix should be identical => elements by elements multiplication)

Matrix operation

Matrix division:

1) With scalar

e.g. a = 3, a \ M = M / a = $[m_{ij}/a]$

M

```
M =

     8     1     6
     3     5     7
     4     9     2
```

a / M or M \ a =>

```
>> M\a
Error using \
Matrix dimensions must agree.
```

```
>> a/M
Error using /
Matrix dimensions must agree.
```

a\M

```
>> a\M

ans =

    2.6667    0.3333    2.0000
    1.0000    1.6667    2.3333
    1.3333    3.0000    0.6667
```

M/a

```
>> M/a

ans =

    2.6667    0.3333    2.0000
    1.0000    1.6667    2.3333
    1.3333    3.0000    0.6667
```

a./M = M.\a = $[a/m_{ij}]$

```
>> a./M

ans =

    0.3750    3.0000    0.5000
    1.0000    0.6000    0.4286
    0.7500    0.3333    1.5000
```

```
>> M.\a

ans =

    0.3750    3.0000    0.5000
    1.0000    0.6000    0.4286
    0.7500    0.3333    1.5000
```

M./a = a.\M

```
>> M./a

ans =

    2.6667    0.3333    2.0000
    1.0000    1.6667    2.3333
    1.3333    3.0000    0.6667
```

a.\M = M./a = $[m_{ij}/a]$

(Please beware "." and the direction of "\" or "/")

Matrix operation

Matrix division:

2) With matrix, "\" and ".\"

$B/M \equiv B*M^{-1}$

where $M*M^{-1} = I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Identity matrix

M

```
M =

     8     1     6
     3     5     7
     4     9     2
```

B

```
B =

     1     0     0
     0     1     0
     0     0     1
```

M\B = B/M

```
>> M\B

ans =

    0.1472   -0.1444    0.0639
   -0.0611    0.0222    0.1056
   -0.0194    0.1889   -0.1028
```

$M.\backslash B = B./M = B_{ij}/M_{ij}$

$B.\backslash M = M./B = M_{ij}/B_{ij}$

Element by element division

M.\B = B./M

```
>> M.\B

ans =

    0.1250         0         0
         0    0.2000         0
         0         0    0.5000
```

B.\ M = M./B

```
>> B.\M

ans =

     8   Inf   Inf
   Inf     5   Inf
   Inf   Inf     2
```

Variable/Data type

Typical data type:

Data type	Size (bit)	Range
uint8	8	0 ~ 255 (2^8 - 1)
uint16	16	0 ~ 65535 (2^16 - 1)
uint32	32	0 ~ 2^32 - 1
uint64	64	0 ~ 2^64 – 1
int8	8	-128 ~ 127 (2^7 – 1)
int16	16	-32,768 ~ 32,767 (2^15 - 1)
int32	32	-2^31 ~ 2^31 - 1
int64	64	-2^63 ~ 2^63 – 1
double	64	Homework
single	32	Homework

Variable/Data type

Different data types can be converted by using the build-in function

http://www.mathworks.com/help/matlab/numeric-types.html

Functions

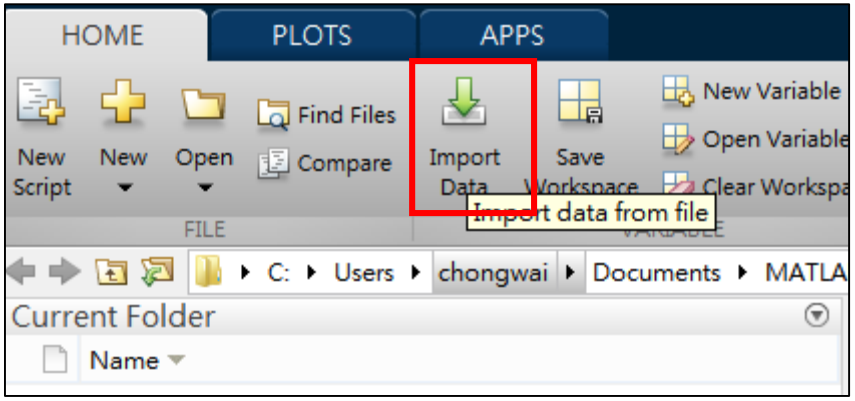
`double`	Convert to double precision
`single`	Convert to single precision
`int8`	Convert to 8-bit signed integer
`int16`	Convert to 16-bit signed integer
`int32`	Convert to 32-bit signed integer
`int64`	Convert to 64-bit signed integer
`uint8`	Convert to 8-bit unsigned integer
`uint16`	Convert to 16-bit unsigned integer
`uint32`	Convert to 32-bit unsigned integer
`uint64`	Convert to 64-bit unsigned integer

Data import

Method 1:

1) Choosing the button “Import Data” on the upper panel

2) Selection your file



3) Specify the range, the variable name and data type

4) Press the button “Import selection”

Variable name
e.g. T, X and Y

Data type
e.g. Number

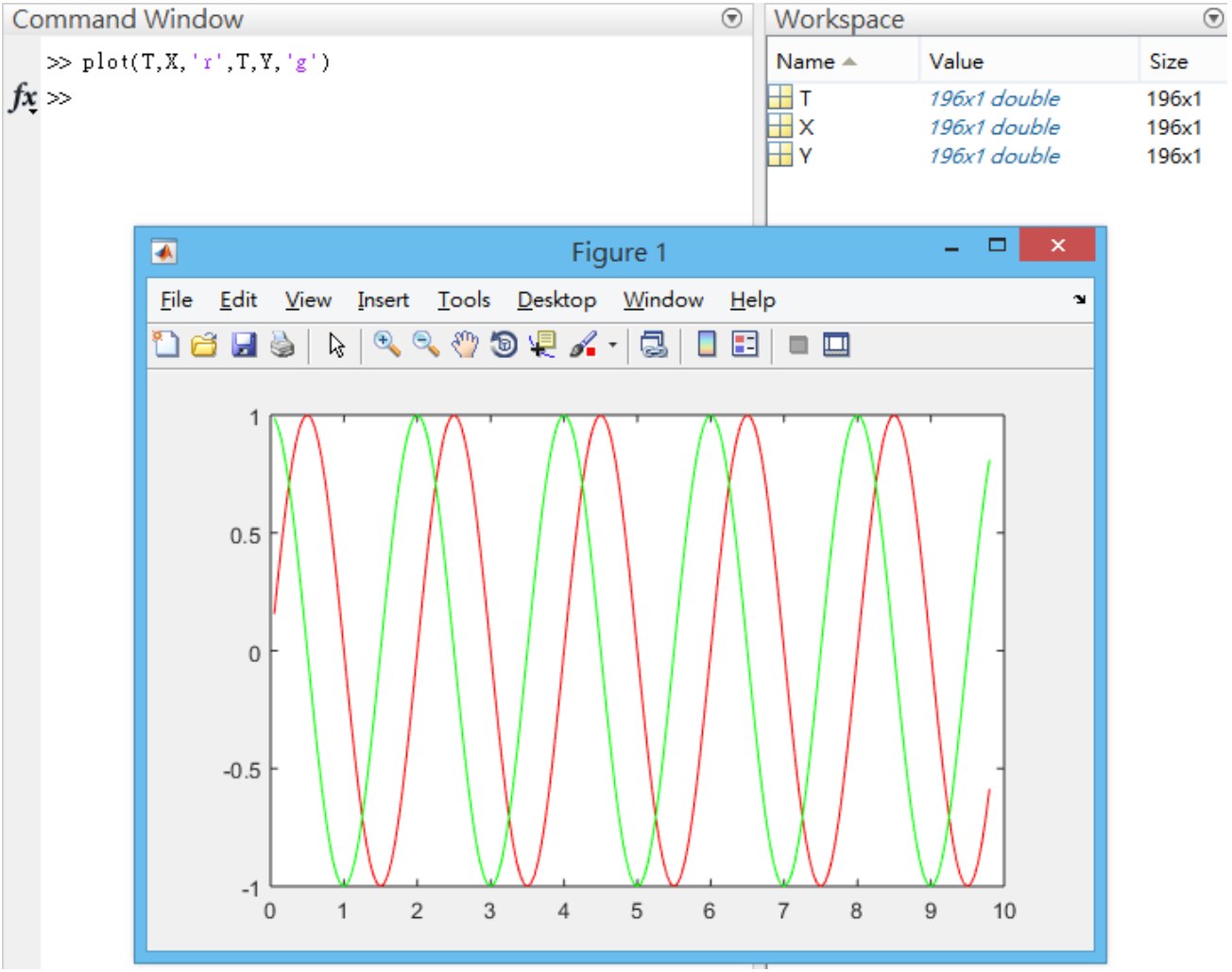
W02_1_data.xlsx

	A T	B X	C Y
	Number	Number	Number
1	T	X	Y
2	0.0500	0.1564	0.9877
3	0.1000	0.3090	0.9511
4	0.1500	0.4540	0.8910
5	0.2000	0.5878	0.8090
6	0.2500	0.7071	0.7071
7	0.3000	0.8090	0.5878
8	0.3500	0.8910	0.4540
9	0.4000	0.9511	0.3090
10	0.4500	0.9877	0.1564
11	0.5000	1.0000	2.6795e-08
12	0.5500	0.9877	-0.1564
13	0.6000	0.9511	-0.3090
14	0.6500	0.8910	-0.4540
15	0.7000	0.8090	-0.5878

Data import

Method 1:

5) Confirm the data you just imported



Data import

Method 2:

1) using the function “xlsread”

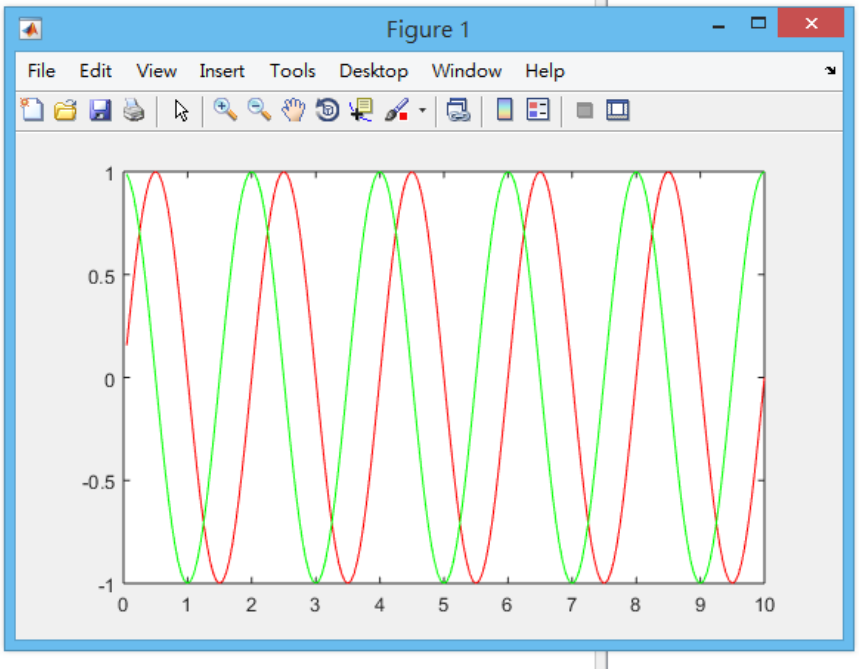
```
filename = 'myExample.xlsx';
sheet = 1;
xlRange = 'B2:C3';

subsetA = xlsread(filename,sheet,xlRange)
```

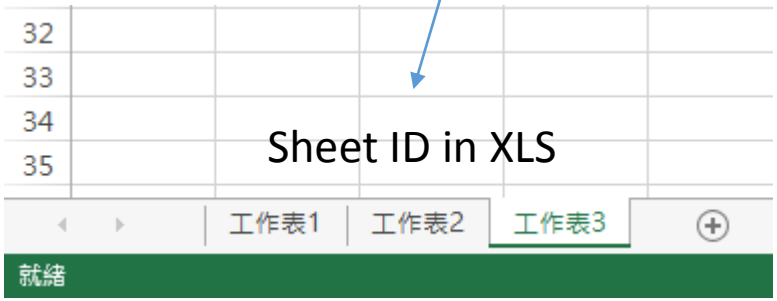
Command Window

```
>> A = xlsread('W02_1_data.xlsx',1);
>> T2 = A(:,1);X2 = A(:,2); Y2 = A(:,3);
>> plot(T2,X2,'r',T2,Y2,'g')
>>
```

Workspace

Name	Value	Size
A	200x3 double	200x3
T	196x1 double	196x1
T2	200x1 double	200x1
X	196x1 double	196x1
X2	200x1 double	200x1
Y	196x1 double	196x1
Y2	200x1 double	200x1

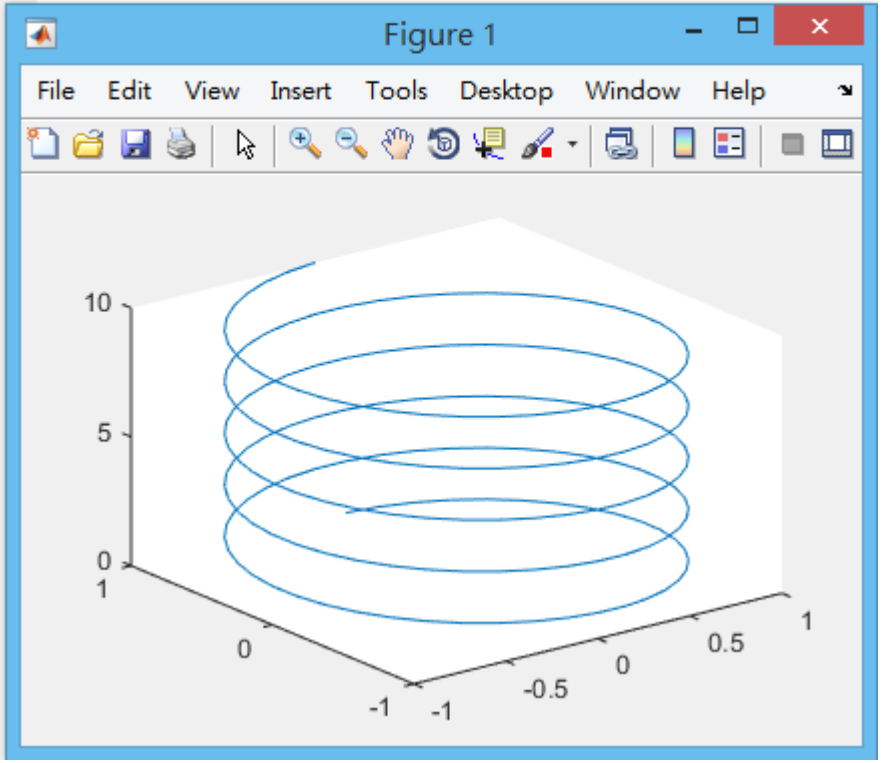




Plotting a 3D figure

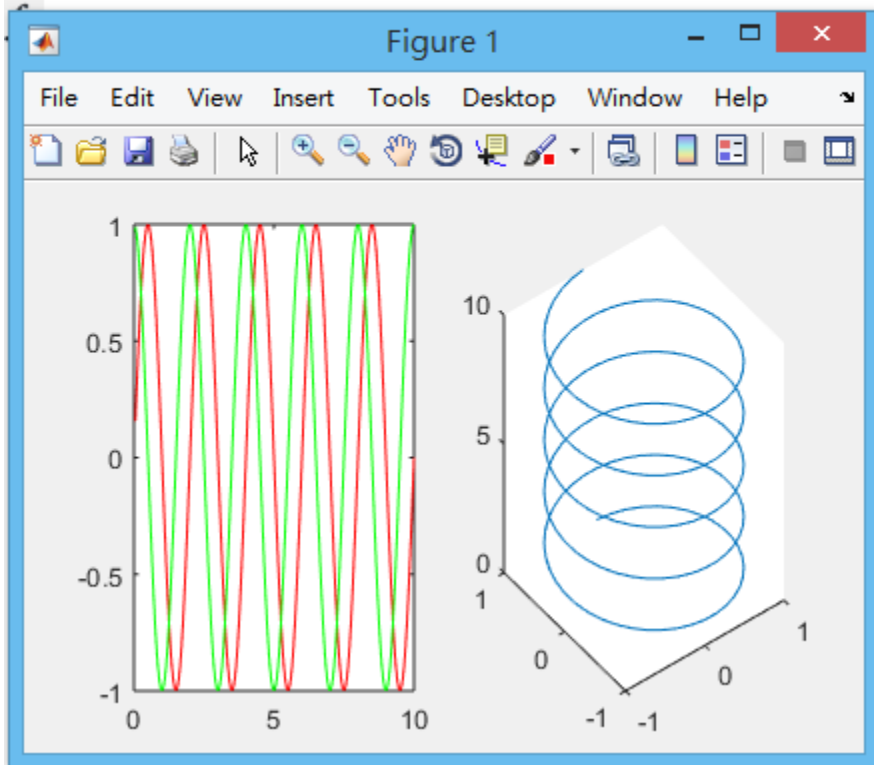
Plot3(x,y,z)

```
>> plot3(X2,Y2,T2)
>>
```



Subplot(m,n,p)

```
>> subplot(1,2,1); plot(T2,X2,'r',T2,Y2,'g');
>> subplot(1,2,2); plot3(X2,Y2,T2)
```



m and n: number of figure in x and y direction, respectively

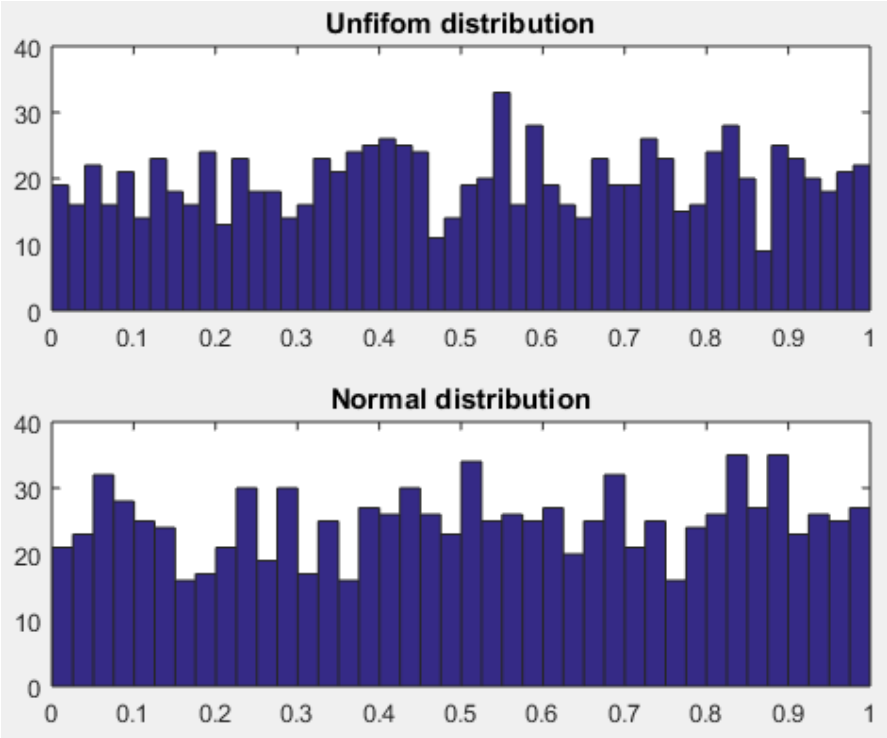
p: plot ID

Homework

1) Figure out what is the different between “double” and “single” and their range.

2) Using “help” in MatLab to learn how to use the function “hist” (histogram)

3) Using function “rand” and “randn” to generate two 5000x1 arrays “xU” and “xN”, respectively, and try to plot the figure shown below (50 bins)



Homework

Cation:

1) Please naming the file name of you home work as "HW01_G##_XXX_XXX.ppt", where ## and XXX are the group number and your name, respectively.

2) Please submit your homework on time (before Friday noon)

3) Please specify the contribution of each member in the first page

4) Please do not copy your HW from your classmate, but you can discuss