

USC GSBME MATLAB CLASS

Second session

Reviewing previous session

Practice time!

A. Matrix-wise and element-wise operations

A.1. Matrix-wise operations

So far we learned how to define variables and how to extract data from them. Here we will learn how to use them.

Assume that:

$x = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

$y = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$

$z = x + y$

Do it yourself!

So we will have

$z =$

$\begin{bmatrix} 6 & 8 \end{bmatrix}$

$\begin{bmatrix} 10 & 12 \end{bmatrix}$

Please note that since it is not possible to add matrices with different sizes (rows and columns), if you try to do so, MATLAB will give you an error.

Practice time!

- Try to define two matrices and subtract them.

Now, assume that:

$x = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

$y = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$

$z = x * y$

Note: in MATLAB, asterisk (“*”) represents the multiplying operator,

Do it yourself!

So we will have

$z =$

19 22

43 50

Which is the result of multiplication of the x and y matrixes.

Please note that in matrix multiplication, we can just multiply matrixes with same inner dimensions.

Example:

$A_{m \times n} * B_{n \times p}$

Where n is the inner dimension of this multiplication.

Practice time!

- Try to divide one matrix to another one.

A.1. Element-wise operations

If we want the operator to operate element-wise (the operator will work on each one of the elements from the first matrix and its corresponding element on the other matrix), we should just put a dot (“.”) in front of the operator.

Example:

$$C=A.*B$$

$$C=A.^B$$

Practice time!

- Practice the skill you’ve learned so far.
- Assume that:

$x=[1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10]; y=[1000\ 100\ 10\ 1]$

try to multiply the third to the sixth elements of the matrix x to the elements of the matrix y with one MATLAB command. The answer should be 3,456.

Notes:

B. Math functions in MATLAB

MATLAB has a diverse library of math functions. Here we will introduce some of the most frequently used ones and show how to use them.

Let's start with the sinusoid function. To calculate the sinusoid of an input number, we use the MATLAB function `sin()`. It is used as `output=sin(input)`. Note that MATLAB will interpret the input as it is in radians. You can also use "pi" for the π number.

Example:

```
y=sin(pi/2)
```

and we will have:

```
y=1
```

or

```
x=pi/2;
```

```
y=sin(x)
```

and we will have:

```
y=1
```

Below is a list of some of the most frequently used math functions in MATLAB.

B.1. Sinusoidal functions

`sin`, `sinh`, `asin`, `asinh`, `cos`, `cosh`, `acos`, `acosh`, `tan`, `tanh`, `atan`, `atanh`, `sec`, `sech`, `asec`, `asech`, `csc`, `csch`, `acsc`, `acsch`, `cot`, `coth`, `acot`, `acoth`, `sinc`.

B.2. Some other frequently used Math functions

exp, fix, log, floor, log10, ceil, log2, round, sqrt, rem, nextpow2, mod, abs, sign, angle, conj, imag, real.

Practice time!

- Investigate what each of the functions above do (using MATLAB help) and use at least 5 of them in the MATLAB environment.

So far we have used the sinusoid function as a one input-one output function. What if we want to enter a string of numbers and have their sinusoids as a string with elements corresponding to the input elements?

You can easily create a matrix as an input to the functions above and they will return the output as a matrix with elements corresponding to the elements in the input matrix.

Example:

`sin([pi/2 0; 0 pi/2])`

ans =

1 0

0 1

Notes:

C. 2D plotting in MATLAB

Here we will explain how to plot data using an example. Let's say we want to plot a sinusoid function for one period (as you know, sinusoid function is a periodic function with the periodicity equal to 2π).

First we will need a time series. We will use the next command to make a time series starting from 0 and ending in 2π with increments of 0.01.

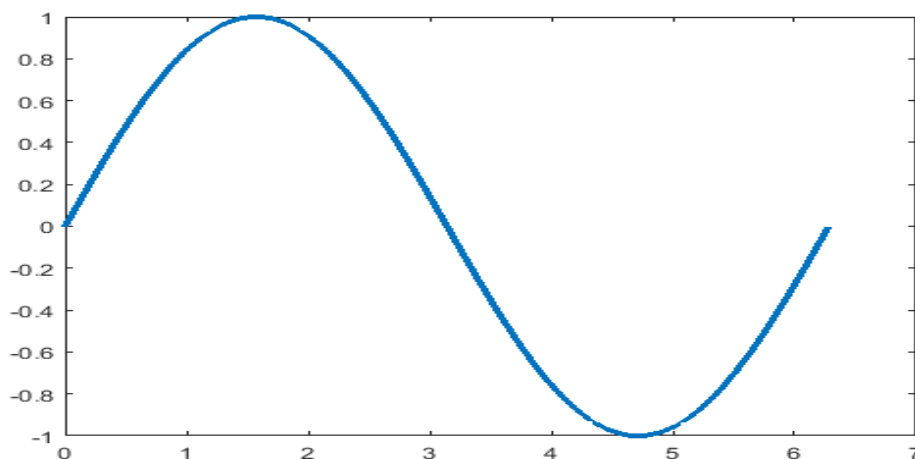
```
t=0:0.01:2*pi;
```

then we will use the `sin()` function to calculate the sinusoid of any given time on the “t” string and name the output as y.

```
y=sin(t);
```

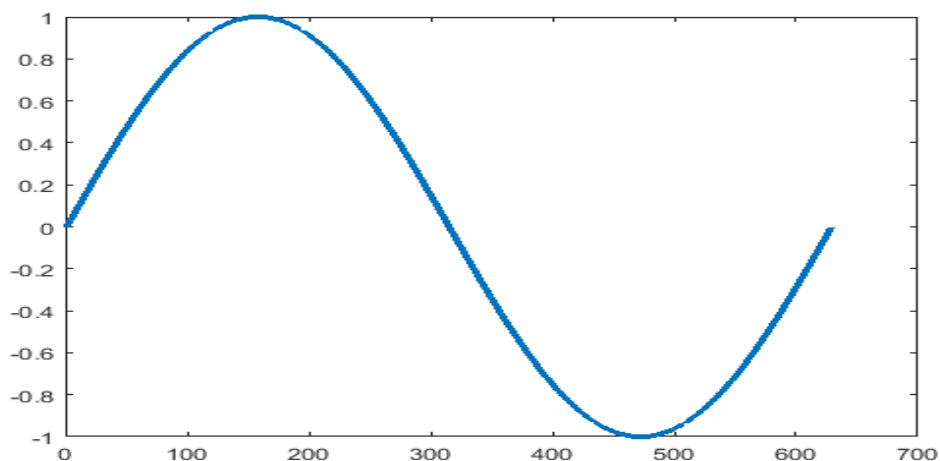
now we can use the plot function in the form of `plot(“independent variable”, “dependent variable”)`. It means we have to type `plot(t,y)` on the command window.

The result will be shown in a new figure box. You should see a result similar to the one shown below:



In fact, you can just use plot function with only one argument. In this case, your x axis (independent variable) will be the sample number.

plot(y)



(You can also use the stem function instead of plot to illustrate your data in a discrete manner)

Practice time!

- Try to plot the following function for the $t=0:0.01:2$ interval.

$$y = 5e^{-3t}$$

Notes:

D. Polynomials in MATLAB

In MATLAB, any polynomial will be described as a string of numbers (let's call it C) which represents the coefficients as described below:

$$C=[C_m \ C_{m-1} \ \dots \ C_1 \ C_0];$$

Represents:

$$C_m X^m + C_{m-1} X^{m-1} + \dots + C_1 X + C_0$$

- To find the roots of a polynomial:
 $R=\text{roots}(C)$
- To find the coefficients for a known set of roots:
 $C=\text{poly}(R)$
- To compute the value of the polynomial with coefficients C at the input equals to x:
 $y=\text{polyval}(C,x)$
- To differentiate from the polynomial with coefficients C:
 $dC=\text{polyder}(C)$
- To integrate from the polynomial with coefficients C:
 $IC=\text{polyint}(C)$
Please note that MATLAB will set the integration constant to zero.
- To find the coefficients that its polynomial fits best to a set of data:
 $C=\text{polyfit}(x,y,n)$

Where x is the input (independent variable), y is the output (dependent variable which we want to fit our polynomial to) and n is the polynomial order.

Practice time!

- Try to choose at least two of the functions above (except polyfit) and use them in MATLAB
- Try to find the set of coefficients for a polynomial of order 5 that fits best the input-output that described below:

Input: $x=0:0.05:2\pi$;

Output: $y=\sin(x)$;

After you find the coefficients, try to plot those using polyval and plot functions to see how good it is fitted on your data.

- Tip: You can use “hold on” command after your first plot to let the other plots to be plotted on the same figure.

Notes: