

MATLAB Reference Card

Command Line

The command line is where you directly run commands and interact with MATLAB. When writing scripts, use the command line to develop and test commands and then copy them into your script when you are satisfied with how they run.

Run a command and display the result	<code>command<enter></code>
Run a command and hide the result	<code>command;<enter></code>
Move through command history	Up/Down Arrows

Finding Help

MATLAB has a very comprehensive help system. MATLAB Help has detailed explanations and examples of how to use commands. The help command provides a quick synopsis on how to use each command.

Open MATLAB Help	F1
List help topics	help
Show help for a command	help <i>command</i>

Variables

In MATLAB, you declare a new variable by assigning a value to it. Variables declared in the command line and in scripts are part of the global workspace. Variables declared in functions are local to the function and disappear when the function finishes.

Assign a value to a variable	<code>x = 2</code>
Display the value of a variable	<code>x<enter></code>
Show all defined variables	whos
Erase a variable	clear <i>x</i>

Create a Matrix

Create a 2x2 matrix	<code>x = [1 4; 5 6]</code>
Create a 6x4 matrix filled with 1	<code>x = ones(6,4)</code>
Create a 3x8 matrix filled with 0	<code>x = zeros(3,8)</code>
Create a 8x8 identity matrix	<code>x = eye(8)</code>
Create a 2x1 column vector	<code>x = [1;2]</code>
Create a 1x2 row vector	<code>x = [1 2]</code>
Create [1 2 3 4 5]	<code>x = 1:5</code>
Create [1 2 3 4 5]	<code>x = 1:1:5</code>
Create [4 6 8 10]	<code>x = 4:2:10</code>
Create [8 7 6 5]	<code>x = 8:-1:5</code>
Create [8 7 6 5]	<code>x = linspace(8,5,4)</code>
Create logarithmically spaced vector	<code>x = logspace(1,5)</code>

Indexing a Matrix

Referring to particular element or set of elements of a matrix is called indexing. Indexing can be used to read a value from a matrix or to store a new value.

Element at 1,3	<code>x(1,3)</code>
2nd Row	<code>x(2,:)</code>
3rd Column	<code>x(:,3)</code>
5-10 elements	<code>x(5:10)</code>
2,5,10 elements	<code>x[2 5 10]</code>

Matrix Operations

Note that since vectors are also matrices, most matrix operations can also be used on vectors.

Size of a matrix	<code>size(x)</code>
Largest dimensions of a matrix	<code>length(x)</code>
Transpose	<code>transpose(x)</code>
Transpose - Shortcut	<code>x'</code>
Matrix Inverse	<code>inv(x)</code>
Matrix Addition	<code>x + y</code>
Matrix Subtraction	<code>x - y</code>
Matrix Multiplication	<code>x * y</code>
Matrix Exponentiation	<code>x ^ y</code>
Component-wise Multiplication	<code>x .* y</code>
Component-wise Division	<code>x ./ y</code>
Component-wise Exponentiation	<code>x .^ y</code>

Vector Operations

These operations are designed to work with vectors, which are 1xN or Nx1 matrices.

Norm (length) of a vector	<code>norm(x)</code>
Vector Dot product	<code>dot(x, y)</code>
Vector Cross product	<code>cross(x, y)</code>
Sum the components of a vector	<code>sum(x)</code>
Cumulative sum of vector elements	<code>cumsum(x)</code>
Subtract each component from the next	<code>diff(x)</code>

Math Functions

These functions can be used on single numbers, vectors, or matrices. On vectors and matrices, the functions are applied to the individual components of the variable.

Square Root	<code>sqrt(x)</code>
Exponential	<code>exp(x)</code>
Natural Logarithm	<code>log(x)</code>
Base-10 Logarithm	<code>log10(x)</code>
Determine the sign of a number	<code>sign(x)</code>

Trig Functions

The following functions take or return radians. For degrees, append a “d” to the end of the function name.

Sine	<code>sin(x)</code>
Cosine	<code>sin(x)</code>
Tangent	<code>tan(x)</code>
Inverse Sine	<code>asin(x)</code>
Inverse Cosine	<code>acos(x)</code>
Inverse Tangent	<code>atan(x)</code>
Inverse Tangent	<code>atan(x,y)</code>

WAVE Files

Open a wave file	<code>[y,Fs] = wavread('filename')</code>
Save a wave file	<code>wavwrite(y,Fs,'filename')</code>
Play a sound	<code>sound(y, Fs)</code>

Plotting

Plotting commands affect the active plot. This is typically the most recently created plot. If no plot is open, plotting commands will create a new window and use it.

Plot y vs. t	<code>plot(t, y)</code>
Plot y vs. t and x vs. t	<code>plot(t, y, t, x)</code>
Plot with a logarithmic x axis	<code>semilogx(f,A)</code>
Plot with a logarithmic y axis	<code>semilogy(t,A)</code>
Plot with logarithmic x,y axes	<code>loglog(x,y)</code>
Create and activate a new plot window	figure
Turn on the grid on a plot	grid on
Turn off the grid on a plot	grid off
Set the x axis limits	<code>xlim([-10 10])</code>
Set the y axis limits	<code>ylim([-10 10])</code>
Set the x and y axis limits	<code>axis([xmin xmax ymin ymax])</code>

You can control the color and line type of each plotted line by specifying a line format after the y-value. Line formats can be combined into a single string.

Red	<code>'r'</code>
Blue	<code>'b'</code>
Green	<code>'g'</code>
Cyan	<code>'c'</code>
Yellow	<code>'y'</code>
Dotted Line	<code>':'</code>
Dashed Line	<code>'--'</code>
Dash-dot Line	<code>'-.'</code>
Plot a blue line	<code>plot(t, y, 'b')</code>
Plot a dotted green line	<code>plot(t, y, 'g:')</code>
Plot green and red lines	<code>plot(t, y, 'g', t, x, 'r')</code>

Label your plots for clarity.

X Axis Label	<code>xlabel('Time (s)')</code>
Y Axis Label	<code>ylabel('Amplitude (V)')</code>
Plot Title	<code>title('Observed Output')</code>
Add a plot legend	<code>legend('Output', 'Input')</code>

You can put multiple plots in a single window with the `subplot` command. For example, to make two plots stacked on top of each other:

<code>subplot(2,1,1)</code>	Select first plot area
<code>plot(t, y)</code>	Create the plot
<code>title('Output Voltage')</code>	Label the plot
<code>subplot(2,1,2)</code>	Select the second plot area
<code>plot(t, x)</code>	Create the plot
<code>title('Input Voltage')</code>	Label the plot

February 2007 v1.0. Copyright © 2007 Elliot Johnson

Permission is granted to make and distribute copies of this card provided the copyright notice and this permission notice are preserved on all copies.

Contact me for comments/concerns: erjohnso@colorado.edu