Matlab Brief Manual

Official Manual
https://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf
Functions Descriptions
https://www.mathworks.com/help/matlab/language-fundamentals.html
https://www.mathworks.com/help/matlab/mathematics.html
Tutorials
https://matlabacademy.mathworks.com

Contents

1 As Calculator

In "Command Window", type expression $\xrightarrow{\text{Press Enter}}$ Output

Quick Example

```
>> (1+2^0.5)^2+(1-2^0.5)^2
ans =
      6
```

Commands Table

Commands	Descriptions	Remarks
`+, -, *, /, ^`	add, subtract, multiply, divide, power	
`sin, cos, tan`	sine, cosine, tangent	in Radian
`asin, acos, atan`	arcsine, arccosine, arctangent	in Radian
`exp, log`	natural exponential function, natural logarithm	
`abs, sign`	absolute value, sign function	
`ceil, floor, fix, round`	upper integer, lower integer, integer part, nearest integer	
many others... e.g. `xxx`	type `help xxx` to obtain descriptions	

2 Variable

Quick Example

```
>> a=2,b=2*a+1
a =
    2
b =
    5
```

Rules for variable name:

1. **Case sensitive**

```
>> a1=2,A1=3,a1+A1
a1 =
    2
A1 =
    3
ans =
    5
```

2. **First letter must be English alphabet, the rest must be English alphabets, numbers, or underline** "_"

e.g. `x1`, `x_1`, `alpha_k` are valid,
`2nd_root`, `mid-point`, `total area` are invalid.

Commands Table

Commands	Descriptions	Before	After
;	suppress print	>> a=2,b=3,c=a+b a = 2 b = 3 c = 5	>> a=2;b=3,c=a+b b = 3 c = 5
format long	15-digit print	>> a=2^0.5 a = 1.4142 >> b=3^0.5 b = 1.7321	>>format long >> a=2^0.5 a = 1.414213562373095 >> b=3^0.5 b = 1.732050807568877
format short	5-digit print	>> a=log(2) a = 0.693147180559945 >> b=log(3) b = 1.098612288668110	>> format short >> a=log(2) a = 0.6931 >> b=log(3) b = 1.0986
clc	clear Command Window	>> format short >> a=log(2) a = 0.6931 >> b=log(3) b = 1.0986 >> clc	>>
clear	clear variables	>> a=5,b=10 a = 5 b = 10 >> a a = 5	a=5,b=10 a = 5 b = 10 >> clear >> a Undefined function or variable ‘a’.

Commands	Descriptions	Examples

`disp`	display text or variable value	1. `>> a=5;`
		`>> disp('a')`
		`a`
		2. `>> a=5;`
		`>> disp(a)`
		`5`

`fprintf`	display text and variable value	1. `>> a=2^0.5;`
		`>> fprintf('value of a is %7.5f\n',a)`
		`value of a is 1.41421`
		2. `>> a=2^0.5;`
		`>> fprintf('root 2 is %9.5f\n',a)`
		`root 2 is  1.41421`
		(`%9.5f` is the data format to be displayed,
		with `9` as the total space occupied by `a`,
		`5` as number of decimal places,
		`f` as floting point number.
		`\n` is the command for "new line")

`input`	user input	1. `>> a=input('The value of a is ')`
		`The value of a is` enter a value here
		⇒`The value of a is 3`
		`a =`
		`3`
		2. `>> a=input('choose a = ')`
		`choose a =` enter a value here
		⇒`choose a = 5`
		`a =`
		`5`

other functions	e.g. `sin`, `cos`, `log`, ...	`>> a=2;`
		`>> sin(a)`
		`ans =`
		`0.9093`
		`>> cos(2*a+1)`
		`ans =`
		`0.2837`
		`>> log(1+cos(a))`
		`ans =`
		`-0.5381`

3 Function

Quick Example

In "Editor"

```
function y = fun1(x)
y=1+2*x+3*x^2;
```

save as m.file
change current folder

In "Command Window"

```
>> b=fun1(-1)
b =
     2
```

Save & Load

To save: "New Sript" → construct the function → save → remember the folder.
To load: Change the "Current Folder" to the folder where the function m.file is saved.
(The m.file name is usually saved as the same as the function name)

Multiple Input

In "Editor"

```
function d = pyth3(a,b,c)
d=(a^2+b^2+c^2)^0.5;
```

In "Command Window"

```
>> a=pyth3(1,2,3)
a =
    3.7417
```

Sub-function(s)

1. In "Editor"

```
function y = mixfun1(x)
y=sin(s1(x))+sin(3*s1(x));

function w = s1(z)
w=1+z+z^2;
```

In "Command Window"

```
>> a=mixfun1(1)
a =
    0.5532
```

(`mixfun1` is main function, `s1` is sub-function.)

2. In "Editor"

```
function y = mixfun2(x)
y=pol1(pol2(x))-pol2(pol1(x));

function s = pol1(r)
s=1-r+r^2;

function s = pol2(r)
s=1+r-r^2;
```

In "Command Window"

```
>> a=mixfun2(-1)
a =
     8
```

(`mixfun2` is main function, `pol1` and `pol2` are sub-functions.)

Commands Table

Commands	Descriptions	Examples

`%`	comment / remark (not executed)	`function A = cyl_area(r,h)`
		`% A is total surface area of the cylinder`
		`% r is base radius, h is height`
		`A1=2*pi*r*h; % lateral surface area`
		`A2=pi*r^2;  % base area`
		`A=A1+2*A2;`

`...`	code separation	`function y = long_fun(a,b,c,d)`
		`y=(a+b+c+d)^2+(a-b+c+d)^2+(a+b...`
		`  -c+d)^2+(a+b+c-d)^2;`

4 Vector & Matrix

Quick Examples

Vector

```
>> a=[2 3 4 5 6]
a =
     2     3     4     5     6
```

Matrix

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

Commands Table

Commands	Descriptions	Results
`[2 3 4]` or `[2,3,4]`	row vector	$\begin{bmatrix}2 & 3 & 4\end{bmatrix}$
`[2;3;4]`	column vector	$\begin{bmatrix}2\\3\\4\end{bmatrix}$
`[2,3,4;3,5,6]`	matrix	$\begin{bmatrix}2 & 3 & 4\\3 & 5 & 6\end{bmatrix}$
`2:5`	row vector with step 1	$\begin{bmatrix}2 & 3 & 4 & 5\end{bmatrix}$
`2:0.5:5`	row vector with step 0.5	$\begin{bmatrix}2 & 2.5 & 3 & 3.5 & 4 & 4.5 & 5\end{bmatrix}$

In the following table, denote

`u=[2,3,4,5,6,7]`$= \begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 \end{bmatrix}$ and `A=[1,2,3;4,5,6;7,8,9]`$= \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$.

Commands	Descriptions	Results
`u(2)`	2-nd entry	3
`u(3:5)`	3-rd to 5-th entries	$\begin{bmatrix} 4 & 5 & 6 \end{bmatrix}$
`A(2,3)`	2-nd row 3-rd column entry	6
`A(2,1:2)`	2-nd row, 1-st to 2-nd column entries	$\begin{bmatrix} 4 & 5 \end{bmatrix}$
`A(2:3,1)`	2-nd to 3-rd row, 1-st column entries	$\begin{bmatrix} 4 \\ 7 \end{bmatrix}$
`A(2:3,1:2)`	2-nd to 3-rd row, 1-st to 2-nd column entries	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
`A(:,1)`	1-st column entries	$\begin{bmatrix} 1 \\ 4 \\ 7 \end{bmatrix}$
`A(3,:)`	3-rd row entries	$\begin{bmatrix} 7 & 8 & 9 \end{bmatrix}$

In the following table, denote `A=[2,3;1,2]`$= \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}$ and `B=[1,2;1,1]`$= \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}$.

Commands	Descriptions	Results
`A+B`	entry-wise addition	$\begin{bmatrix} 3 & 5 \\ 2 & 3 \end{bmatrix}$
`A-B`	entry-wise subtraction	$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$
`A*B`	matrix multiplication AB	$\begin{bmatrix} 5 & 7 \\ 3 & 4 \end{bmatrix}$
`A/B`	AB^{-1}, or solution to $XB = A$	$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$
`A\B`	$A^{-1}B$, or solution to $AX = B$	$\begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}$
`A^2`	matrix power A^2	$\begin{bmatrix} 7 & 12 \\ 4 & 7 \end{bmatrix}$
`A.*B`	entry-wise multiplication	$\begin{bmatrix} 2 & 6 \\ 1 & 2 \end{bmatrix}$
`A./B`	entry-wise division	$\begin{bmatrix} 2 & 1.5 \\ 1 & 2 \end{bmatrix}$
`A.^B`	entry-wise power	$\begin{bmatrix} 2 & 9 \\ 1 & 2 \end{bmatrix}$
`A'`	matrix conjugate transpose $A^* = (\bar{A})^\intercal$	$\begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix}$
`A.'`	matrix transpose $A^\intercal$	$\begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix}$

Commands	Descriptions	Examples

`zeros`	zero vector / matrix	1. `>> A=zeros(3)`
		`A =`
		`0 0 0`
		`0 0 0`
		`0 0 0`
		2. `>> A=zeros(2,3)`
		`A =`
		`0 0 0`
		`0 0 0`

`eye`	identity matrix	1. `>> A=eye(3)`
		`A =`
		`1 0 0`
		`0 1 0`
		`0 0 1`
		2. `>> A=eye(2,3)`
		`A =`
		`1 0 0`
		`0 1 0`

`diag`	matrix → vector of diagonal
	vector→ diagonal matrix
		1. `>> A=[2,3,1;4,0,2;1,1,3]`
		`A =`
		`2 3 1`
		`4 0 2`
		`1 1 3`
		`>> diag(A)`
		`ans =`
		`2`
		`0`
		`3`
		2. `>> a=[1,2,3]`
		`a =`
		`1 2 3`
		`>> diag(a)`
		`ans =`
		`1 0 0`
		`0 2 0`
		`0 0 3`

`ones`	"one" vector / matrix	1. `>> A=ones(3)`
		`A =`
		`1 1 1`
		`1 1 1`
		`1 1 1`
		2. `>> A=ones(2,3)`
		`A =`
		`1 1 1`
		`1 1 1`

`rand`	random number / vector / matrix with entries between 0 and 1	1. `>> rand()`
		`ans =`
		`0.3922`
		2. `>> rand(3)`
		`ans =`
		`0.6555 0.0318 0.0971`
		`0.1712 0.2769 0.8235`
		`0.7060 0.0462 0.6948`
		3. `>> rand(2,3)`
		`ans =`
		`0.3171 0.0344 0.3816`
		`0.9502 0.4387 0.7655`

`length`	number of entries in a vector	`>> a=[2,3,5,7,11]`
		`a =`
		`2 3 5 7 11`
		`>> length(a)`
		`ans =`
		`5`

`size`	`size(A,1)`: number of rows
	`size(A,2)`: number of columns
		`>> A=[1,3,5;2,4,6]`
		`A =`
		`1 3 5`
		`2 4 6`
		`>> size(A,1)`
		`ans =`
		`2`
		`>> size(A,2)`
		`ans =`
		`3`

5 If, For, While

Quick Examples

In "Editor" In "Command Window"

```
a=rand()
if a<0.5
    disp('below 0.5')
else
    disp('above 0.5')
end
```

copy & paste to "Command Window" →

```
>> a=rand()
if a<0.5
    disp('below 0.5')
else
    disp('above 0.5')
end
a =
    0.2551
below 0.5
```

```
a=0;
for i=1:10
    a=a+i;
end
disp(a)
```

copy & paste to "Command Window" →

```
>> a=0;
for i=1:10
    a=a+i;
end
disp(a)
    55
```

```
a=1;
while a>0.001
    a=a/2;
end
disp(a)
```

copy & paste to "Command Window" →

```
>> a=1;
while a>0.001
    a=a/2;
end
disp(a)
   9.7656e-04
```

Commands Table

Commands	Descriptions
`&&`	and
`||`	or
`<`, `<=`	less than, less than or equal to
`>`, `>=`	greater than, greater than or equal to
`~=`	not equal to
`break`	exit from "for", "while"
ctrl+c	terminate execution

1 branch: `if...end`

e.g.

```
c=0;
a=rand();
if a>0.4 && a<0.6
    c=1;
end
disp([c,a])
```

⟶

```
>> c=0;
a=rand();
if a>0.4 && a<0.6
    c=1;
end
disp([c,a])
       0   0.9058
```

2 branches: `if...else...end`

e.g.

```
a=floor(6*rand());
if a==1 || a==3 || a==5
    fprintf('%1.0f is odd \n',a)
else
    fprintf('%1.0f is even \n',a)
end
```

⟶

```
>> a=floor(6*rand());
if a==1 || a==3 || a==5
    fprintf('%1.0f is odd \n',a)
else
    fprintf('%1.0f is even \n',a)
end
2 is even
```

3 or more branches: `if...elseif...else...end`

e.g.

```
x=rand();
if x<=1/3
    y=3*x;
elseif x<=2/3
    y=1;
else
    y=3*(1-x);
end
fprintf('f(x) = %4.2f\n',y)
```

⟶

```
>> x=rand();
if x<=1/3
    y=3*x;
elseif x<=2/3
    y=1;
else
    y=3*(1-x);
end
fprintf('f(x) = %4.2f\n',y)
f(x) = 0.43
```

Example 5.1

```
function n=min_n(b)
% Given that a_(n+1) = a_n + 1/a_n
% find minimum n such that a_n >= b
% stop when n > 1000
n=1;
a(n)=1;
while a(n)<b
    a(n+1)=a(n)+1/a(n);
    n=n+1;
    if n==1001
        disp('n>1000')
        break
    end
end
disp(a(n))
```

⟶

```
>> min_n(10)
ans =
    50
>> m=min_n(20)
m =
    199
>> m=min_n(50)
n>1000
m =
    1001
```

Example 5.2

```
function [a,b]=sol_py(c)
% find integer pair (a,b) such that a^2+b^2=c
% a,b are between 0 and 9
d=0; % change to 1 when solution is found
for i=1:9
    for j=1:9
        if i^2+j^2==c
            a=i;b=j;d=1;
            break; % exit from 'for' loop of i
        end
    end
    if d==1
        break % exit from 'for' loop of i
    end
end
if d~=1
    a=0;b=0;
    disp('no solution between 0 and 9')
end
```

→

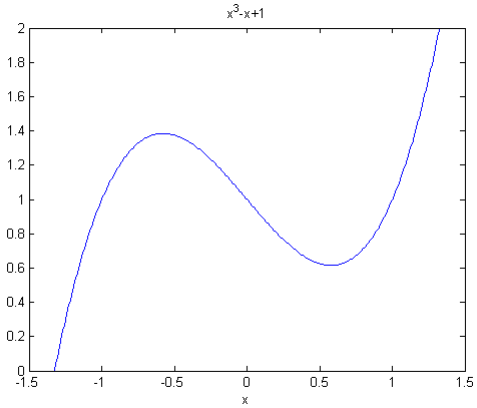
```
>> [a,b]=sol_py(10)
a =
    1
b =
    3
>> [x,y]=sol_py(20)
x =
    2
y =
    4
>> [m,n]=min_n(30)
no solution between 0 and 9
m =
    0
n =
    0
```

6 Graph

Quick Examples

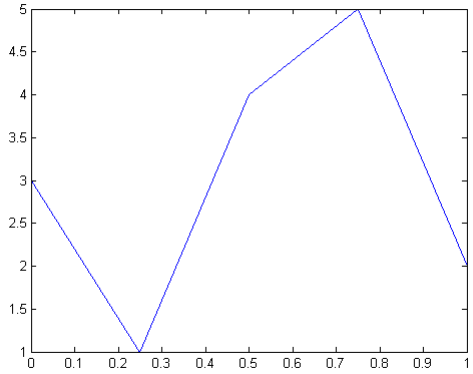
```
>> ezplot('x.^3-x+1',[-1.5,1.5,0,2])
```

→




```
>> x=0:0.25:1;
>> y=[3,1,4,5,2];
>> plot(x,y)
```

⟶



Commands Table

Commands	Descriptions	Examples (in "Editor")
`ezplot`	plot function	1. `ezplot('x.^2+1')` 2. `ezplot('x.^2+1',[-2,2,0,5])`

`plot`	plot points	`x=-2:0.1:2;`
		`y=x.^2+1;`
		`plot(x,y)`

`subplot`	plot multiple graphs	`subplot(1,2,1),ezplot('x.^2+1')`
		`x=-2:0.1:2;y=x.^2+1;`
		`subplot(1,2,2),plot(x,y)`

`grid on`	display grid	`ezplot('x.^2+1')` `grid on`
`axis`	limits of coordinates	`ezplot('x.^2+1')` `axis([-2,2,0,5])`
`title`	graph title	`ezplot('x.^2+1')` `title('function 1')`

`xlabel`
`ylabel`
`zlabel`
	labels of coordinates	`ezplot('x.^2+1')`
		`xlabel('length')`
		`ylabel('height')`

`legend`	labels of functions	`x=-2:0.1:2;`
		`y1=x.^2+1;`
		`y2=2-x.^2;`
		`plot(x,y1,x,y2)`
		`legend('fun 1','fun 2')`

`plot3`	plot 3D curve	`t=0:0.1:12` `plot3(cos(t),sin(t),t)`

`mesh`	plot 3D "net" surface	`t=-1:0.1:1;`
		`[x,y]=meshgrid(t);`
		`z=sin(x.*y);`
		`mesh(x,y,z)`

`meshc`	plot 3D "net" surface with contour	`t=-1:0.1:1;`
		`[x,y]=meshgrid(t);`
		`z=sin(x.*y);`
		`meshc(x,y,z)`

`surf`	plot 3D "smooth" surface	`t=-1:0.1:1;`
		`[x,y]=meshgrid(t);`
		`z=sin(x.*y);`
		`surf(x,y,z)`

`surfc`	plot 3D "smooth" surface with contour	`t=-1:0.1:1;`
		`[x,y]=meshgrid(t);`
		`z=sin(x.*y);`
		`surfc(x,y,z)`

Styles of `plot`

Colours	Lines	Points
`y` yellow	`-` solid line (default)	`+` plus sign
`m` magenta	`--` dashed line	`o` circle
`c` cyan	`:` dotted line	`*` asterisk
`r` red	`-.` dash-dot line	`.` point
`g` green		`x` cross
`b` blue		`s` square
`w` white		`d` diamond
`k` black		

Example 6.1

```
x=-2:0.1:2;
y=x.^2+1;
subplot(2,3,1),plot(x,y,'g'),title('g'),legend('1')
subplot(2,3,2),plot(x,y,'g--'),title('g--'),legend('2')
subplot(2,3,3),plot(x,y,'g--o'),title('g--o'),legend('3')
subplot(2,3,4),plot(x,y,'--'),title('--'),legend('4')
subplot(2,3,5),plot(x,y,'o'),title('o'),legend('5')
subplot(2,3,6),plot(x,y,'--o'),title('--o'),legend('6')
```