

# MATLAB CHEAT SHEET

Throughout this document  $x$  and  $y$  will be either row or column vectors and  $A$  will always be a matrix. In all cases the entries are real numbers.

## Basics

<code>clc</code>	Clear command window
<code>clear</code>	Clear all variables
<code>clf</code>	Clear all plots
<code>close all</code>	Close all plots
<code>doc function</code>	Open help page for function
<code>who</code>	list variables on workspace
<code>% This is a comment</code>	Comments
<code>ctrl-c</code>	Abort the current operation. On mac $\mathbb{H}$ + . (the Command key and the period key) will also abort.
<code>format short</code>	Display 4 decimal places
<code>format long</code>	Display 15 decimal places
<code>disp('text')</code>	Print text

## Defining and Changing Variables

<code>a = 3</code>	Define variable $a$ to be 3
<code>b = 3e-5</code>	Define variable $b$ to be $0.00003 = 3e-5$
<code>x = [1, 2, 3]</code>	Set $x$ to be the row vector $[1, 2, 3]$
<code>x = [1; 2; 3]</code>	Set $x$ to be the column vector $[1, 2, 3]^T$
<code>A = [1, 2, 3, 4; 5, 6, 7, 8; 9, 10, 11, 12]</code>	Set $A$ to be a $3 \times 4$ matrix
<code>x(2) = 7</code>	Change $x$ from $[1, 2, 3]$ to $[1, 7, 3]$
<code>A(2,1) = 0</code>	Change $A_{2,1}$ from 5 to 0

## Basic Arithmetic and Functions

<code>3*4, 7+4, 2-6, 8/3</code>	multiply, add, subtract and divide
<code>3^7</code>	Compute $3^7$
<code>sqrt(5)</code>	Compute $\sqrt{5}$
<code>log(3)</code>	Compute $\ln(3)$
<code>log10(100)</code>	Compute $\log_{10}(100)$
<code>exp(3.5)</code>	Compute $e^{3.5}$
<code>abs(-5)</code>	Compute $ -5 $
<code>sin(5*pi/3)</code>	Compute $\sin(5\pi/3)$
<code>floor(3.8)</code>	Compute $\lfloor 3.8 \rfloor$
<code>ceil(3.8)</code>	Compute $\lceil 3.8 \rceil$

## Constructing Matrices and Vectors

<code>zeros(12, 5)</code>	Make a $12 \times 5$ matrix of zeros
<code>ones(12, 5)</code>	Make a $12 \times 5$ matrix of ones
<code>A = []</code>	make $A$ an empty matrix
<code>eye(5)</code>	Make a $5 \times 5$ identity matrix
<code>eye(12, 5)</code>	Make a $12 \times 5$ identity matrix
<code>linspace(1.4, 6.3, 1004)</code>	Make a row vector with 1004 elements evenly spaced between 1.4 and 6.3
<code>logspace(1.4, 6.3, 1004)</code>	Make a row vector with 1004 elements that are logarithmically spaced between $10^{1.4}$ and $10^{6.3}$
<code>7:15</code>	Row vector of 7, 8, ..., 14, 15
<code>7:2:15</code>	Row vector of 7, 9, 11, ..., 15
<code>4:-2:-2</code>	Row vector of 4, 2, ..., -2

## Operations on Matrices and Vectors

<code>3 * x</code>	Multiply every element of $x$ by 3
<code>x + 2</code>	Add 2 to every element of $x$
<code>x + y</code>	Element-wise addition of two vectors $x$ and $y$
<code>A * y</code>	Product of a matrix and vector
<code>A * B</code>	Product of two matrices
<code>A .* B</code>	Element-wise product of two matrices
<code>A ^ 3</code>	Square matrix $A$ to the third power
<code>A .^ 3</code>	Every element of $A$ to the third power
<code>cos(A)</code>	Compute the cosine of every element of $A$
<code>abs(A)</code>	Compute the absolute values of every element of $A$
<code>A'</code>	Transpose of $A$
<code>inv(A)</code>	Compute the inverse of $A$
<code>det(A)</code>	Compute the determinant of $A$
<code>[eVecs, eVals]=eig(A)</code>	Compute the eigenvectors (eVecs) and eigenvalues (eVals) of $A$
<code>[rows, cols] = size(A)</code>	Get the size of $A$
<code>numel(A)</code>	Get the number of elements of $A$

## Constants

<code>pi</code>	$\pi = 3.141592653589793$
<code>i</code>	Imaginary unit $\sqrt{-1}$
<code>j</code>	Imaginary unit $\sqrt{-1}$
<code>NaN</code>	Not a number (i.e. 0/0)
<code>Inf</code>	Infinity
<code>realmax</code>	Largest positive floating-point number $1.7977 \cdot 10^{308}$
<code>realmin</code>	Smallest positive floating-point number $2.2251 \cdot 10^{-308}$

## Random Numbers

<code>rand(m,n)</code>	returns an $m$ by $n$ matrix of random numbers sampled from a standard uniform distribution $[0, 1]$
<code>randn(m,n)</code>	returns an $m$ by $n$ matrix of random numbers sampled from a standard Normal distribution with $\mu = 0$ and $\sigma^2 = 1$

## Slicing portions of Matrices or Vectors

<code>x(2:12)</code>	The 2 <sup>nd</sup> to the 12 <sup>th</sup> elements of $x$
<code>x(2:end)</code>	The 2 <sup>nd</sup> to the last elements of $x$
<code>x(1:3:end)</code>	Every third element of $x$ from the first to last
<code>A(5,:)</code>	Get the 5 <sup>th</sup> row of $A$
<code>A(:,5)</code>	Get the 5 <sup>th</sup> column of $A$
<code>A(5, 1:3)</code>	Get the first to third elements in the 5 <sup>th</sup> row

## Array concatenation

<code>A = [x;y]</code>	Concatenate arrays $x$ and $y$ vertically. $x$ and $y$ must have the same number of columns
<code>A = [x y]</code>	Concatenate arrays $x$ and $y$ horizontally. $x$ and $y$ must have the same number of rows

## Summary statistics and others

<code>mean(x)</code>	returns the mean value of the entries in vector $x$
<code>mean(A,1)</code>	returns a row vector containing the mean of each column of $A$
<code>mean(A,2)</code>	returns a column vector containing the mean of each row of $A$
<code>var(x)</code>	returns the variance of the entries in vector $x$
<code>var(A,[],1)</code>	returns a row vector containing the variance of each column of $A$
<code>var(A,[],2)</code>	returns a column vector containing the variance of each row of $A$
<code>min(x)</code>	returns the minimum value of the entries in vector $x$
<code>[minVal,idx] =min(A,[],1)</code>	<code>minVal</code> returns a row vector with the minimum value of each column of $A$ . <code>idx</code> returns a row vector with the indices at which the minimum of each column occurs. For <code>max</code> the syntax is the same.
<code>[minVal,idx] =min(A,[],2)</code>	<code>minVal</code> returns a column vector with the minimum value of each row of $A$ . <code>idx</code> returns a column vector with the indices at which the minimum of each row occurs. For <code>max</code> the syntax is the same.
<code>diff(x)</code>	if $x$ is a vector with $N$ elements, this command returns a vector with $N - 1$ elements as follows: $x_2 - x_1, x_3 - x_2, \dots, x_N - x_{N-1}$

## Plotting

<code>figure</code>	Create new empty figure window
<code>subplot(m,n,p)</code>	divides the current figure into an $m$ -by- $n$ grid and creates axes in the position specified by $p$ (MATLAB counts row-wise for subplot)
<code>plot(x,y)</code>	Plot $y$ versus $x$ (must be the same length)
<code>scatter(x,y)</code>	scatter plot of $y$ versus $x$ .
<code>loglog(x,y)</code>	Plot $y$ versus $x$ on a log-log scale (both axes have a logarithmic scale)
<code>semilogx(x, y)</code>	Plot $y$ versus $x$ with $x$ on a log scale
<code>semilogy(x, y)</code>	Plot $y$ versus $x$ with $y$ on a log scale
<code>bar(x, y)</code>	Bar chart. Draw bars of height $y$ at locations specified by $x$
<code>xlim([xmin xmax])</code>	Force the $x$ axis to be scaled between $x_{\min}$ and $x_{\max}$
<code>ylim([ymin ymax])</code>	Force the $y$ axis to be scaled between $y_{\min}$ and $y_{\max}$
<code>axis equal</code>	Force the $x$ and $y$ axes to be scaled equally
<code>title('A Title')</code>	Add a title to the plot
<code>xlabel('x label')</code>	Add a label to the $x$ axis
<code>ylabel('y label')</code>	Add a label to the $y$ axis
<code>legend('foo', 'bar')</code>	Label 2 curves for the plot
<code>grid</code>	Add a grid to the plot
<code>hold on</code>	Multiple plots on single figure
<code>hold off</code>	Release hold on current figure

## Loading Files

<code>load('data.mat')</code>	Load variables from file into workspace
-------------------------------	---

## Logicals

```
a = 10; % Assign a the value of 10
a == 5 % Test if a is equal to 5
false
a == 10 % Test if a is equal to 10
true
a >= 5 % Test if a is greater than or equal to 5
true
a < 11 % Test if a is less than 11
true
a ~= 4 % Test if a is not equal to 4
true
a > 1 && a ~= 10 % Test if a is greater than 1 AND
false % not equal to 10
a > 1 || a ~= 10 % Test if a is greater than 1 OR
true % not equal to 10
```

## For loops

```
for k = 1:5
    disp(k);
end
```

## Conditional Statements

```
if a > 10
    disp('Greater than 10');
elseif a == 5
    disp('a is 5');
else
    disp('Neither condition met');
end
```

## While loops

```
k = 0;
while k < 7
    k = k + 1;
end
```

## Functions

```
function output = addNumbers(x, y)
    output = x + y;
end

addNumbers(10, -5)
5
```

## Function Handles

```
f = @ (x) sin(x.^2)./(5*x);

f(pi/2)
0.0795
f([-pi/2, 0, pi/2])
-0.0795 NaN 0.0795
```

## Plotting

```
x = linspace(-3*pi, 3*pi, 1000);
y1 = sin(x);
y2 = cos(x);

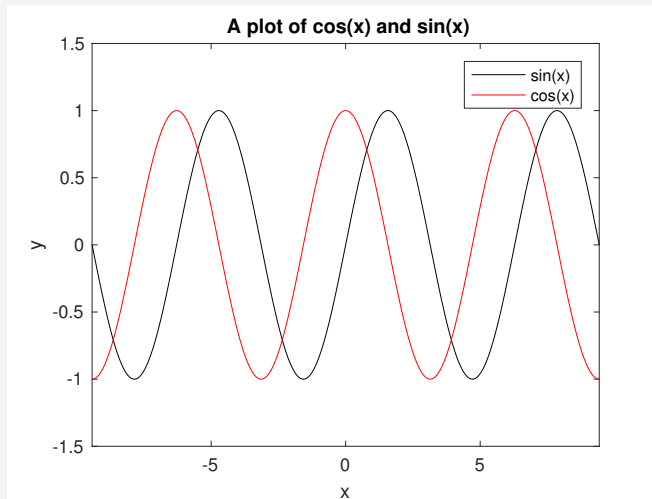
figure;
plot(x, y1, 'k-'); % Plot sin(x) as a black line
hold on           % Now we can add another curve
plot(x, y2, 'r-'); % Plot cos(x) as a red line
hold off

% Set the axis limits
xlim([-3*pi, 3*pi]);
ylim([-1.5, 1.5]);

% Add axis labels
xlabel('x');
ylabel('y');

% Add a title
title('A plot of cos(x) and sin(x)');

% Add a legend
legend('sin(x)', 'cos(x)');
```



## Plotting with subplots

```
x = linspace(-3*pi, 3*pi, 1000);
y1 = sin(x);
y2 = cos(x);

figure;

% First panel
subplot(2,1,1)
plot(x, y1, 'k-'); % Plot sin(x) as a black line
hold on           % Now we can add another curve
plot(x, y2, 'r-'); % Plot cos(x) as a red line
hold off
% Set the axis limits
xlim([-3*pi, 3*pi]);
ylim([-1.5, 1.5]);

% Add axis labels
xlabel('x');
ylabel('y');

% Add a title
title('A plot of cos(x) and sin(x)');

% Add a legend
legend('sin(x)', 'cos(x)');

subplot(2,1,2)
scatter(y1,y2);
axis equal;
title('scatter plot of sine vs. cosine');
xlabel('sine');
ylabel('cosine');
```

