

Matlab Notes for Calculus 1

Lia Vas

Content

1. Introduction to Matlab
2. Basic arithmetic
3. Solving equations using **solve**
4. Representing functions
5. Graphing
6. Solving equations using **fzero**
7. Limits
8. Differentiation
9. Optimization
10. Integration
11. Practice Problems

1. Introduction to Matlab

The main Matlab window is divided into four (or five, depending on the version) parts:

Menu Bar (at the very top)

Current Folder

Command Window

Workspace

Clicking on “New Script”, which is the very first command in the Menu Bar, the Matlab window opens another part, called the

Editor

In some versions of Matlab, the Editor automatically opens on the start.

All Matlab commands are executed in the **Command Window**. If your code is not longer than one line, you can type it in the Command Window and execute it by pressing “enter”.

The **Editor** is used for more complex code. You can have multiple Editor tabs open if you need to work on more than one set of code at a time. The default file is titled “Untitled” so when you type some code in it, you need to save the file and give it a meaningful name. One benefit of typing the code in the Editor is that you can save it while the code typed in the Command Window cannot be saved, just executed. Hence, you can execute the saved code multiple times while executing a command in the Command Window again requires typing the whole command again.

The Directory (**Current Folder** part) shows your current location. If you want to execute some

code typed in the Editor, change the location of your current folder to match the folder where you saved the code from the Editor. The current location of the Directory is listed in the long white bar above the Editor.

You can change your current folder by clicking on the folder icon with a green arrow (see the icons on the right side of the white bar listing your current directory). Note that the directory that is opened when Matlab starts is usually not the directory where your files are saved so you will need to change the current folder.

The **Workspace** lists all variables that have been defined and specifies their type and value.

The **Menu Bar** contains some familiar commands, such as New, Open, and Save. When the Editor is open, the Menu Bar contains also Comment and Run. Comment allows us to write text in the code that is not executed which helps your code be more meaningful to you, the programmer, or to a user. Alternatively, you can type “%” before any text and it will be included as a comment.

2. Basic Arithmetic

You can use +, -, *, \ and ^ to add, subtract, multiply, divide or raise to a power, respectively. For example if you enter:

```
>> 2^3 - 2*2
```

Matlab calculates the answer:

```
ans = 4
```

If you want to perform further calculations with the value of the answer, you can type **ans** rather than retyping the specific answer value. For example,

```
>> sqrt(ans)
```

```
ans = 2
```

To perform symbolic calculations in Matlab, use **syms** to declare the variables you plan to use. For example, suppose that you need factor x^2-3x+2 . First you need

```
>> syms x (you are declaring that x is a variable)
```

Then you can use the command **factor**.

```
>> factor(x^2-3*x+2)
```

```
ans = (x-1)*(x-2)
```

Note that we entered **3*x** to represent $3x$ in the command above. **Entering * for multiplication is always necessary in Matlab.**

Besides **factor** command, you have **simplify** and **expand**.

3. Solving equations using “solve”

For solving equations, you can use the command **solve**.

- Represent the variable you are solving using **syms** command.
- Move every term to the left side of the equation so that the equations of the form $g(x)=h(x)$ become $g(x)-h(x)=0$
- If the term on the left side is $f(x)$ and the equation is $f(x)=0$

the command you want to execute is

solve(f(x))

Note that the left side of the equation is in parenthesis. Thus, the command **solve** has the following form

solve(the left side of the equation if the right side is 0)

For example, to solve the equation $x^3-2x-4=0$, we can use

```
>> solve(x^3-2*x-4)
```

and get the following answer

```
ans = 2 -1+i -1-i .
```

Here i stands for the imaginary number $\sqrt{-1}$. This answer tells us that there is just one real solution, 2.

The command **solve** often produces a symbolic answer. For example, let us solve the equation $3x^2-8x+2=0$. When executing

```
>> solve(3*x^2-8*x+2)
```

```
ans = 4/3-10^(1/2)/3 10^(1/2)/3+4/3
```

If we want to get, often more meaningful, numerical answer in the decimal form with, say, three significant digits, we can use the command **vpa**, refer to the previous answer as **ans**, and specify how many digits of the answer we need to see. For example.

```
>> vpa(ans, 3)
```

```
ans = 0.279 2.39
```

The command **vpa** has the general form

vpa(expression you want to approximate, number of significant digits)

4. Representing a function

To represent a function given by a formula containing a variable x , start by **syms x** if you have not defined x to be your variable already. If we want to call the function f , the following command defines $f(x)$ to be a function defined by the given formula.

f = @(x) formula defining the function

For example, the following command defines the function x^2+3x-2 .

```
>> f = @(x) x^2+3*x-2
```

```
f(x) = x^2+3*x-2
```

After defining a function, we can evaluate it at a point. For example,
>> f(2) produces the answer **ans = 8.**

If a function is simple, it might be faster to evaluate a function at a point simply by typing the value of x directly for x . For example, to evaluate $\sin(x)$ at $x=2$, simply type
>> sin(2) and obtain the answer **ans = .909297.**

The following table gives an overview of how most commonly used functions or expressions are represented in Matlab.

function or symbol	representation in MATLAB
e^x	exp(x)
$\ln x$	log(x)
$\log x$	log(x)/log(10)
log. base a of x	log(x)/log(a)
$\sin x$	sin(x)
$\cos x$	cos(x)
arctan(x)	atan(x)
π	pi

As when using the calculator, one must be careful when representing a function. For example,

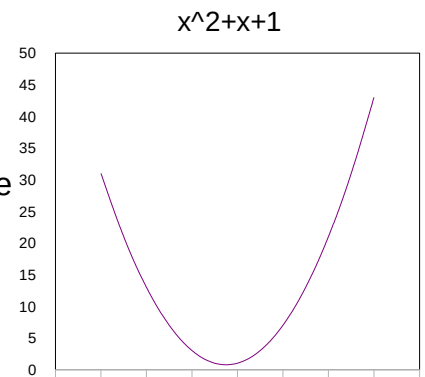
- $\frac{1}{x(x+6)}$ should be represented as **1/(x*(x+6))** not as **1/x*(x+6)** nor as **1/x(x+6),**
- $\frac{3}{x^2+5x+6}$ should be represented as **3/(x^2+5*x+6)** not as **3/x^2+5*x+6,**
- e^{5x^2} should be represented as **exp(5*x^2)** not as **e^(5*x^2), exp*(5*x^2), exp(5x^2)** nor as **exp^(5*x^2).**
- $\ln(x)$ should be represented as **log(x),** not **ln(x).**
- $\log_3(x^2)$ should be represented as **log(x^2)/log(3)** not as **log(x)/log(3)*x^2.**

5. Graphing

Let us start by declaring that x is a variable:
>> syms x

The simplest command in Matlab for graphing is **ezplot**. The command has the following form
ezplot(function)

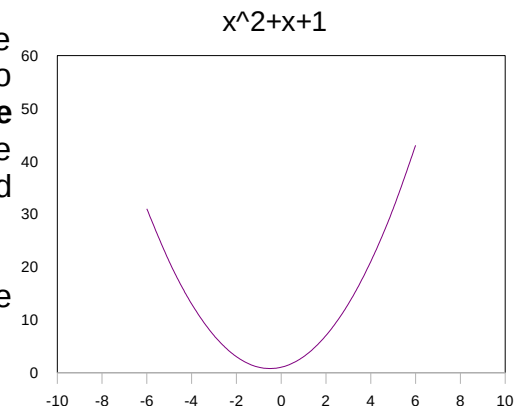
For example, to graph the function x^2+x+1 , you simply type
>> ezplot(x^2+x+1)



A new window will open and graph will be displayed. To copy the figure to a text file, you can choose the format in which you want to save the figure file (go to the **File** menu of the figure, choose **Save as**, and then select the location, file name and file type, for example png). You can import the figure into the word file using **Insert** and **Image**.

We can specify the different scale on x and y axis. To do this, the command **axis** is used. It has the following form

axis([Xmin, Xmax, Ymin, Ymax])



This command parallels the commands in menu WINDOW on the TI83 calculators.

For example, to see the above graph between x-values -10 and 10 and y-values 0 and 60, you can enter

```
>> axis([-10 10 0 60])
```

Note that the domain of function did not change by command axis. To see the graph on the entire domain (in this case [-10, 10]), add that domain after the function in the command ezplot:

```
ezplot(function, [Xmin, Xmax])
```

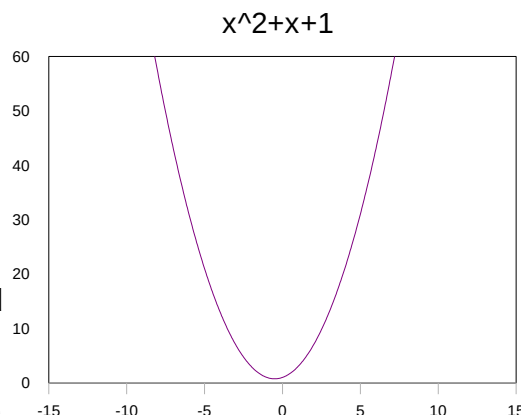
In this case,

```
>> ezplot(x^2+x+1, [-10, 10])
```

will give you the desired graph.

For the alternative command for graphics, **plot**, you can find more details by typing **help**.

To graph multiple curves on the same plot, you can also use the **ezplot** command.



To graph multiple curves on the same window, you can use the **ezplot** command in combination with hold on and hold off on the following way:

```
ezplot(1st function)
hold on
ezplot(2nd function)
ezplot(3rd function)
...
ezplot(n-th function)
hold off
```

For example to graph the functions $\sin(x)$ and e^{-x^2} , you can use:

```
>> ezplot(sin(x))
>> hold on           >> ezplot(exp(-x^2))           >> hold off
```

6. Solving equations using “fzero”

In some cases, the command **solve** may fail to produce all the solutions of an equation. In those cases, you can try to find solutions using **fzero** (short for “find zero”) command. Just as for **solve**, you need to write equation in the form

$$f(x)=0.$$

So, you need to put all the terms of the equations on one side leaving just zero on the other. To find a solution near the x-value $x=a$, you can use

```
fzero('left side of the equation', a)
```

The command **fzero**, similarly as **solve** is always followed by expression in parenthesis. Note that the equation should be in single quotes.

If it is not clear what a convenient x -value a should be, you may want to graph the function on the left side of the equation first, check where it intersects the x -axis. Alternatively, you can graph left and right side of the equation that is not in $f(x)=0$ form and see where the two functions intersect. Then decide which x -value you should use.

Example. To solve the equation $e^{x^2}-2=x+4$, we can first graph the functions on the left and right side of the equation using

```
syms x      ezplot(exp(x^2)-2)      hold on      ezplot(x+4)      hold off
```

From the graph, we can see that the two functions intersect at a value near -1 and at a value near 1. To use **fzero**, we need to represent the equation in the form $e^{x^2}-2-(x+4)=0$ (or simplified form $e^{x^2}-x-6=0$). Then, we can find the positive solution by using **fzero** to find a zero near 1 and then to find the negative solution near -1, for example. Thus, both solutions can be obtained by:

```
>> fzero('exp(x^2)-2-(x+4)', 1)      ans = 1.415
>> fzero('exp(x^2)-2-(x+4)', -1)     ans = -1.248
```

Note also that the command **solve(exp(x^2)-2-(x+4))** returns just the positive solution. Thus, knowing how to use **fzero** command may be really useful in some cases.

7. Limits

You can use **limit** to compute limits, left and right limits as well as infinite limits. For example, to evaluate the limit when $x \rightarrow 2$ of the function $\frac{x^2-4}{x-2}$, we have:

```
>> syms x
>> limit((x^2-4)/(x-2), x, 2)      ans = 4
```

You can also evaluate left and right limits. For example:

```
>> limit(abs(x)/x, x, 0, 'left')    ans = -1
>> limit(abs(x)/x, x, 0, 'right')   ans = 1
```

Limits at infinity:

```
>> limit(exp(-x^2-5)+3, x, Inf)     ans = 3
```

8. Differentiation

Start by declaring x for a variable. The command for differentiation is **diff**. It has the following form

```
diff(function)
```

For example,

```
>> syms x
>> diff(x^3-2*x+5)      ans = 3*x^2-2
```

To get n-th derivative use

diff(function, n)

For example, to get the second derivative of x^3-2x+5 , use: **>> diff(x^3-2*x+5, 2)**

ans = 6*x

Similarly, the 23rd derivative of $\sin(x)$ is obtained as follows. **>> diff(sin(x), 23)**

ans = -cos(x)

To evaluate derivative at a point, we need to represent the derivative as a new function. For example, to find the slope of a tangent line to x^2+3x-2 at point 2, we need to find the derivative and to evaluate it at $x=2$.

>> diff(x^2+3*x-2) (first we find the derivative) **ans = 2*x+3**

>> f = @(x) 2*x+3 (then we representative the derivative as a function)

>> f(2) (and, finally, we evaluate the derivative at 2) Obtain **ans = 7**

You can make a derivative (or integral or an output of some other function operator) into a function automatically using **matlabFunction** command using the following format.

f = matlabFunction(command involving an operation on a given function)

For example, this last problem can be solved by

f = matlabFunction(diff(x^2+3*x-2))

followed by **f(2)**.

9. Optimization

Recall the steps needed in order to find minimum or maximum values of a given function (using second derivative test)

- Find first derivative
- Solve it for zeros. The x-values you obtain are called critical
- Find second derivative
- Plug critical points in second derivative. If your answer is negative, the function has a maximum value at a critical point used. If your answer is positive, the function has a minimum value at a critical point used.
- Plug critical points in your function. The y-values you obtain are your maximum or minimum values.

In MATLAB, start with **syms x**.

1. Finding derivative: **diff(function)**
2. Finding critical points: **solve(copy-paste the answer from step 1)**
3. Finding second derivative: **diff(function, 2)**
4. Evaluating second derivative at critical points: **g=@(x) paste the second derivative** Then use **g(critical value)** to find the value of the second derivative at the critical value.
5. Evaluating function at critical points: **f=@(x) function formula** followed by **f(critical value)**

For example, to find extreme values of x^3-2x+5 , start by finding first derivative:

```
>> diff(x^3-2*x+5)          ans = 3*x^2-2
Then find critical point(s):
>> solve(3*x^2-2)          ans = 6^(1/2)/3, -6^(1/2)/3  vpa(ans, 3)    ans = .816, -.816
```

```
Find second derivative      >> diff(x^3-2*x+5, 2)          ans = 6*x
Evaluate this at critical points. >> g=@(x) 6*x      g(.816)          ans = 4.896
Positive answer means that the function has minimum at x=.816
>> g(-.816)                ans = -4.896
Negative answer means that the function has maximum at x=.816
```

```
Finding y-values of maximum and minimum:
>> f=@(x) x^3-2*x+5
>>f(.816)                  ans = 3.911 This is the local minimum value.
>>f(-.816)                 ans = 6.088 This is the local maximum value.
```

10. Integration

We can use Matlab for computing both definite and indefinite integrals using the command **int**. For the indefinite integrals, start with **syms x** followed by the command

int(function)

For example, the command

```
>> int(x^2)
```

evaluates the integral $\int x^2 dx$ and gives us the answer **ans = 1/3*x^3**

For definite integrals, the command is

int(function, lower bound, upper bound)

For example,

```
>> int(x^2, 0, 1)
```

evaluates the integral $\int_0^1 x^2 dx$ The answer is **ans = 1/3**

Matlab can evaluate the definitive integrals of the functions that do not have elementary primitive functions. Recall that the integrals $\int \frac{\sin x}{x} dx$, $\int \frac{e^x}{x} dx$, $\int e^{x^2} dx$

can not be represented via elementary functions. Suppose that we need to find the integral of $\frac{\sin x}{x}$ from 1 to 3. The command **>> int(sin(x)/x, 1, 3)**

does not gives us a numerical value. We have just **ans = sinint(3)-sinint(1)**

Using the command **vpa**, we obtain the answer in numerical form. For example,

```
>> vpa(ans, 4) gives us    ans = 0.9026
```

11. Practice problems

- Factor $x^3+3x^2y+3xy^2+y^3$.
- Simplify $\frac{x^3-8}{x-2}$.
- Evaluate the following expressions. (a) $\sin(\pi/6)$ (b) $\frac{\sqrt{5}+3}{\sqrt{3}-1}$ (c) $\log_2(5)$
- Solve the following equations and express the answers as decimal numbers. (a) $x^3-2x+5=0$ (b) $\log_2(x^2-9)=4$.
- Let $f(x)=\frac{x^3+x+1}{x}$ (a) Represent $f(x)$ as a function in Matlab and evaluate it at 3 and -2. (b) Find x -value(s) that corresponds to y -value $y=2$. (c) Graph $f(x)$ on domain $[-4, 4]$.
- Graph $\ln(x+1)$ and $1-x^2$ on the same plot for x in $[-2, 6]$ and y in $[-4, 4]$.
- Find the limits of the following functions at indicated values. (a) $f(x)=\frac{x^{12}-1}{x^3-1}$, $x \rightarrow 1$ (b) $f(x)=3+e^{-2x}$, $x \rightarrow \infty$ (c) $f(x)=\frac{6x^3-4x+5}{2x^3-1}$, $x \rightarrow \infty$
- Let $f(x)=\frac{x^3+x+1}{x}$ Find the first derivative of $f(x)$ and evaluate it at $x=1$.
- Let $f(x)=e^{3x^2+1}$. (a) Find the first derivative of $f(x)$. (b) Find the slope of the tangent line to $f(x)$ at $x=1$. (c) Find the critical points of $f(x)$.
- Find the 12th derivative of the function $(\frac{x}{2}+1)^{65}$.
- Find the extreme values of (a) x^3-4x+8 (b) xe^{-3x}
- Evaluate the following integrals. (a) $\int xe^{-3x} dx$ (b) $\int_0^1 xe^{-3x} dx$.

Solutions.

- `syms x y` followed by `factor(x^3+3*x^2*y+3*x*y^2+y^3)` gives you `ans=(x+y)^3`
- `syms x` followed by `simplify((x^3-8)/(x-2))` gives you `ans=x^2+2x+4`
- (a) `sin(pi/6)` `ans=.5` (b) `(sqrt(5)+3)/(sqrt(3)-1)` `ans=7.152` (c) `log(5)/log(2)` `ans=2.3219`.
- (a) `solve(x^3-2*x+5)` `ans=-2.09`. (b) `solve(log(x^2-9)/log(2)-4)`. `ans=5, -5`.
- (a) `>> f=@(x) (x^3+x+1)/x`, `>> f(3)` `ans= 10.333`, `>>f(-2)` `ans=4.5`.
 (b) The problem is asking you to solve equation $\frac{x^3+x+1}{x}=2$. Using solve command, `solve(f(x)-2)`. you get `ans=-1.3247` (c) `ezplot(f(x), [-4,4])`.
- `hold on ezplot(log(x+1)) ezplot(1-x^2)` hold off `axis([-2 6 -4 4])`
- (a) `syms x limit((x^12-1)/(x^3-1), x, 1)` `ans=4`
 (b) `limit(3+exp(-2*x), x, Inf)` `ans=3` (c) `limit((6*x^3-4*x+5)/(2*x^3-1), x, Inf)` `ans=3`
- (a) `syms x diff((x^3+x+1)/x)` `ans = 2*x-1/x^2` or `(2*x^3-1)/x^2`.
 (b) `g=@(x) 2*x-1/x^2` then `g(1)` gives you `ans=1`.
- (a) `diff(exp(3*x^2+1))` `ans=6*x*exp(3*x^2+1)`
 (b) Represent the derivative as function `g=@(x) 6*x*exp(3*x^2+1)` then evaluate `g(1)`. Get `6*exp(4)`. To see the answer as a decimal number (say to five nonzero digits) use `vpa(ans, 5)`. Get 327.58. (c) `solve(6*x*exp(3*x^2+1))` `ans=0`
- `diff((x/2+1)^65, 12)`
- (a) `max(-1.15, 11.079)`, `min(1.15, 4.92)`. (b) `max(.333, .1226)`, no min.
- (a) `syms x int(x*exp(-3*x))` `ans=-1/3*x*exp(-3*x)-1/9*exp(-3*x)`
 (b) `int(x*exp(-3*x), 0,1)` `ans=-4/9*exp(-3)+1/9` `vpa(ans, 4)` `ans=.08898`