MATLAB Commands and Functions

Dr. Brian Vick
Mechanical Engineering Department
Virginia Tech

General Purpose Commands
Operators and Special Characters / 3
Commands for Managing a Session / 3
Special Variables and Constants / 4
System and File Commands / 4

Input/Output and Formatting Commands
Input/Output Commands / 5
Format Codes for fprintf and fscanf / 5
Numeric Display Formats / 5

Vector, Matrix and Array Commands
Array Commands / 6
Special Matrices / 6
Matrix Arithmetic / 6
Matrix Commands for Solving Linear Equations / 6
Cell Array Functions / 7
Structure Functions / 7

Plotting Commands
Basic xy Plotting Commands / 8
Plot Enhancement Commands / 8
Specialized Plot Commands / 8
Colors, Symbols and Line Types / 9
Three-Dimensional Plotting Commands / 9
Histogram Functions / 9

Programming

Logical and Relational Operators / 10
Program Flow Control / 10
Logical Functions / 10
M-Files / 11
Timing /11

Mathematical Functions

Exponential and Logarithmic Functions / 12
Trigonometric Functions / 12
Hyperbolic Functions / 12
Complex Functions / 13
Statistical Functions / 13
Random Number Functions / 13
Numeric Functions / 13
String Functions / 13

Numerical Methods

Polynomial and Regression Functions / 14
Interpolation Functions / 14
Numerical Integration Functions / 14
Numerical Differentiation Functions / 14
ODE Solvers / 15
Predefined Input Functions / 15

Symbolic Math Toolbox

Functions for Creating and Evaluating Symbolic Expressions / 16
Functions for Manipulating Symbolic Expressions / 16
Symbolic Calculus Functions / 16
Symbolic Solution of Algebraic and Transcendental Equations / 17
Symbolic Solution of Differential Equations / 17
Laplace Transform Functions / 17
Symbolic Linear Algebra Functions / 17

General Purpose Commands

Operators and Special Characters

+	Plus; addition operator.
-	Minus; subtraction operator.
*	Scalar and matrix multiplication operator.
.*	Array multiplication operator.
^	Scalar and matrix exponentiation operator.
.^	Array exponentiation operator.
\	Left-division operator.
/	Right-division operator.
.\	Array left-division operator.
./	Array right-division operator.
:	Colon; generates regularly spaced elements and represents an entire row or column.
( )	Parentheses; encloses function arguments and array indices; overrides precedence.
[ ]	Brackets; enclosures array elements.
.	Decimal point.
...	Ellipsis; line-continuation operator.
,	Comma; separates statements and elements in a row.
;	Semicolon; separates columns and suppresses display.
%	Percent sign; designates a comment and specifies formatting.
_	Quote sign and transpose operator.
._	Nonconjugated transpose operator.
=	Assignment (replacement) operator.

Commands for Managing a Session

clc	Clears Command window.
clear	Removes variables from memory.
exist	Checks for existence of file or variable.
global	Declares variables to be global.
help	Searches for a help topic.
lookfor	Searches help entries for a keyword.
quit	Stops MATLAB.
who	Lists current variables.
whos	Lists current variables (long display).

Special Variables and Constants

`ans`	Most recent answer.
`eps`	Accuracy of floating-point precision.
`i,j`	The imaginary unit $\sqrt{-1}$.
`Inf`	Infinity.
`NaN`	Undefined numerical result (not a number).
`pi`	The number π.

System and File Commands

`cd`	Changes current directory.
`date`	Displays current date.
`delete`	Deletes a file.
`diary`	Switches on/off diary file recording.
`dir`	Lists all files in current directory.
`load`	Loads workspace variables from a file.
`path`	Displays search path.
`pwd`	Displays current directory.
`save`	Saves workspace variables in a file.
`type`	Displays contents of a file.
`what`	Lists all MATLAB files in the current directory.
`wklread`	Reads .wk1 spreadsheet file.

Input/Output and Formatting Commands

Input/Output Commands	
`disp`	Displays contents of an array or string.
`fscanf`	Read formatted data from a file.
`format`	Controls screen-display format.
`fprintf`	Performs formatted writes to screen or file.
`input`	Displays prompts and waits for input.
`;`	Suppresses screen printing.

Format Codes for `fprintf` and `fscanf`	
`%s`	Format as a string.
`%d`	Format as an integer.
`%f`	Format as a floating point value.
`%e`	Format as a floating point value in scientific notation.
`%g`	Format in the most compact form: %f or %e.
`\n`	Insert a new line in the output string.
`\t`	Insert a tab in the output string.

Numeric Display Formats	
`format short`	Four decimal digits (default).
`format long`	16 decimal digits.
`format short e`	Five digits plus exponent.
`format long e`	16 digits plus exponents.
`format bank`	Two decimal digits.
`format +`	Positive, negative, or zero.
`format rat`	Rational approximation.
`format compact`	Suppresses some line feeds.
`format loose`	Resets to less compact display mode.

Vector, Matrix and Array Commands

Array Commands	
`cat`	Concatenates arrays.
`find`	Finds indices of nonzero elements.
`length`	Computers number of elements.
`linspace`	Creates regularly spaced vector.
`logspace`	Creates logarithmically spaced vector.
`max`	Returns largest element.
`min`	Returns smallest element.
`prod`	Product of each column.
`reshape`	Change size
`size`	Computes array size.
`sort`	Sorts each column.
`sum`	Sums each column.

Special Matrices	
`eye`	Creates an identity matrix.
`ones`	Creates an array of ones.
`zeros`	Creates an array of zeros.

Matrix Arithmetic	
`cross`	Computes cross products.
`dot`	Computes dot products.

Matrix Commands for Solving Linear Equations	
`det`	Computes determinant of an array.
`inv`	Computes inverse of a matrix.
`pinv`	Computes pseudoinverse of a matrix.
`rank`	Computes rank of a matrix.
`rref`	Computes reduced row echelon form.

Cell Array Functions

`cell`	Creates cell array.
`celldisp`	Displays cell array.
`cellplot`	Displays graphical representation of cell array.
`num2cell`	Converts numeric array to cell array.
`deal`	Matches input and output lists.
`iscell`	Identifies cell array.

Structure Functions

`fieldnames`	Returns field names in a structure array.
`getfield`	Returns field contents of a structure array.
`isfield`	Identifies a structure array field.
`isstruct`	Identifies a structure array.
`rmfield`	Removes a field from a structure array.
`setfield`	Sets contents of field.
`struct`	Creates structure array.

Plotting Commands

Basic xy Plotting Commands	
`axis`	Sets axis limits.
`fplot`	Intelligent plotting of functions.
`grid`	Displays gridlines.
`plot`	Generates xy plot.
`print`	Prints plot or saves plot to a file
`title`	Puts text at top of plot.
`xlabel`	Adds text label to x-axis.
`ylabel`	Adds text label to y-axis.

Plot Enhancement Commands	
`axes`	Creates axes objects.
`close`	Closes the current plot.
`close all`	Closes all plots.
`figure`	Opens a new figure window.
`gtext`	Enables label placement by mouse.
`hold`	Freezes current plot.
`legend`	Legend placement by mouse.
`refresh`	Redraws current figure window.
`set`	Specifies properties of objects such as axes.
`subplot`	Creates plots in subwindows.
`text`	Places string in figure.

Specialized Plot Commands	
`bar`	Creates bar chart.
`loglog`	Creates log-log plot.
`polar`	Creates polar plot.
`semilogx`	Creates semilog plot (logarithmic abscissa).
`semilogy`	Creates semilog plot (logarithmic ordinate).
`stairs`	Creates stairs pot.
`stem`	Creates stem plot.

Colors, Symbols and Line Types

Color		Symbol		Line	
`y`	yellow	`.`	point	`-`	solid
`m`	magenta	`o`	circle	`:`	dotted
`c`	cyan	`x`	x-mark	`-.`	dash dotted
`r`	red	`+`	plus	`--`	dashed
`g`	green	`*`	star		
`b`	blue	`d`	diamond		
`w`	white	`v`	triangle (down)		
`k`	black	`^`	triangle (up)		
		`<`	triangle (left)		
		`>`	triangle (right)		
		`p`	pentagram		
		`h`	hexagram		

Three-Dimensional Plotting Commands

`contour`	Creates contour plot.
`mesh`	Creates three-dimensional mesh surface plot.
`meshc`	Same as mesh with contour plot underneath.
`meshz`	Same as mesh with vertical lines underneath.
`plot3`	Creates three-dimensional plots from lines and points.
`surf`	Creates shaded three-dimensional mesh surface plot.
`surfc`	Same as surf with contour plot underneath.
`meshgrid`	Creates rectangular grid.
`waterfall`	Same as mesh with mesh lines in one direction.
`zlabel`	Adds text label to *z*-axis.

Histogram Functions

`bar`	Creates a bar chart.
`hist`	Aggregates the data into equally spaced bins.
`histc`	Aggregates the data into unequally spaced bins.

Programming

Logical and Relational Operators	
==	Relational operator: equal to.
~=	Relational operator: not equal to.
<	Relational operator: less than.
<=	Relational operator: less than or equal to.
>	Relational operator: greater than.
>=	Relational operator: greater than or equal to.
&	Logical operator: AND.
|	Logical operator: OR.
~	Logical operator: NOT.
xor	Logical operator: EXCLUSIVE OR.

Program Flow Control	
break	Terminates execution of a loop.
case	Provides alternate execution paths within switch structure.
else	Delineates alternate block of statements.
elseif	Conditionally executes statements.
end	Terminates for, while, and if statements.
error	Display error messages.
for	Repeats statements a specific number of times
if	Executes statements conditionally.
otherwise	Default part of switch statement.
return	Return to the invoking function.
switch	Directs program execution by comparing point with case expressions.
warning	Display a warning message.
while	Repeats statements an indefinite number of times.

Logical Functions	
any	True if any elements are nonzero.
all	True if all elements are nonzero.
find	Finds indices of nonzero elements.
finite	True if elements are finite.
isnan	True if elements are undefined.
isinf	True if elements are infinite.
isempty	True if matrix is empty.
isreal	True if all elements are real.

M-Files	
`eval`	Interpret strings containing Matlab expressions.
`feval`	Function evaluation.
`function`	Creates a user-defined function M-file.
`global`	Define global variables.
`nargin`	Number of function input arguments.
`nargout`	Number of function output arguments.
`script`	Script M-files

Timing	
`cputime`	CPU time in seconds.
`clock`	Current date and time as date vector.
`tic, toc`	Start, stop a stopwatch timer.

Mathematical Functions

Exponential and Logarithmic Functions

`exp(x)`	Exponential; e^x.
`log(x)`	Natural logarithm; ln(x).
`log10(x)`	Common (base 10) logarithm; log(x)= $\log_{10}$(x).
`sqrt(x)`	Square root; $\sqrt{x}$.

Trigonometric Functions

`acos(x)`	Inverse cosine; arcos x = $\cos^{-1}$ (x).
`acot(x)`	Inverse cotangent; arccot x = $\cot^{-1}$(x).
`acsc(x)`	Inverse cosecant; arcs x = $\csc^{-1}$ (x).
`asec(x)`	Inverse secant; arcsec x = $\sec^{-1}$ (x).
`asin(x)`	Inverse sine; arcsin x = $\sin^{-1}$ (x).
`atan(x)`	Inverse tangent; arctan x = $\tan^{-1}$ (x).
`atan2(y,x)`	Four-quadrant inverse tangent.
`cos(x)`	Cosine; cos(x).
`cot(x)`	Cotangent; cot(x).
`csc(x)`	Cosecant; csc(x).
`sec(x)`	Secant; sec(x).
`sin(x)`	Sine; sin(x).
`tan(x)`	Tangent; tan(x).

Hyperbolic Functions

`acosh(x)`	Inverse hyperbolic cosine; $\cosh^{-1}$ (x).
`acoth(x)`	Inverse hyperbolic cotangent; $\coth^{-1}$ (x).
`acsch(x)`	Inverse hyperbolic cosecant; csch^{-1} (x).
`asech(x)`	Inverse hyperbolic secant; sech^{-1} (x).
`asinh(x)`	Inverse hyperbolic sine; $\sinh^{-1}$ (x).
`atanh(x)`	Inverse hyperbolic tangent; $\tanh^{-1}$ (x).
`cosh(x)`	Hyperbolic cosine; cosh(x).
`coth(x)`	Hyperbolic cotangent; cosh(x)/sinh(x).
`csch(x)`	Hyperbolic cosecant; 1/sinh(x).
`sech(x)`	Hyperbolic secant; 1/cosh(x).
`sinh(x)`	Hyperbolic sine; sinh(x).
`tanh(x)`	Hyperbolic tangent; sinh(x)/cosh(x).

Complex Functions

`abs(x)`	Absolute value; |x|.
`angle(x)`	Angle of a complex number x.
`conj(x)`	Complex conjugate of x.
`imag(x)`	Imaginary part of a complex number x.
`real(x)`	Real part of a complex number x.

Statistical Functions

`erf(x)`	Computes the error function *erf* (x).
`mean`	Calculates the average.
`median`	Calculates the median.
`std`	Calculates the standard deviation.

Random Number Functions

`rand`	Generates uniformly distributed random numbers between 0 and 1.
`randn`	Generates normally distributed random numbers.

Numeric Functions

`ceil`	Rounds to the nearest integer toward .
`fix`	Rounds to the nearest integer toward zero.
`floor`	Rounds to the nearest integer toward - .
`round`	Rounds towards the nearest integer.
`sign`	Signum function.

String Functions

`findstr`	Finds occurrences of a string.
`strcmp`	Compares strings.
`char`	Creates character string array

Numerical Methods

Polynomial and Regression Functions	
`conv`	Computes product of two polynomials
`deconv`	Computes ratio of polynomials.
`eig`	Computes the eigenvalues of a matrix.
`poly`	Computes polynomial from roots.
`polyfit`	Fits a polynomial to data.
`polyval`	Evaluates polynomial and generates error estimates.
`roots`	Computes polynomial roots.

Interpolation Functions	
`interp1`	Linear and cubic-spline interpolations of a function of one variable.
`interp2`	Linear interpolation of a function of two variables.
`spline`	Cubic-spline interpolation.
`unmkpp`	Computes the coefficients of cubic-spine polynomials.

Root Finding and Minimization	
`fmin`	Finds minimum of single-variable function.
`fmins`	Finds minimum of multivariable function.
`fzero`	Finds zero of single-variable function.

Numerical Integration Functions	
`quad`	Numerical integration with adaptive Simpson's rule.
`quadl`	Numerical integration with adaptive Lobatto quadrature.
`trapz`	Numerical integration with the trapezoidal rule.

Numerical Differentiation Functions	
`diff(x)`	Computes the difference between adjacent elements in the vector x.
`polyder`	Differentiates a polynomial, a polynomial product, or a polynomial quotient.

ODE Solvers	
`ode23`	Nonstiff, low-order solver.
`ode45`	Nonstiff, medium-order solver.
`ode113`	Nonstiff, variable-order solver.
`ode23s`	Stiff, low-order.
`ode23t`	Moderately stiff, trapezoidal rule solver.
`ode23b`	Stiff, low-order solver.
`ode15s`	Stiff, variable-order solver.
`odeset`	Creates integrator options structure for ODE solvers.

Predefined Input Functions	
`gensig`	Generates a periodic sine, square, or pulse input.
`sawtooth`	Generates a periodic sawtooth input.
`square`	Generates a square wave input.
`stepfun`	Generates a step function input.

Symbolic Math Toolbox

Functions for Creating and Evaluating Symbolic Expressions	
`class`	Returns the class of an expression.
`digits`	Sets the number of decimal digits used to do variable precision arithmetic.
`double`	Converts an expression to numeric form.
`ezplot`	Generates a plot of a symbolic expression.
`findsym`	Finds the symbolic variables in a symbolic expression.
`numden`	Returns the numerator and denominator of an expression.
`sym`	Creates a symbolic variable.
`syms`	Creates one or more symbolic variables.
`vpa`	Sets the number of digits used to evaluate expressions.

Functions for Manipulating Symbolic Expressions	
`collect`	Collects coefficients of like powers in an expression.
`expand`	Expands an expression by carrying out jpowers.
`factor`	Factors an expression.
`poly2sym`	Converts a polynomial coefficient vector to a symbolic polynomial.
`pretty`	Displays an expression in a form that resembles typeset mathematics.
`simple`	Searches for the shortest form of an expression.
`simplify`	Simplifies an expression using Maple's simplification rules.
`subs`	Substitutes variables or expressions.
`sym2poly`	Converts an expression to a polynomial coefficient vector.

Symbolic Calculus Functions	
`diff`	Returns the derivative of an expression.
`Dirac`	Dirac delta function (unit impulse).
`Heaviside`	Heaviside function (unit step).
`int`	Returns the integral of an expression.
`limit`	Returns the limit of an expression.
`symsum`	Returns the symbolic summation of an expression.
`taylor`	Returns the Taylor series of a function.

Symbolic Solution of Algebraic and Transcendental Equations	
`solve`	Solves symbolic equations.

Symbolic Solution of Differential Equations	
`dsolve`	Returns a symbolic solution of a differential equation or set of equations.

Laplace Transform Functions	
`ilaplace`	Returns the inverse Laplace transform.
`laplace`	Returns the Laplace transform.

Symbolic Linear Algebra Functions	
`det`	Returns the determinant of a matrix.
`eig`	Returns the eigenvalues (characteristic roots) of a matrix.
`inv`	Returns the inverse of a matrix.
`poly`	Returns the characteristic polynomial of a matrix.