MATLAB CHEAT SHEET - ENGINEERED-MIND.COM

Small variables like x and y will be either row or column vectors and A will always be a matrix.

Basic Commands

`clc`	Clear command window
`clear (all)`	Clear all variables
`close all`	Close all plots
`clf`	Clear all plots
`doc command`	Extensive help page for command
`help command`	Quick help page for command
`%This is a comment`	Indicates a comment
`a = 5;`	Semicolon suppresses output
`whos`	Lists all variables defined
`disp('text')`	Print text
`save 'file.mat'`	Save variables to file.mat
`load 'file.mat'`	Load variables from file.mat
`diary on`	Record input/output to file diary

Keyboard Shortcuts

`F1`	Help/documentation for highlighted function
`F5`	Run Code
`F9`	Run highlighted code
`F10`	Run code line
`F11`	Run code line, enter functions
`F12`	Insert break point
`Ctrl-D`	Open highlighted codes file
`Ctrl-R`	Comment code
`Ctrl-T`	Uncomment code
`Ctrl-N`	Open new script
`Ctrl-W`	Close script
`Ctrl-C`	Abort operation

Entries of Matrices and Vectors

`abs(x)`	The absolute value of x
`eps`	Floating point accuracy
`1e6`	10^6
`sum(x)`	Sums elements in x
`round`	Rounds to the nearest integer
`ceil`	Rounds to nearest integer greater than or equal to that element
`fix`	Rounds to the nearest integer toward zero
`floor`	Rounds to the nearest integer less than or equal to that element

Cell Manipulation

`x = cell(a, b)`	a × b cell array
`x{n,m}`	Access cell element n,m
`cell2mat(x)`	Transforms cell to matrix

Manipulation of Variables

`a = 500`	Define variable a to be 500
`x = [3, 1, 4]`	Set x to be the row vector $[3, 1, 4]$
`x = [3; 1; 4]`	Set x to be the column vector $[3, 1, 4]^T$

`A = [3, 1, 4, 1;`
`5, 9, 2, 6;`
`5, 3, 5, 8]`
	Set A to be a 3×4 matrix

`x(2) = 7`	Change x from $[3, 1, 4]$ to $[3, 7, 4]$
`A(2,1) = 0`	Change $A_{2,1}$ from 5 to 0

Basic Arithmetic and Functions

`1*2, 3+4, 5-6, 7/8`	Multiply, add, subtract and divide
`2^8`	Compute 2^8
`sqrt(16)`	Compute $\sqrt{16}$
`log(5)`	Compute $\ln(5)$
`log10(100)`	Compute $\log_{10}(100)$
`abs(-10)`	Compute $\lvert -10 \rvert$
`sin(2*pi/6)`	Compute $\sin(2\pi/6)$
`ceil(3.8)`	Outputs 4.0
`floor(3.8)`	Outputs 3.0

Construct Matrices and Vectors

`zeros(5, 5)`	Create a 5×5 matrix of zeros (Pre-Allocation!)
`ones(6, 7)`	Create a 6×7 matrix of ones
`eye(3)`	Create a 3×3 identity matrix
`eye(9, 10)`	Make a 9×10 identity matrix
`linspace(X, Y, N)`	Generates N points between X and Y
`logspace(X1, Y1, N1)`	Creates a vector with N1 elements where the log of the spacing is evenly increasing between X1 and Y1
`1:99`	Row vector of $1, 2, \ldots, 88, 99$

Operations on Matrices and Vectors

`x + 5`	Add 5 to every element of x
`x + y`	Elementwise addition of two vectors x and y
`10 * x`	Multiply every element of x by 10
`A * y`	Product of a matrix and vector
`A * B`	Product of two matrices
`A .* B`	Element-wise product of two matrices (Important if matrices are not quadratic)
`A ^ 4`	Square matrix A to the fourth power
`A .^ 4`	Every element of A to the fourth power
`cos(A)`	Compute the cosine of every element of A
`abs(A)`	Compute the absolute values of every element of A
`A'`	Transpose of A
`det(A)`	Compute the determinant of A
`size(A)`	Get the size of A

MATLAB CHEAT SHEET - MODELING AND SIMULATION

Plotting

`plot(x,y)`	Plot y versus x (same dimension!)
`loglog(x,y)`	Plot y versus x on a log-log scale (both axes have a log scale)
`semilogx(x, y)`	Plot y versus x with x on a log scale
`semilogy(x, y)`	Plot y versus x with y on a log scale
`axis equal`	Force the x and y axes to be scaled equally
`title('A Title')`	Add a title to the plot
`xlabel('x text')`	Add a label to the x axis
`ylabel('y text')`	Add a label to the y axis
`legend('foo', 'bar')`	Label 2 curves for the plot
`grid on/off`	Add a grid to the plot
`figure`	Start a new plot
`figure(i)`	i^{th} figure
`subplot(a,b,c)`	For multiple figures in one plot
`hold on`	Retains current figure when adding new stuf
`hold off`	Default settings (no hold on!)
`set(fig1, 'LineWidth', 2)`	Change line width
`set(fig1, 'LineStyle', '-')`	Change dot marker
`set(fig1, 'Marker', '.')`	Change marker type
`set(fig1, 'MarkerSize', 10)`	Change marker size
`set(fig1, 'FontSize', 14)`	Change font size

Entries of Matrices and Vectors

`x(5:8)`	The 5^{th} to the 8^{th} elements of x
`x(5:end)`	The 5^{th} to the last elements of x
`x(1:2:end)`	Every second element of x from the first to last
`A(3,:)`	Get the third row of A
`A(:,5)`	Get the 5^{th} column of A
`A(5, 2:5)`	Get the first to fifth elements in the 5^{th} row

Constants

`pi`	$\pi = 3.1415926533$
`Inf`	Infinity
`NaN`	Not a number (i.e. $0/0$)
`realmax`	Largest positive floating-point number $1.7977 \cdot 10^{308}$
`realmin`	Smallest positive floating-point number $2.2251 \cdot 10^{-308}$

Debugging

`tic`	Starts timer
`toc`	Stops timer
`try/catch`	Good to track errors
`dbclear`	Clears breakpoints
`break`	Terminates execution of for/while loop

Data Import & Export

`xlsread/xlswrite`	Spreadsheets (.xls,.xlsm)
`load/save -ascii`	Text files (txt,csv)
`load/save`	Matlab Files (.m)
`imread/imwrite`	Image Files

Solving linear equations

`inv(A)`	Compute the inverse A^{-1}
`eig(A)`	Compute the eigenvalues of A
`[L,U,P] = lu(A)`	The LU factorization $PA = LU$
`[V,D] = eig(A)`	V are the eigenvectors of A, and the diagonals $diag(D)$ are the eigenvalues of A
`A\b`	Compute the solution x to $Ax = b$

MATLAB CHEAT SHEET - MODELING AND SIMULATION

Logicals

```
a = 20;  % Assign a the value of 10
a == 5   % Test if a is equal to 5
    false
a == 20  % 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
    false
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 ~= 5 % Test if a is greater than 1 OR
    true         % not equal to 10
```

For loops

```
for k = 1:10
    disp(k);
end
```

Conditional Statements

```
if a > 90
    disp('Greater than 90');
elseif a == 90
    disp('a is 90');
else
    disp('None of the conditions is mets');
end
```

While loops

```
k = 0;
while k < 5
    k = k + 1;
end
```

Functions

```
function [a, b] = testfct(x, y)
    a = x + y;
    b = x * y;
end

testfct(2, 3) %Call function in script or command window
```

Function Handles

```
sqr = @(n) n.^2;

x = sqr(3) %Outputs 9
```

Plotting & Subplot

```
x = linspace(-5*pi, 5*pi, 1000);
y1 = sin(x);
y2 = cos(x);

plot(x, y1, 'g-', 'LineWidth',3);  % Plot black sin(x) curve
hold on                            % Adding additional curve
plot(x, y2, 'r-', 'LineWidth',3);  % Plot red cos(x) curve
grid on
set(gca,'fontsize',20)

% Set the axis limits
axis([-5*pi, 5*pi, -1.5, 1.5])

% Add axis labels
xlabel('x', 'FontSize',20);
ylabel('y', 'FontSize',20);

% Add a title
title('A plot of cos(x) and sin(x)', 'FontSize', 20);

% Add a legend
legend('sin(x)', 'cos(x)');
-----
% Code for Subplots

x = linspace(0,10,50);
y = rand(50,1);

subplot(2,2,1), plot(x,sin(x),'Color','red','LineWidth',3)
set(gca,'fontsize',14)
axis([0,2*pi,-1,1]), axis square

subplot(2,2,2), plot(x,cos(x),'Linewidth',3,'Color','blue')
set(gca,'fontsize',14)
axis([0,2*pi,-1,1]), axis square

subplot(2,2,3:4)
y2 = rand(50,1); plot(x,y2,'LineWidth',3)
set(gca,'fontsize',14)
```