

APPENDIX A

MATLAB Special Characters, Reserved Words, and Functions

Special Characters	Description
<...>	Used to indicate template parameters—data to be supplied
%	Indicates a comment in an m-file
{...}	Defines a cell array
[]	The empty vector
[...]	Concatenates data, vectors, and arrays
()	Used to override operator precedence
()	Used to identify the formal and actual parameters of a function
(...)	Used to index an array
(<variable>)	Used to allow a variable to be used as a structure field
'abc'	Encloses a literal character string
'	Transposes an array
;	Suppresses output when used in commands
;	Separates rows in an array definition
:	Specifies a vector in the form <from:incr:to>
:	Used in slicing vectors and arrays
.	Used to access fields of a structure
...	Used to continue a MATLAB command to the next line

Mathematical Operators	Description
=	Assignment operator—assigns a value to a variable (memory location); not the same as an equality test
+	Scalar and array addition
—	Scalar and array subtraction
-	Unary negation
*	Matrix multiplication
.*	Element-by-element multiplication
/	Matrix division
./	Element-by-element division
^	Matrix exponentiation
.^	Element-by-element exponentiation

A-2

Appendix A MATLAB Special Characters, Reserved Words, and Functions

Logical Operators	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
~=	Not equal to
&	Element-by-element logical AND
&&	Short-circuit logical AND (scalar)
	Element-by-element logical OR (vectors)
	Short-circuit logical OR (scalar)
~	Unary not

Logical Functions	Description
all(a)	True if all the values in a (a logical vector) are true
and(a, b)	True if both a and b are true
any(a)	True if any of the values in a (a logical vector) are true
not(a)	True if a is false; false if a is true
or(a, b)	True if either a or b is true

File Input and Output	Description
[nums txt raw] = csvread(<file>)	Read comma-separated text files
csvread(<file>)	Read comma-separated text files
csvwrite(<file>, <data>)	Write comma-separated text files
dlmread (<file>, < dlm>)	Read text files separated by the given delimiting character(s)
dlmwrite(<file>, <data>, <dlm>)	Write text files separated by the given delimiting character(s)
fclose(<handle>)	Close a text file
fgetl(<handle>)	Read a line omitting the new-line character
fgets(<handle>)	Read a line including the new-line character
fh = fopen (<handle>, <why>)	Open a text file for reading or writing
fprintf (<handle>, ...)	Write to the console, or to plain text files (when <handle> is present
imread(<file>)	Read an image file
imwrite(<data>, <file>, <format>)	Write an image file
load <file>	Load the current workspace from a file

Appendix A MATLAB Special Characters, Reserved Words, and Functions A-3

<code>save <file></code>	Save workspace variables in a file
<code>[tk rest] = strtok(<str>, <dlm>)</code>	Extract a token from a string and return the remainder of the string
<code>ca = textscan(<handle>, <format>)</code>	Acquire and scan a line of text according to a specific format and save the data in a cell array
<code>[data Fs nb] = wavread(<file>)</code>	Read a sound file in .wav format
<code>wavwrite(<data>, <Fs>, <nb>, <file>)</code>	Write a sound file in .wav format
<code>[nums, txt, raw] = xlsread(<file>)</code>	Read an Excel spreadsheet
<code>xlswrite(<file>, <data>, <sheet>, <range>)</code>	Write an Excel spreadsheet in a specific row/column range

Format Control	Description
<code>%<m>.<n>e</code>	Exponential notation
<code>%<m>.<n>f</code>	Fixed point or decimal notation
<code>%<m>.<n>g</code>	Fixed point or exponential notation
<code>%q</code>	A quoted string delimited by double quotes
<code>%<n>s</code>	Character string
<code>\b</code>	Backspace
<code>\n</code>	New Line
<code>\t</code>	Tab

Display Formatting	Description
<code>format compact</code>	Set format to compact form
<code>format long</code>	Set format to 14 decimal places
<code>format long e</code>	Set format to 14 exponential places
<code>format loose</code>	Set format back to default, non-compact form
<code>format short</code>	Set format back to default, 4 decimal places
<code>format short e</code>	Set format to 4 exponential places

User Interface Management	Description
<code>ans</code>	Default variable name for results of calculations
<code>clc</code>	Clear the interactions window
<code>clear <selection></code>	Remove all (or slected) variables from the workspace
<code>clf</code>	Clear the current figure

continued on next page

A-4

Appendix A MATLAB Special Characters, Reserved Words, and Functions

close all	Close all graphics windows
exit	Terminate the user interface system
help <topic or function>	Invoke the help utility
load <file>	Load the current workspace from a file
quit	Terminate the user interface system
save <file>	Save workspace variables in a file
who	List variables in the workspace
whos	List variables and their sizes

Special Constants	Description
eps	Smallest possible difference between two floating point numbers
false	Logical false
inf	Infinity
NaN	Not a number
pi	Ratio of the circumference of a circle to its diameter
true	Logical true

Basic Mathematical Functions	Description
abs(x)	Compute the absolute value
ceil(x)	Round x to the nearest integer toward positive infinity
cross(a, b)	Vector cross product
exp(x)	Compute e to the power x
fix(x)	Round x to the nearest integer toward zero
floor(x)	Round x to the nearest integer toward minus infinity
log(x)	Compute the natural log of x
log10(x)	Compute the log base 10 of x
mod(x, a)	Compute the remainder when x is divided by a
rem(x, a)	Compute the remainder when x is divided by a
round(x)	Round x to the nearest integer
sqrt(x)	Calculate the square root of x

Trigonometry	Description
acos(x)	Compute the inverse cosine (arcsine) of x
asin(x)	Compute the inverse sine (arcsine) of x
atan(x)	Compute the inverse tangent (arctan) of x
atan2(y, x)	Compute the inverse tangent given the x and y values (4 quadrant resolution)
cos(x)	Compute the cosine of x
sin(x)	Compute the sine of x
tan(x)	Compute the tangent of x

Appendix A MATLAB Special Characters, Reserved Words, and Functions A-5

Vector, Array, and Matrix Operations	Description
<code>cumsum(v)</code>	Compute a cumulative sum of the values in <code>v</code>
<code>deal(...)</code>	Distribute cell array results among variables
<code>det(a)</code>	Compute the determinant of a matrix
<code>diag(a)</code>	Extract the diagonal from a matrix or (if provided <code>a</code> is a vector) construct a matrix with <code>a</code> as the diagonal
<code>eye(n)</code>	Generate the identity matrix of size $n \times n$
<code>find(<logical a>)</code>	Compute a linear list of the locations of the true values in a logical array
<code>fliplr(a)</code>	Flip a matrix from left to right
<code>inv(a)</code>	Compute the inverse of a matrix
<code>length(a)</code>	Determine the largest dimension of an array
<code>linspace(from, to, n)</code>	Define a linearly spaced vector
<code>magic(n)</code>	Generate a magic square of size $n \times n$
<code>[v,in] = max(a)</code>	Find the maximum value and its position in <code>a</code>
<code>mean(a)</code>	Compute the average of the elements in <code>a</code>
<code>meshgrid(x, y)</code>	Map each of two vectors into separate 2-D arrays
<code>[v,in] = min(a)</code>	Find the minimum value and its position in <code>a</code>
<code>ones(r, c)</code>	Generate an array filled with the value 1
<code>prod(x)</code>	Compute the product of all the items in <code>x</code>
<code>rand(r, c)</code>	Calculate an $r \times c$ array of evenly distributed random numbers in the range 0...1
<code>randn(r, c)</code>	Calculate an $r \times c$ array of normally distributed random numbers in the range 0...1
<code>size(a)</code>	Determine the dimensions of an array
<code>sparse</code>	Define a sparse matrix
<code>[v,in] = sort(v)</code>	Sort the vector <code>v</code> (a vector or a cell array of strings)
<code>sum(a)</code>	Find the sum of an array
<code>zeros(r, c)</code>	Build an array filled with the value 0

2-D Plotting	Description
<code>bar</code>	Generate a bar graph
<code>barh</code>	Generate a horizontal bar graph
<code>contour</code>	Generate a contour plot
<code>hist</code>	Draw a histogram
<code>loglog</code>	Generate an x-y plot, with both axes scaled logarithmically
<code>pie</code>	Generate a pie chart
<code>plot</code>	Create an x-y plot
<code>polar</code>	Create a polar plot
<code>semilogx</code>	Generate an x-y plot, with the x-axis scaled logarithmically
<code>semilogy</code>	Generate an x-y plot, with the y-axis scaled logarithmically

A-6

Appendix A MATLAB Special Characters, Reserved Words, and Functions

3-D Plotting	Description
bar3	Generate a 3-D bar graph
barh3	Generate a horizontal 3-D bar graph
gplot	Plot a graph
mesh	Generate a mesh plot of a surface
meshc	Generate a mesh plot of a surface with contours
meshz	Generate a mesh plot of a surface with a skirt
meshgrid(r, c)	Create a plaid for 3-D plots
peaks	Create a sample matrix used to demonstrate graphing functions
pie3	Generate a 3-D pie chart
plot3	Generate a 3-D line plot
sphere	Example function used to demonstrate graphing
surf	Generate a surface plot
surfc	Generate a combination surface and contour plot
waterfall	Generate a mesh plot of a surface with one skirt edge

Plot Appearance Line Type Control	Description
-	Solid
:	Dotted
-.	Dash-dot
--	Dashed
.	Point
o	Circle
x	x-mark
+	Plus
*	Star
s	Square
d	Diamond
v	Triangle down
^	Triangle up
<	Triangle left
>	Triangle right
p	Pentagram
h	Hexagram

Color Control Character	Description
b	Blue
c	Cyan
g	Green
k	Black
m	Magenta

Appendix A MATLAB Special Characters, Reserved Words, and Functions A-7

r	Red
w	White
y	Yellow

Figure Control	Description
axis	Freezes the current axis scaling for the current plot or specifies the axis dimensions
figure <n>	Open a new figure window. If present, <n> specifies a figure number
grid off/on	Turn the grid off or on
hold off/on	If hold is not set, erase figure contents before the next plotting instruction
legend(ca)	Add a legend to a graph
shading <value>	Shade a surface plot with one color per grid section
subplot(plts, n)	Divide the graphics window up into sections available for plotting
text(x,y,str)	Add text to a plot
title(str)	Add a title to a plot
xlabel(str)	Add a label to the x-axis
ylabel(str)	Add a label to the y-axis
zlabel(str)	Add a label to the z-axis

Color Map Values	Description
autumn	yellow, orange, and red colors
bone	shades of gray
colorcube	multiple multi-color bands
cool	light blue to purple
copper	shades of red-brown
flag	multiple red, white, and blue bands
hot	deep red through orange to white
hsv	single spectrum from red to purple
jet	(default) rainbow from blue to red
pink	from dark to light pink
prism	multiple bands of spectrum colors
spring	from purple to yellow
summer	from dark green to yellow
white	all white
winter	from dark blue to light green

String Operations	Description
disp(...)	Display matrix or text
fprintf(...)	Print formatted information
input(...)	Prompt the user to enter a value and parse the result
int2str(a)	Convert an integer to its numerical representation

continued on next page

A-8

Appendix A MATLAB Special Characters, Reserved Words, and Functions

<code>num2str(a,n)</code>	Convert a number to its numerical representation with <code>n</code> decimal places
<code>sprintf(...)</code>	Format a string result
<code>sscanf(...)</code>	Formatted input conversion
<code>strcmp(s1, s2)</code>	Compare two strings—returns <code>true</code> if equal
<code>strcmpi(s1, s2)</code>	Compare two strings without regard to case—returns <code>true</code> if equal
<code>textscan</code>	Scan a text string

Time-Related Functions	Description
<code>clock</code>	Determine the current time on the CPU clock
<code>etime</code>	Find elapsed time
<code>pause</code>	Pause the execution of a program, either until any key is hit or for a specified number of seconds
<code>tic</code>	Start a timing sequence
<code>toc</code>	Stop a timing sequence and returns the elapsed time

Numerical Methods	Description
<code>diff(v)</code>	Compute the differences between adjacent values in a vector
<code>interp1</code>	Compute linear and cubic interpolation
<code>interp2</code>	Compute linear and cubic interpolation
<code>interp3</code>	Compute linear and cubic interpolation
<code>polyfit(x, y, n)</code>	Compute a least-squares polynomial
<code>polyval(c, x)</code>	Evaluate a polynomial
<code>spline(x, y)</code>	Spline interpolation

Program Control	Description
<code>break</code>	A command within a loop module that forces control to the statement following the innermost loop
<code>case</code>	A specific value alternative within a <code>switch</code> statement
<code>catch</code>	End of a suspect code block where the exception is trapped
<code>continue</code>	Skip to the end of the innermost loop, but remains inside it
<code>else</code>	Within an <code>if</code> statement, begin the code block executed when the condition is false
<code>elseif</code> <code><expression></code>	Within an <code>if</code> statement, begin a subsequent test when the result of the previous test is false
<code>end</code>	Terminate a function specification or an <code>if</code> , <code>switch</code> , <code>for</code> , <code>while</code> , or <code>catch</code> block.
<code>end</code>	When indexing, the value of the last element in an index vector
<code>for var = v</code>	A code block repeated as many times as there are elements in the vector <code>v</code>
<code>function</code>	Identify an m-file as a function or begin a helper function within a function file

Appendix A MATLAB Special Characters, Reserved Words, and Functions A–9

<code>error(str)</code>	Throw an exception to announce an error with the string provided
<code>global var</code>	Define a variable as globally accessible
<code>if <expression></code>	Begin a conditional module—the following code block is executed if the logical expression is true
<code>lasterror</code>	Provide a structure describing the environment from which an exception was thrown
<code>nargin</code>	Determine the number of input parameters actually supplied by a function's caller
<code>nargout</code>	Determine the number of output parameters actually requested by a function's caller
<code>otherwise</code>	Catch-all code block at the end of a <code>switch</code> statement
<code>switch <variable></code>	Begin a code module selecting specific values of the <code>variable</code> (must be countable)
<code>try</code>	Begin a block of suspect code from which an exception might be thrown
<code>while <expression></code>	A code module repeated as long as the logical expression is true

Data Class Operations	Description
<code>char(...)</code>	Cast to a character type
<code>class(<object>)</code>	Determine the data type of an object
<code>double(a)</code>	Cast a to type <code>double</code>
<code>int8/16/32/64(a)</code>	Cast a to integer type with the specified number of bits
<code>uint8/16/32/64(a)</code>	Cast a to unsigned integer type with the specified number of bits
<code>isa(obj, str)</code>	Test for a given data type
<code>ischar(ch)</code>	Determine whether the given object is of type <code>char</code>
<code>iscell(...)</code>	Determine whether the given object is a cell
<code>isempty(a)</code>	Test for the empty vector <code>[]</code>
<code>islogical(a)</code>	Determine whether the given object is of type <code>logical</code>
<code>isnumeric(a)</code>	Determine whether the given object is of type <code>double</code>
<code>isspace(a)</code>	Test for the space character
<code>isstruct(a)</code>	Determine whether the given object is a structure

Structure Operations	Description
<code>fieldnames(str)</code>	Return a cell array containing strings that are the names of the fields in the structure
<code>getfield(str, field)</code>	Extract the value of the field
<code>isfield(str, field)</code>	Return true if the string is a field in the specified structure
<code>str = rmfield(str, field)</code>	Return a copy of the given structure with the given field removed
<code>str = setfield(str, field, value)</code>	Construct a structure in which the value of the field has been changed to the given value
<code>struct(...)</code>	Construct a structure from <code><fieldname> <value></code> pairs of parameters

