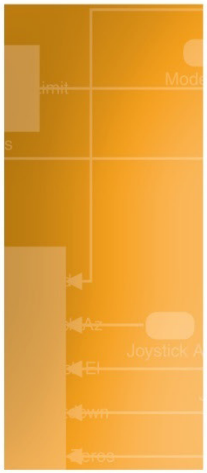
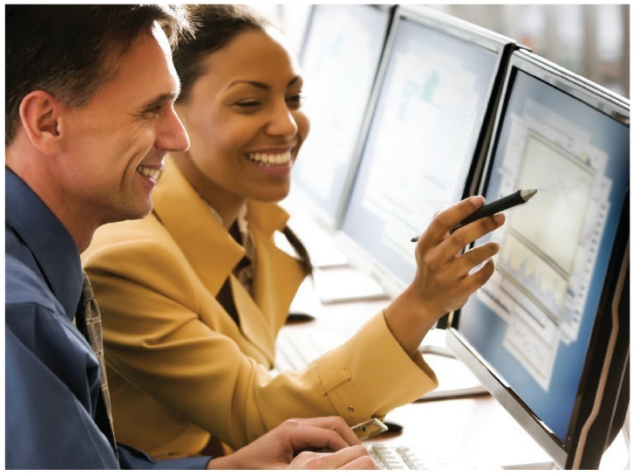
MathWorks | Training Services

Appendix A: MATLAB® Reference

MATLAB® Fundamentals for Aerospace Applications





Development Environment

Command Window

`clc`	Clear Command Window
`format`	Set numeric format
`;`	Suppress output display
`...`	Continue to next line without execution
↑	Scroll through previously typed commands
s↑	Scroll through commands beginning with s
Esc	Clear command line
Home, End	Move to beginning, end of expression
Tab	Complete partial expressions
`!`	Invoke operating system

Files and directories

`pathtool`	Set search path
`which`	Locate files on search path
`matlabroot`	Locate MATLAB® installation directory
`pwd`	Identify current directory
`cd`	Change current directory
`dir,ls`	List current directory
`what`	List current directory by file type
`mkdir`	Make new directory
`rmdir`	Remove directory
`copyfile`	Copy file from source to destination
`movefile`	Move file from source to destination
`delete`	Delete file
`type`	Display text file
`echo`	Display commands in file during execution
`more`	Display paged output
Ctrl+C	Return control to the command line

Help

`help`	Display help information
`doc`	Open documentation
`demo`	Application-specific demos
`lookfor`	Search help by keyword
`ver`	Version information

Variables

General

`clear`	Remove variables from workspace
`who`	Display workspace variables
`whos`	Display workspace variables (verbose)
`size`	Variable dimensions
`length`	Longest dimension
`linspace,:`	Create vectors
`meshgrid`	Create grid from vectors
`.'`	Transpose
`'`	Conjugate transpose
`horzcat,[ , ]`	Horizontal concatenation
`vertcat,[ ; ]`	Vertical concatenation
`cat`	Concatenate along specified dimension
`reshape`	Change dimensions
`repmat`	Replicate and tile
`=`	Assignment
`(),{},.`	Subscripted assignment and reference
`global`	Declare global variables

Workspace allocation

`zeros`	All elements 0 (any numerical type)
`ones`	All elements 1 (any numerical type)
`cell`	Preallocate for cell array
`struct`	Preallocate for structure array
`pack`	Consolidate workspace memory

Random numbers

`rand`	Uniformly distributed random numbers
`randn`	Normally distributed random numbers
`randperm`	Random permutation of integers

Operators and Constants

Arithmetic operators

`plus`	Plus	`+`
`uplus`	Unary plus	`+`
`minus`	Minus	`-`
`uminus`	Unary minus	`-`
`mtimes`	Matrix multiplication	`*`
`times`	Array multiplication	`.*`
`mpower`	Matrix power	`^`
`power`	Array power	`.^`
`mldivide`	Backslash or left matrix divide	`\`
`mrdivide`	Slash or right matrix divide	`/`
`ldivide`	Left array divide	`.\`
`rdivide`	Right array divide	`./`

Relational operators

`eq`	Equal	`==`
`ne`	Not equal	`~=`
`lt`	Less than	`<`
`gt`	Greater than	`>`
`le`	Less than or equal	`<=`
`ge`	Greater than or equal	`>=`

Set operators

`union`	Set union
`intersect`	Set intersection
`setdiff`	Set difference
`setxor`	Set exclusive OR
`ismember`	Test for membership
`unique`	Find nonrepeated values

Logical operators

`and`	Element-wise AND	`&`
`or`	Element-wise OR	`|`
`xor`	Exclusive OR	
`not`	Negation	`~`
`true`	Logical 1	
`false`	Logical 0	
`any`	Test for any nonzero elements	
`all`	Test for all nonzero elements	
`bitand`	Bitwise AND	
`bitcmp`	Complement bits	
`bitor`	Bitwise OR	
`bitxor`	Bitwise XOR	
`bitshift`	Bitwise shift	

Constants

`i,j,1i,1j`	Imaginary unit ($\sqrt{-1}$)
`pi`	Trigonometric unit (π)
`Inf,inf`	IEEE® Infinity
`NaN,nan`	IEEE Not-a-Number
`eps`	Floating-point relative accuracy
`realmax`	Largest positive floating-point number
`realmin`	Smallest positive floating-point number
`intmax`	Largest value of specified integer type
`intmin`	Smallest value of specified integer type

Mathematical Operations

Trigonometry

`sin`	Sine	Similarly for other trigonometric functions
`sind`	Sine (degrees)	
`asin`	Inverse sine	
`sinh`	Hyperbolic sine	

Exponentials and logarithms

`sqrt`	Square root
`nthroot`	Nth root
`exp`	Natural exponential
`pow2`	Power of 2
`log`	Natural logarithm
`log2`	Base 2 logarithm
`log10`	Base 10 logarithm

Complex numbers

`real`	Real part
`imag`	Imaginary part
`abs`	Modulus
`angle`	Phase angle
`conj`	Conjugate

Rounding

`fix`	Round toward zero
`floor`	Round toward `-Inf`
`ceil`	Round toward `Inf`
`round`	Round toward nearest integer
`mod`	Modulus after division
`rem`	Remainder after division
`rat,rats`	Rational fraction approximation

Discrete

`perms`	All permutations
`nchoosek`	All combinations (binomial coefficients)
`factorial`	Factorial
`lcm,gcd`	LCM, GCD
`factor`	Prime factors
`primes,isprime`	List primes, test for primality

Statistical Operations

Descriptive statistics

`sort`	Sort data
`max`	Maximum value
`min`	Minimum value
`mean`	Average value
`std`	Standard deviation
`var`	Variance
`median`	Median
`cumprod`	Cumulative product
`cumsum`	Cumulative sum

Interpolation and curve fitting

`spline`	Cubic spline interpolation
`pchip`	Piecewise cubic Hermite interpolation
`interp1`	General vector data interpolation
`interp2`	General matrix data interpolation
`griddata`	Interpolate irregularly spaced data
`polyfit`	Polynomial curve fit
`polyval`	Polynomial evaluation
`roots`	Polynomial roots

Covariance and correlation

`cov`	Covariance
`corrcoef`	Correlation coefficient

Convolution

`conv`	Convolution
`conv2`	Two-dimensional convolution
`convn`	N-dimensional convolution

Fourier transforms

`fft`	Discrete Fourier transform
`fftshift`	Rearrange `fft` for 0-centered spectrum
`ifft`	Inverse discrete Fourier transform

Plotting

Vector data

`plot,plot3`	Line plots
`yyaxis`	Line plots with two vertical scales
`comet,comet3`	Animated line plots
`scatter,scatter3`	Scatter plots
`bar,bar3,bar3h`	Bar plots
`histogram,hist`	Histogram
`rose`	Angle histogram
`polar`	Polar coordinate plot
`stem, stem3`	Stem plots
`stairs`	Stairstep plot
`area`	Filled area plot
`pie, pie3`	Pie charts
`compass`	Radial vector plot
`feather`	Horizontal vector plot
`quiver,quiver3`	Vector field plots
`triplot`	Triangle plot
`fill`	Filled polygon plot
`voronoi`	Voronoi diagram
`convhull`	Convex hull

Matrix data

`surf,surfc,surfl`	Surface plots
`surfnorm`	Surface normals
`cylinder`	Cylinder plot
`sphere,ellipsoid`	Sphere, ellipsoid plot
`mesh,meshc,meshz`	Wire-frame surface plots
`waterfall`	Row mesh plot
`ribbon`	Column ribbon plot
`trimesh,tetramesh`	Triangle, tetrahedron mesh plots
`contour,contourf,` `contour3,contourc`	Contour plots
`image,imagesc`	Images
`bar,bar3,bar3h`	Matrix bar plots
`plotmatrix`	Matrix scatter plots
`fill3`	Three-dimensional filled polygon plot
`patch`	Patch object
`spy`	Sparsity pattern

Volume data

`slice`	Slice plot
`contourslice`	Slice plot with contours
`isosurface`	Isosurface plot
`quiver3`	Vector field plot
`coneplot`	Vector field plot with cone markers
`streamline`	Vector field streamline plot
`streamribbon`	Vector field streamline and curl plot

Animation

`pause`	Pause execution (use to slow frame rate)
`drawnow`	Force figure to be rendered immediately
`getframe`	Capture movie frame
`movie`	Play recorded movie
`movie2avi`	Convert movie to AVI file
`frame2im`	Convert movie frame to image
`im2frame`	Convert image to movie frame

Plot Formatting

Figure

`figure`	Create or bring forward figure
`refresh`	Refresh figure
`clf`	Clear figure
`close`	Close figure

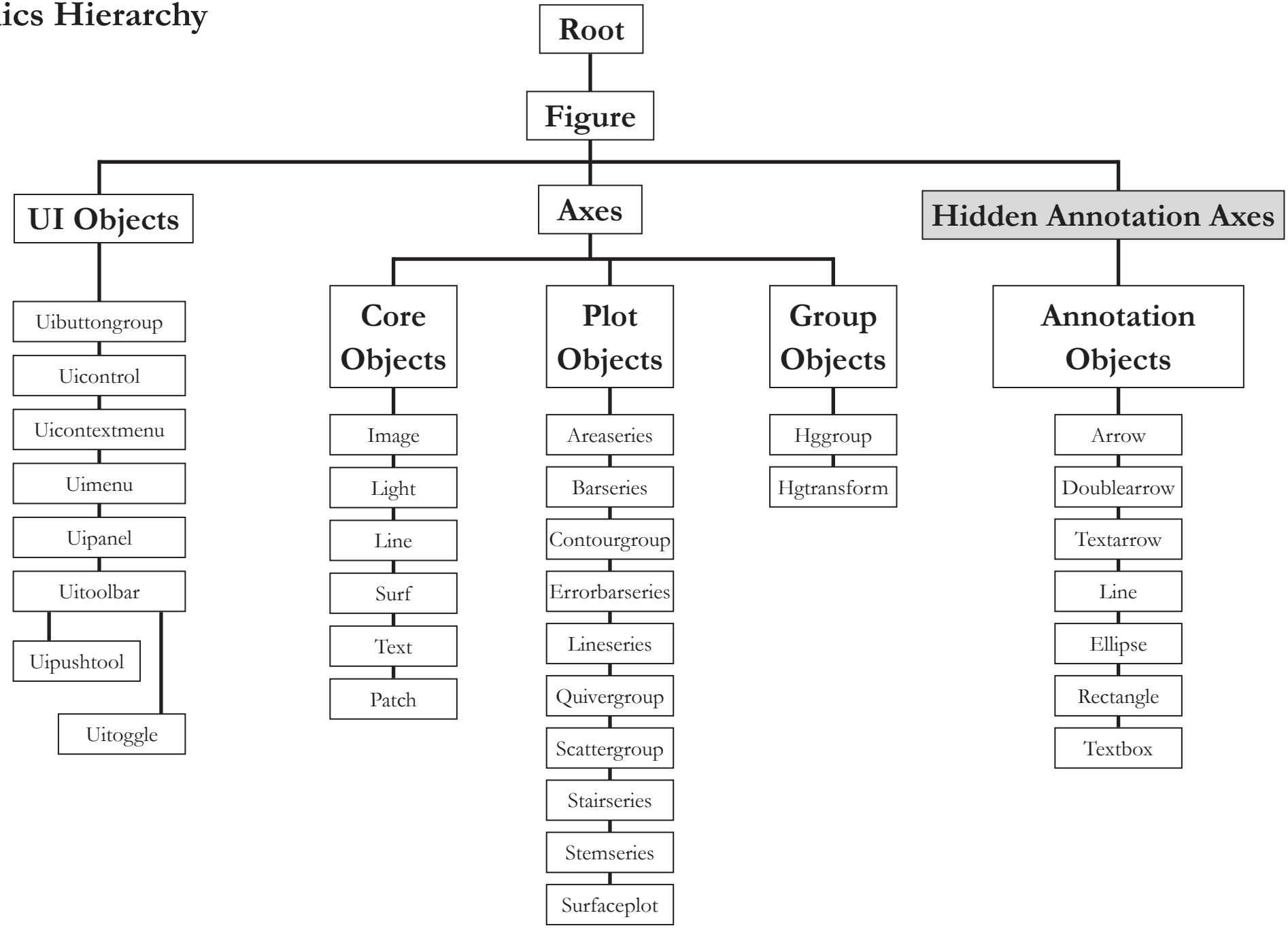
Axes and scales

`hold`	Retain/reset current axes properties
`axis`	Modify axes properties
`subplot`	Create subplots
`linkaxes`	Synchronize axes limits
`pan`	Turn panning on or off
`loglog`	Log scales
`semilogx,semilogy`	Semilog scales
`grid`	Add grid lines
`zoom`	Turn zooming on or off
`plottools`	Start plotting tools

Annotation

`xlabel,ylabel,` `zlabel`	Axes labels
`title`	Figure title
`legend`	Multiple plot legend
`xlim,ylim,zlim`	Control axis limits
`xticks,yticks,` `zticks`	Set axis tick values
`xticklabel,` `yticklabel,` `zticklabel`	Set axis tick labels
`text`	Text annotation
`gtext`	Place text annotation with mouse
`annotation`	Annotation objects
`texlabel`	T_EX format from character vector

Graphics Hierarchy



Characters and Text

Characters

`char`	Convert numbers to ASCII characters
`double`	Convert ASCII characters to numbers
`native2unicode`	Convert numbers to Unicode® characters
`unicode2native`	Convert Unicode characters to numbers

Text

`' '`	Create character vector
`str2mat`	Create character matrix (with empty characters)
`strcat, [ , ]`	Concatenate text horizontally
`num2str,mat2str`	Convert numbers to character arrays
`str2num`	Convert characters to numbers
`cellstr`	Create cell array of character vectors
`deblank`	Strip trailing blanks
`strtrim`	Remove leading and trailing whitespace
`lower`	Convert text to lowercase
`upper`	Convert text to uppercase
`strjust`	Justify character array
`date`	Current date character vector
`datestr`	Convert date number to character vector
`datenum`	Convert date to number
`strcmp`	Compare character vectors
`strcmpi`	Compare character vectors, ignoring case
`strncmp`	Compare n characters
`strncmpi`	Compare n characters, ignoring case
`strfind`	Find character vector within another
`strrep`	Find and replace text
`strtok`	Return selected parts of character vector
`regexp`	Match regular expression
`regexpi`	Match regular expression, ignoring case
`regexprep`	Replace text using regular expression
`sscanf`	Read text under format control
`sprintf`	Write formatted data to text
`ischar,isletter`	Test for character arrays, alphabetic letters
`disp`	Display text
`eval`	Execute MATLAB expression text

T_EX Characters

Markup

^	Superscript
_	Subscript
\bf	Bold font
\it	Italic font
\rm	Normal font
{}	Extent of markup
\fontname	Set font
\fontsize	Set font size
\color	Set font color

Special characters

\\	\
\{ \}	{ }
_	_
\^	^
\newline	Carriage return

Arrows

\leftarrow	←
\rigtharrow	→
\leftrightarrow	↔
\uparrow	↑
\downarrow	↓

Miscellaneous math

\infty	∞
\nabla	∇
\int	∫
\partial	∂
\prime	′
\surd	√
\Im	ℑ
\Re	ℜ
\wp	℘
\aleph	ℵ
\0 \oslash	∅
\o	o
\circ	°
\forall	∀
\exists	∃
\angle	∠
\ldots	…

Greek letters

\alpha	α
\beta	β
(etc)	
\varpi	ϖ
\varsigma	ς
\vartheta	ϑ
\Gamma	Γ
\Delta	Δ
(etc)	

Operations

\otimes	⊗
\oplus	⊕
\bullet	•
\ast	*
\cdot	·
\times	×
\div	÷
\pm	±
\neg	¬
\cap	∩
\cup	∪
\vee	∨
\wedge	∧
\perp	⊥

Delimiters

\lfloor	⌊
\rfloor	⌋
\lceil	⌈
\rceil	⌉
\langle	⟨
\rangle	⟩
\mid	|

Relations

\equiv	≡
\cong	≅
\approx	≈
\sim	~
\propto	∝
\neq	≠
\leq	≤
\geq	≥
\supset	⊃
\supseteq	⊇
\subset	⊂
\subseteq	⊆
\in	∈
\ni	∋

Miscellaneous

\copyright	©
\clubsuit	♣
\diamondsuit	♦
\heartsuit	♥
\spadesuit	♠

Code Files and Programming

Code files

`script`	Script file description
`run`	Run a script not on the path
`function`	Function declaration
`@`	Create function handle
`( , )`	Functional input arguments
`[ , ]`	Functional output arguments
`end`	Terminate block of code
`%`	Comment code
`...`	Continue expression on next line
`depfun`	List file dependencies
`depdir`	List dependent directories
`mfilename`	Name of script file currently executing
`inputname`	Name of function input
`varargin`	Variable number of input arguments
`varargout`	Variable number of output arguments
`nargin`	Number of input arguments
`nargout`	Number of output arguments
`narginchk`	Check number of input arguments
`nargoutchk`	Check number of output arguments
`persistent,global`	Declare persistent, global variable
`echo`	Echo code to console during execution
`input`	Request user input
`pause`	Halt execution temporarily
`pcode`	Create encrypted file (P-file)
`rehash`	Refresh file system path caches

Error Handling

`try`	Attempt block of code, catch errors
`catch`	Specify how to respond to an error in `try`
`error`	Display error message
`ferror`	Query for errors during file input or output
`rethrow`	Reissue error
`warning`	Display warning message
`lastwarn`	Return last warning message

Keywords

`for`	Execute code a specified number of times
`while`	Execute code until a condition fails
`break`	Terminate execution of `for` or `while` loop
`continue`	Go to next iteration of `for` or `while` loop
`return`	Return to invoking function
`if`	Conditionally execute code
`elseif`	Conditionally execute other code
`else`	Execute remaining case (`if-else`)
`switch`	Declare switch variable
`case`	Execute code conditional on `switch`
`otherwise`	Execute remaining case (`switch-case`)
`end`	Terminate conditional block

Graphics Programming

`get`	Query for property value
`set`	Modify property value
`findobj`	Obtain visible handles of described objects
`findall`	Obtain hidden handles of described objects
`gcf`	Handle of current figure window
`gca`	Handle of current axis
`gco`	Handle of current object
`groot`	Handle of root object

Timing

`bench`	Time and compare hardware
`tic`	Start stopwatch timer
`toc`	Stop stopwatch timer
`profile`	Time and profile code
`timer`	Create timer object
`start`	Start timer object
`wait`	Wait for timer object to complete
`stop`	Stop timer object
`set`	Display or set timer object properties
`get`	Retrieve timer object properties
`delete`	Delete timer object from memory

Precedence

Interpreter precedence

1. Variable
2. Function from imported package
3. Nested function
4. Local function
5. Private function
6. Class method
7. Class constructor in @ directory
8. File in the current directory
9. File on the path

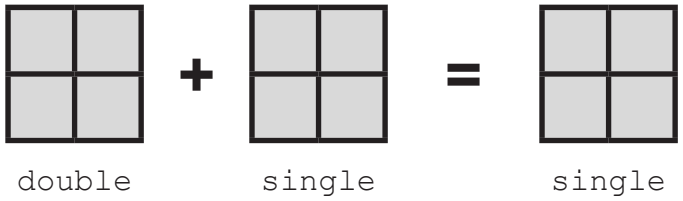
File precedence

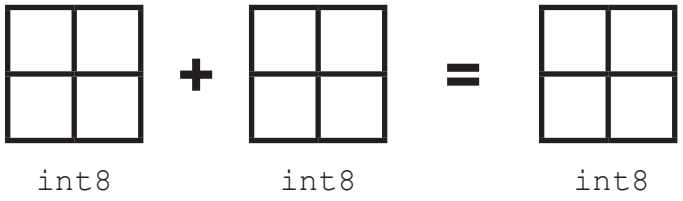
1. Built-in file
2. MATLAB executable (MEX) file
3. SLX- file or MDL-file (Simulink® model)
4. Live script (`.mlx`)
5. P-code file
6. Text code file (`.m`)

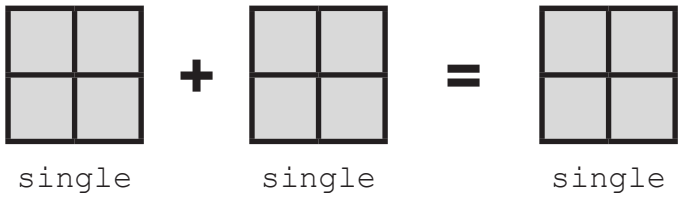
Operator precedence

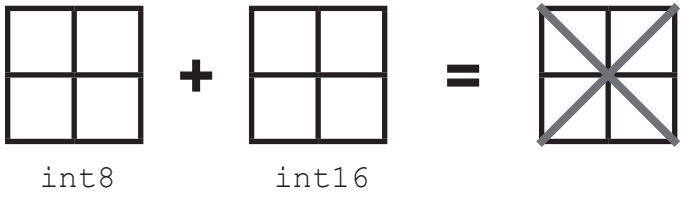
1. Parentheses `()`
2. Transpose (`.'`), power (`.^`), complex conjugate transpose (`'`), matrix power (`^`)
3. Unary plus (+), unary minus (−), logical negation (~)
4. Multiplication (`.*`), right division (`./`), left division (`.\`), matrix multiplication (*), matrix right division (/), matrix left division (\)
5. Addition (+), subtraction (−)
6. Colon operator (`:`)
7. Less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=), equal to (==), not equal to (~=)
8. Element-wise AND (`&`)
9. Element-wise OR (`|`)
10. Short-circuit AND (`&&`)
11. Short-circuit OR (`||`)

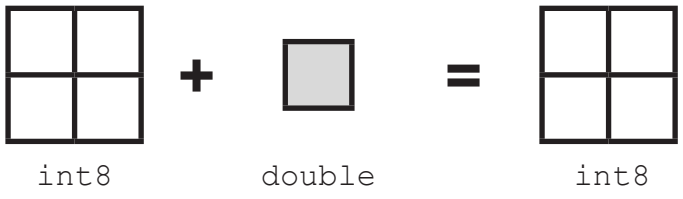
Nondouble Arithmetic

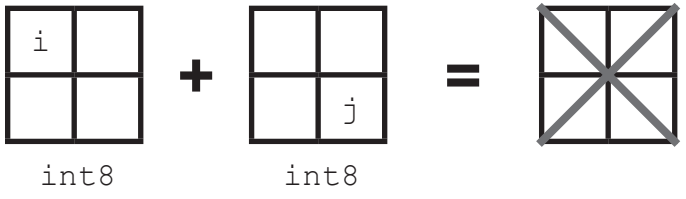


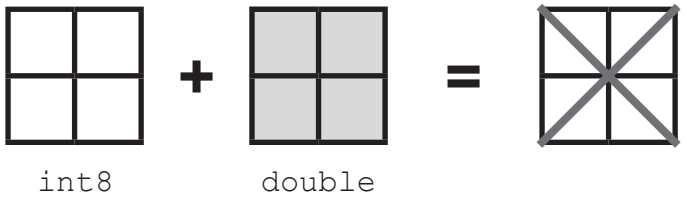












"+" = + - .* ./ .^ '

Data Types

Storage formats

`full`	Convert sparse matrix to full
`sparse`	Convert full matrix to sparse

Numeric arrays

`zeros,ones`	Preallocate for any numeric type
`double`	Convert to double precision
`single`	Convert to single precision
`uint8,uint16, uint32,uint64`	Convert to unsigned integer precision
`int8,int16, int32,int64`	Convert to signed integer precision
`isnumeric`	Test for numeric arrays
`isfloat`	Test for double and single arrays
`isinteger`	Test for integer arrays

Logical arrays

`logical`	Convert to logical
`islogical`	Test for logical arrays

Structure arrays

`struct`	Construct structure array
`fieldnames`	Get field names
`rmfield`	Remove field
`orderfields`	Order fields
`struct2cell`	Convert structure array to cell array
`isstruct`	Test for structure arrays
`isfield`	Test if field exists

Cell arrays

`cell`	Construct cell array
`cellfun`	Apply function to cell array elements
`celldisp`	Display cell array contents
`cellplot`	Graphical display of cell array structure
`cell2mat`	Convert cell array of matrices to matrix
`cell2struct`	Convert cell array to structure array
`num2cell`	Convert matrix to cell array
`mat2cell`	Convert matrix to cell array of submatrices
`iscell`	Test for cell arrays

Function handles

`@`	Construct function handle
`functions`	Information on function handles
`feval`	Evaluate function handle or file function
`func2str`	Convert function handle to name text
`str2func`	Convert function name text to handle
`isa`	Test for type (including function handles)
`isequal`	Test for equality (handles and other arrays)

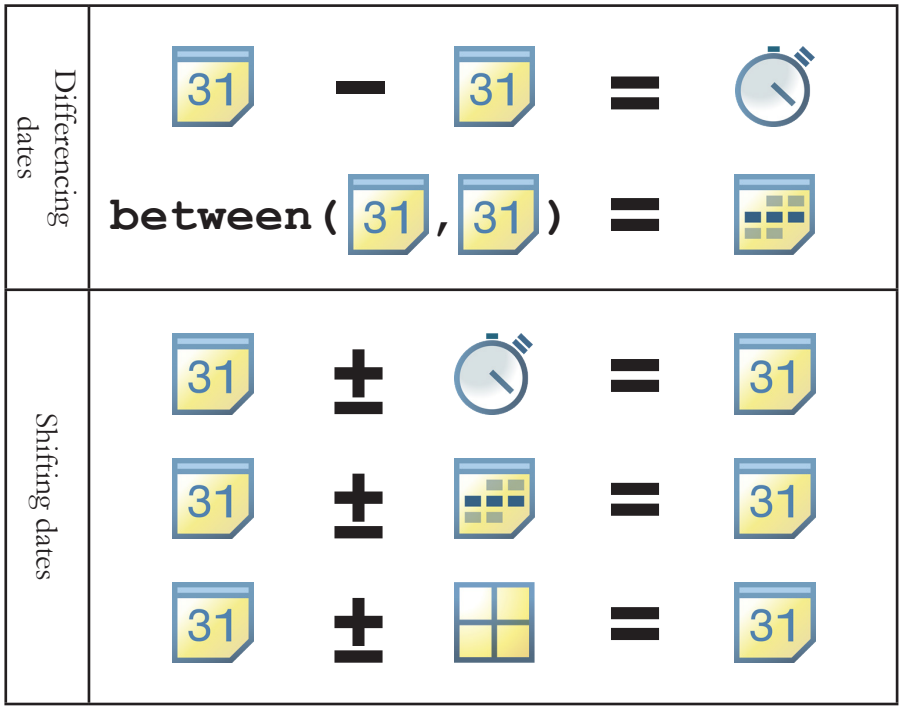
Java™ classes

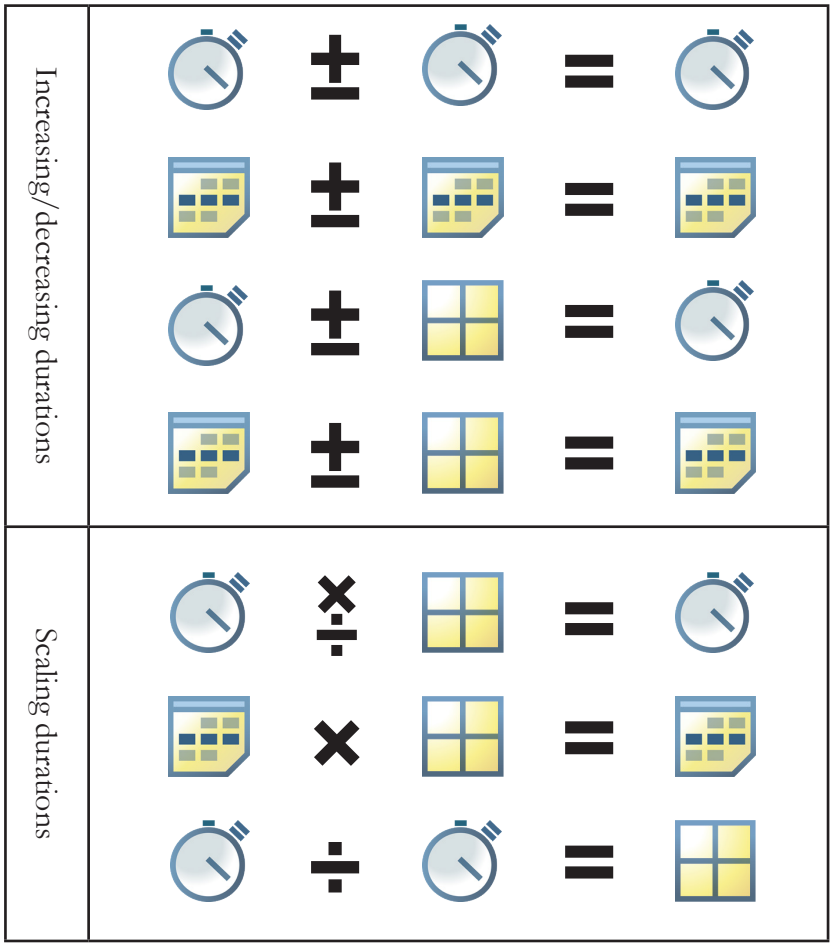
`javaArray`	Construct a Java array
`javaObject`	Construct a Java object
`javaMethod`	Invoke a Java method
`methods`	Display class methods
`javaaddpath`	Add entries to dynamic class path
`javarmpath`	Remove entries from dynamic class path
`isjava`	Test for Java object

Converting Data Types

To→ ↓ From		char	cell	datetime	duration	table	logical	struct	function_handle	categorical
		num2str	num2cell mat2cell	datetime	See 8-9	array2table	logical			categorical
	str2double str2num		cellstr	datetime					str2func	categorical
	cell2mat	char		datetime		cell2table		cell2struct		categorical
	datenum	char	cellstr							
	See 8-9	char	cellstr							
	table2array		table2cell					table2struct		
	double		num2cell mat2cell							categorical
			struct2cell			struct2table				
		func2str								
	double	char	cellstr num2cell							

Date Arithmetic





File I/O

Workspace variables

`save`	Save workspace to MAT-file
`load`	Load workspace from MAT-file

***read functions**

`audioread`	Audio file
`cdfread`	Common data format file
`csvread`	Comma-separated value file
`dlmread`	ASCII-delimited file of numeric data
`fitsread`	Flexible image transport system file
`h5read`	Hierarchical data format file
`imread`	Image or graphics file
`multibandread`	Binary band interleaved data file
`VideoReader`	Video file
`webread`	Web content
`xlsread`	Microsoft Excel® spreadsheet file
`xmlread`	XML document

***write functions**

`audiowrite`	Audio file
`cdfwrite`	Common data format file
`csvwrite`	Comma-separated value file
`dlmwrite`	ASCII-delimited file of numeric data
`h5write`	Hierarchical data format file
`imwrite`	Image or graphics file
`multibandwrite`	Binary band interleaved data file
`VideoWriter`	Video file
`webwrite`	Web content
`xlswrite`	Microsoft Excel spreadsheet file
`xmlwrite`	XML document

Opening and closing

`fopen`	Open file
`fclose`	Close file

Binary data

`fread`	Read binary data from file
`fwrite`	Write binary data to file

Formatted text

`textscan`	Read formatted data from file
`fscanf`	Read formatted data from file
`fprintf`	Write formatted data to file
`fgetl`	Read line (discard newline character)
`fgets`	Read line (keep newline character)

Conversion characters

`%d,%i`	Integer
`%u`	Unsigned integer
`%o`	Octal integer
`%x`	Hexadecimal integer
`%f`	Floating-point number (six decimal places)
`%e`	Floating-point number (exponential format)
`%g`	Floating-point number (`%f` or `%e` format)
`%c`	Single character
`%s`	Null-terminated character vector
`%D`	`datetime` variable
`%C`	`categorical` variable

Text conversion

`sprintf`	Write formatted data to text
`sscanf`	Read text under format control

Positioning

`fseek`	Set file position indicator
`ftell`	Get file position indicator
`frewind`	Rewind file
`feof`	Test for end-of-file
`ferror`	Inquire file error status