

# MATLAB Quick Reference

---

<b>Introduction</b> . . . . .	A-2
<b>General Purpose Commands</b> . . . . .	A-3
<b>Operators and Special Characters</b> . . . . .	A-5
<b>Logical Functions</b> . . . . .	A-5
<b>Language Constructs and Debugging</b> . . . . .	A-5
<b>Elementary Matrices and Matrix Manipulation</b> . . . . .	A-7
<b>Specialized Matrices</b> . . . . .	A-8
<b>Elementary Math Functions</b> . . . . .	A-8
<b>Specialized Math Functions</b> . . . . .	A-9
<b>Coordinate System Conversion</b> . . . . .	A-10
<b>Matrix Functions - Numerical Linear Algebra</b> . . . . .	A-10
<b>Data Analysis and Fourier Transform Functions</b> . . . . .	A-11
<b>Polynomial and Interpolation Functions</b> . . . . .	A-12
<b>Function Functions - Nonlinear Numerical Methods</b> . . . . .	A-13
<b>Sparse Matrix Functions</b> . . . . .	A-14
<b>Sound Processing Functions</b> . . . . .	A-15
<b>Character String Functions</b> . . . . .	A-16
<b>File I/O Functions</b> . . . . .	A-17
<b>Bitwise Functions</b> . . . . .	A-17
<b>Structure Functions</b> . . . . .	A-18
<b>MATLAB Object Functions</b> . . . . .	A-18
<b>MATLAB Interface to Java Functions</b> . . . . .	A-18
<b>Cell Array Functions</b> . . . . .	A-19
<b>Multidimensional Array Functions</b> . . . . .	A-19
<b>Data Visualization</b> . . . . .	A-19
<b>Graphical User Interfaces</b> . . . . .	A-24
<b>Serial Port I/O</b> . . . . .	A-25

## Introduction

This appendix lists the MATLAB functions as they are grouped in Help by subject. Each table contains the function names and brief descriptions. For complete information about any of these functions, refer to Help and either:

- Select the function from the MATLAB Function Reference (Functions by Category or Alphabetical List of Functions), or
- From the **Search** tab in the Help Navigator, select **Function Name** as **Search type**, type the function name in the **Search for** field, and click **Go**.

---

**Note** If you are viewing this book from Help, you can click on any function name and jump directly to the corresponding MATLAB function page.

---

## General Purpose Commands

This set of functions lets you start and stop MATLAB, work with files and the operating system, control the command window, and manage the environment, variables, and the workspace.

### Managing Commands and Functions

<code>addpath</code>	Add directories to MATLAB's search path
<code>doc</code>	Display HTML documentation in Help browser
<code>docopt</code>	Display location of help file directory for UNIX platforms
<code>genpath</code>	Generate a path string
<code>help</code>	Display M-file help for MATLAB functions in the Command Window
<code>helpbrowser</code>	Display Help browser for access to all MathWorks online help
<code>helpdesk</code>	Display Help browser
<code>helpwin</code>	Display M-file help and provide access to M-file help for all functions
<code>lasterr</code>	Last error message
<code>lastwarn</code>	Last warning message
<code>license</code>	Show MATLAB license number
<code>lookfor</code>	Search for specified keyword in M-file help entries
<code>partialpath</code>	Partial pathname
<code>path</code>	Control MATLAB's directory search path
<code>pathtool</code>	Open the GUI for viewing and modifying MATLAB's path
<code>profile</code>	Start the M-file profiler, a utility for debugging and optimizing code
<code>profreport</code>	Generate a profile report

### Managing Commands and Functions (Continued)

<code>rehash</code>	Refresh function and file system caches
<code>rmpath</code>	Remove directories from MATLAB's search path
<code>support</code>	Open MathWorks Technical Support Web page
<code>type</code>	List file
<code>ver</code>	Display version information for MATLAB, Simulink, and toolboxes
<code>version</code>	Get MATLAB version number
<code>web</code>	Point Help browser or Web browser at file or Web site
<code>what</code>	List MATLAB-specific files in current directory
<code>whatsnew</code>	Display README files for MATLAB and toolboxes
<code>which</code>	Locate functions and files

### Managing Variables and the Workspace

<code>clear</code>	Remove items from the workspace
<code>disp</code>	Display text or array
<code>length</code>	Length of vector
<code>load</code>	Retrieve variables from disk
<code>memory</code>	Help for memory limitations
<code>mlock</code>	Prevent M-file clearing
<code>munlock</code>	Allow M-file clearing
<code>openvar</code>	Open workspace variable in Array Editor for graphical editing
<code>pack</code>	Consolidate workspace memory
<code>save</code>	Save workspace variables on disk
<code>saveas</code>	Save figure or model using specified format

---

**Managing Variables and the Workspace (Continued)**

size	Array dimensions
who, whos	List the variables in the workspace
workspace	Display the Workspace browser, a GUI for managing the workspace

---

**Controlling the Command Window**

clc	Clear Command Window
echo	Echo M-files during execution
format	Control the display format for output
home	Move cursor to upper left corner of Command Window
more	Control paged output for the command window

---

**Working with Files and the Operating Environment**

beep	Produce a beep sound
cd	Change working directory
checkin	Check file into source control system
checkout	Check file out of source control system
cmopts	Get name of source control system, and PVCS project filename
copyfile	Copy file
customverctrl	Allow custom source control system
delete	Delete files and graphics objects
diary	Save session to a disk file
dir	Display a directory listing
edit	Edit an M-file
fileparts	Get filename parts

---

**Working with Files and the Operating Environment (Continued)**

filebrowser	Display Current Directory browser, for viewing files
fullfile	Build full filename from parts
info	Display contact information or toolbox Readme files
inmem	Functions in memory
ls	List directory on UNIX
matlabroot	Get root directory of MATLAB installation
mkdir	Make new directory
open	Open files based on extension
pwd	Display current directory
tempdir	Return the name of the system's temporary directory
tempname	Unique name for temporary file
undocheckout	Undo previous checkout from source control system
unix	Execute a UNIX command and return the result
!	Execute operating system command

---

**Starting and Quitting MATLAB**

finish	MATLAB termination M-file
exit	Terminate MATLAB
matlab	Start MATLAB (UNIX systems only)
matlabrc	MATLAB startup M-file
quit	Terminate MATLAB
startup	MATLAB startup M-file

## Operators and Special Characters

These are the actual operators you use to enter and manipulate data, for example, matrix multiplication, array multiplication, and line continuation.

Operators and Special Characters	
+	Plus
-	Minus
*	Matrix multiplication
. *	Array multiplication
^	Matrix power
. ^	Array power
kron	Kronecker tensor product
\	Backslash or left division
/	Slash or right division
. / and . \	Array division, right and left
:	Colon
( )	Parentheses
[ ]	Brackets
{ }	Curly braces
.	Decimal point
...	Continuation
,	Comma
;	Semicolon
%	Comment
!	Exclamation point
'	Transpose and quote
. '	Nonconjugated transpose
=	Assignment
==	Equality
< >	Relational operators
&	Logical AND
	Logical OR
~	Logical NOT
xor	Logical EXCLUSIVE OR

## Logical Functions

This set of functions performs logical operations such as checking if a file or variable exists and testing if all elements in an array are nonzero. “Operators and Special Characters” contains other operators that perform logical operations.

Logical Functions	
all	Test to determine if all elements are nonzero
any	Test for any nonzeros
exist	Check if a variable or file exists
find	Find indices and values of nonzero elements
is*	Detect state
isa	Detect an object of a given class
iskeyword	Test if string is a MATLAB keyword
isvarname	Test if string is a valid variable name
logical	Convert numeric values to logical
mi sl ocked	True if M-file cannot be cleared

## Language Constructs and Debugging

These functions let you work with MATLAB as a programming language. For example, you can control program flow, define global variables, perform interactive input, and debug your code.

### MATLAB as a Programming Language

builtin	Execute builtin function from overloaded method
eval	Interpret strings containing MATLAB expressions
evalc	Evaluate MATLAB expression with capture

<b>MATLAB as a Programming Language (Continued)</b>	
<code>evalin</code>	Evaluate expression in workspace
<code>feval</code>	Function evaluation
<code>function</code>	Function M-files
<code>global</code>	Define global variables
<code>nargchk</code>	Check number of input arguments
<code>persistent</code>	Define persistent variable
<code>script</code>	Script M-files

<b>Control Flow</b>	
<code>break</code>	Terminate execution of <code>for</code> loop or <code>while</code> loop
<code>case</code>	Case switch
<code>catch</code>	Begin catch block
<code>continue</code>	Pass control to the next iteration of <code>for</code> or <code>while</code> loop
<code>else</code>	Conditionally execute statements
<code>elseif</code>	Conditionally execute statements
<code>end</code>	Terminate <code>for</code> , <code>while</code> , <code>switch</code> , <code>try</code> , and <code>if</code> statements or indicate last index
<code>error</code>	Display error messages
<code>for</code>	Repeat statements a specific number of times
<code>if</code>	Conditionally execute statements
<code>otherwise</code>	Default part of <code>switch</code> statement
<code>return</code>	Return to the invoking function
<code>switch</code>	Switch among several cases based on expression
<code>try</code>	Begin try block

<b>Control Flow (Continued)</b>	
<code>warning</code>	Display warning message
<code>while</code>	Repeat statements an indefinite number of times

<b>Interactive Input</b>	
<code>input</code>	Request user input
<code>keyboard</code>	Invoke the keyboard in an M-file
<code>menu</code>	Generate a menu of choices for user input
<code>pause</code>	Halt execution temporarily

<b>Object-Oriented Programming</b>	
<code>class</code>	Create object or return class of object
<code>double</code>	Convert to double precision
<code>inferiorto</code>	Inferior class relationship
<code>inline</code>	Construct an inline object
<code>int8</code> , <code>int16</code> , <code>int32</code>	Convert to signed integer
<code>isa</code>	Detect an object of a given class
<code>loadobj</code>	Extends the <code>load</code> function for user objects
<code>saveobj</code>	Save filter for objects
<code>single</code>	Convert to single precision
<code>superiorto</code>	Superior class relationship
<code>uint8</code> , <code>uint16</code> , <code>uint32</code>	Convert to unsigned integer

<b>Debugging</b>	
<code>dbclear</code>	Clear breakpoints
<code>dbcont</code>	Resume execution
<code>dbdown</code>	Change local workspace context
<code>dbmex</code>	Enable MEX-file debugging
<code>dbquit</code>	Quit debug mode
<code>dbstack</code>	Display function call stack
<code>dbstatus</code>	List all breakpoints

<b>Debugging (Continued)</b>	
dbstep	Execute one or more lines from a breakpoint
dbstop	Set breakpoints in an M-file function
dbtype	List M-file with line numbers
dbup	Change local workspace context

### Function Handles

function_handle	MATLAB data type that is a handle to a function
functions	Return information about a function handle
func2str	Constructs a function name string from a function handle
str2func	Constructs a function handle from a function name string

## Elementary Matrices and Matrix Manipulation

Using these functions you can manipulate matrices, and access time, date, special variables, and constants, functions.

### Elementary Matrices and Arrays

blkdiag	Construct a block diagonal matrix from input arguments
eye	Identity matrix
linspace	Generate linearly spaced vectors
logspace	Generate logarithmically spaced vectors
numel	Number of elements in a matrix or cell array
ones	Create an array of all ones
rand	Uniformly distributed random numbers and arrays
randn	Normally distributed random numbers and arrays

### Elementary Matrices and Arrays (Continued)

zeros	Create an array of all zeros
: (colon)	Regularly spaced vector

### Special Variables and Constants

ans	The most recent answer
computer	Identify the computer on which MATLAB is running
eps	Floating-point relative accuracy
i	Imaginary unit
Inf	Infinity
inputname	Input argument name
j	Imaginary unit
NaN	Not-a-Number
nargin, nargout	Number of function arguments
nargoutchk	Validate number of output arguments
pi	Ratio of a circle's circumference to its diameter
realmax	Largest positive floating-point number
realmin	Smallest positive floating-point number
varargin, varargout	Pass or return variable numbers of arguments

### Time and Dates

calendar	Calendar
clock	Current time as a date vector
cputime	Elapsed CPU time
date	Current date string
datenum	Serial date number
datestr	Date string format
datevec	Date components
eomday	End of month
etime	Elapsed time
now	Current date and time

---

**Time and Dates (Continued)**

tic, toc	Stopwatch timer
weekday	Day of the week

---

**Matrix Manipulation**

cat	Concatenate arrays
diag	Diagonal matrices and diagonals of a matrix
flipr	Flip matrices left-right
flipud	Flip matrices up-down
repmat	Replicate and tile an array
reshape	Reshape array
rot90	Rotate matrix 90 degrees
tril	Lower triangular part of a matrix
triu	Upper triangular part of a matrix
: (colon)	Index into array, rearrange array

---

**Vector Functions**

cross	Vector cross product
dot	Vector dot product
intersect	Set intersection of two vectors
ismember	Detect members of a set
setdiff	Return the set difference of two vectors
setxor	Set exclusive or of two vectors
union	Set union of two vectors
unique	Unique elements of a vector

**Specialized Matrices**

These functions let you work with matrices such as Hadamard, Hankel, Hilbert, and magic squares.

---

**Specialized Matrices**

compan	Companion matrix
gallery	Test matrices
hadamard	Hadamard matrix
hankel	Hankel matrix
hilb	Hilbert matrix
invhilb	Inverse of the Hilbert matrix
magic	Magic square
pascal	Pascal matrix
toeplitz	Toeplitz matrix
wilkinson	Wilkinson's eigenvalue test matrix

**Elementary Math Functions**

These are many of the standard mathematical functions such as trigonometric, hyperbolic, logarithmic, and complex number manipulation.

---

**Elementary Math Functions**

abs	Absolute value and complex magnitude
acos, acosh	Inverse cosine and inverse hyperbolic cosine
acot, acoth	Inverse cotangent and inverse hyperbolic cotangent
acsc, acsch	Inverse cosecant and inverse hyperbolic cosecant
angle	Phase angle
asec, asech	Inverse secant and inverse hyperbolic secant
asin, asinh	Inverse sine and inverse hyperbolic sine

---



**Elementary Math Functions (Continued)**

atan, atanh	Inverse tangent and inverse hyperbolic tangent
atan2	Four-quadrant inverse tangent
ceil	Round toward infinity
complex	Construct complex data from real and imaginary components
conj	Complex conjugate
cos, cosh	Cosine and hyperbolic cosine
cot, coth	Cotangent and hyperbolic cotangent
csc, csch	Cosecant and hyperbolic cosecant
exp	Exponential
fix	Round towards zero
floor	Round towards minus infinity
gcd	Greatest common divisor
imag	Imaginary part of a complex number
lcm	Least common multiple
log	Natural logarithm
log2	Base 2 logarithm and dissect floating-point numbers into exponent and mantissa
log10	Common (base 10) logarithm
mod	Modulus (signed remainder after division)
nchoosek	Binomial coefficient or all combinations
real	Real part of complex number
rem	Remainder after division
round	Round to nearest integer
sec, sech	Secant and hyperbolic secant
sign	Signum function
sin, sinh	Sine and hyperbolic sine
sqrt	Square root
tan, tanh	Tangent and hyperbolic tangent

**Specialized Math Functions**

This set of functions includes Bessel, elliptic, gamma, factorial, and others.

**Specialized Math Functions**

airy	Airy functions
besselh	Bessel functions of the third kind (Hankel functions)
besseli, bessely	Modified Bessel functions
besselj, bessely	Bessel functions
beta, betainc, betaln	beta, betainc, betaln
ellipj	Jacobi elliptic functions
ellipke	Complete elliptic integrals of the first and second kind
erf, erfc, erfcx, erfi	Error functions
expint	Exponential integral
factorial	Factorial function
gamma, gammaln, gammai	Gamma functions
legendre	Associated Legendre functions
pow2	Base 2 power and scale floating-point numbers
rat, rats	Rational fraction approximation

## Coordinate System Conversion

Using these functions you can transform Cartesian coordinates to polar, cylindrical, or spherical, and vice versa.

---

### Coordinate System Conversion

---

<code>cart2pol</code>	Transform Cartesian coordinates to polar or cylindrical
<code>cart2sph</code>	Transform Cartesian coordinates to spherical
<code>pol2cart</code>	Transform polar or cylindrical coordinates to Cartesian
<code>sph2cart</code>	Transform spherical coordinates to Cartesian

---

## Matrix Functions - Numerical Linear Algebra

These functions let you perform matrix analysis including matrix determinant, rank, reduced row echelon form, eigenvalues, and inverses.

---

### Matrix Analysis

---

<code>cond</code>	Condition number with respect to inversion
<code>condei g</code>	Condition number with respect to eigenvalues
<code>det</code>	Matrix determinant
<code>norm</code>	Vector and matrix norms
<code>null</code>	Null space of a matrix
<code>orth</code>	Range space of a matrix
<code>rank</code>	Rank of a matrix
<code>rcond</code>	Matrix reciprocal condition number estimate
<code>rref, rrefmvie</code>	Reduced row echelon form
<code>subspace</code>	Angle between two subspaces
<code>trace</code>	Sum of diagonal elements

---



---

### Linear Equations

---

<code>chol</code>	Cholesky factorization
<code>inv</code>	Matrix inverse
<code>lsqov</code>	Least squares solution in the presence of known covariance
<code>lu</code>	LU matrix factorization
<code>lsqnonneg</code>	Nonnegative least squares
<code>minres</code>	Minimum Residual Method
<code>pinv</code>	Moore-Penrose pseudoinverse of a matrix
<code>qr</code>	Orthogonal-triangular decomposition
<code>symml q</code>	Symmetric LQ method

---



---

### Eigenvalues and Singular Values

---

<code>balance</code>	Improve accuracy of computed eigenvalues
<code>cdf2rdf</code>	Convert complex diagonal form to real block diagonal form
<code>eig</code>	Eigenvalues and eigenvectors
<code>gsvd</code>	Generalized singular value decomposition
<code>hess</code>	Hessenberg form of a matrix
<code>poly</code>	Polynomial with specified roots
<code>qz</code>	QZ factorization for generalized eigenvalues
<code>rsf2csf</code>	Convert real Schur form to complex Schur form
<code>schur</code>	Schur decomposition
<code>svd</code>	Singular value decomposition

---



---

### Matrix Functions

---

<code>expm</code>	Matrix exponential
<code>funm</code>	Evaluate general matrix function
<code>logm</code>	Matrix logarithm
<code>sqrtm</code>	Matrix square root

---

**Low Level Functions**

qrdelete	Delete column from QR factorization
qrinsert	Insert column in QR factorization

**Data Analysis and Fourier Transform Functions**

Using the data analysis functions, you can find permutations, prime numbers, mean, median, variance, correlation, and perform convolutions and other standard array manipulations. A set of vector functions lets you operate on vectors to find cross product, union, and other standard vector manipulations. The Fourier transform functions let you perform discrete Fourier transformations in one or more dimensions and their inverses.

**Basic Operations**

cumprod	Cumulative product
cumsum	Cumulative sum
cumtrapz	Cumulative trapezoidal numerical integration
factor	Prime factors
inpolygon	Detect points inside a polygonal region
max	Maximum elements of an array
mean	Average or mean value of arrays
median	Median value of arrays
min	Minimum elements of an array
perms	All possible permutations
polyarea	Area of polygon
primes	Generate list of prime numbers
prod	Product of array elements
rectint	Rectangle intersection area
sort	Sort elements in ascending order
sortrows	Sort rows in ascending order

**Basic Operations (Continued)**

std	Standard deviation
sum	Sum of array elements
trapz	Trapezoidal numerical integration
var	Variance

**Finite Differences**

del2	Discrete Laplacian
diff	Differences and approximate derivatives
gradient	Numerical gradient

**Correlation**

corrcoef	Correlation coefficients
cov	Covariance matrix

**Filtering and Convolution**

conv	Convolution and polynomial multiplication
conv2	Two-dimensional convolution
deconv	Deconvolution and polynomial division
filter	Filter data with an infinite impulse response (IIR) or finite impulse response (FIR) filter
filter2	Two-dimensional digital filtering

**Fourier Transforms**

abs	Absolute value and complex magnitude
angle	Phase angle
cplxpair	Sort complex numbers into complex conjugate pairs
fft	One-dimensional fast Fourier transform

**Fourier Transforms (Continued)**

<code>fft2</code>	Two-dimensional fast Fourier transform
<code>fftshift</code>	Shift DC component of fast Fourier transform to center of spectrum
<code>ifft</code>	Inverse one-dimensional fast Fourier transform
<code>ifft2</code>	Inverse two-dimensional fast Fourier transform
<code>ifftn</code>	Inverse multidimensional fast Fourier transform
<code>ifftshift</code>	Inverse FFT shift
<code>nextpow2</code>	Next power of two
<code>unwrap</code>	Correct phase angles

**Vector Functions**

<code>cross</code>	Vector cross product
<code>intersect</code>	Set intersection of two vectors
<code>ismember</code>	Detect members of a set
<code>setdiff</code>	Return the set difference of two vector
<code>setxor</code>	Set exclusive or of two vectors
<code>union</code>	Set union of two vectors
<code>unique</code>	Unique elements of a vector

**Polynomial and Interpolation Functions**

These functions let you operate on polynomials such as multiply, divide, find derivatives, and evaluate. The data interpolation functions let you perform interpolation in one, two, three, and higher dimensions.

**Polynomials**

<code>conv</code>	Convolution and polynomial multiplication
<code>deconv</code>	Deconvolution and polynomial division
<code>poly</code>	Polynomial with specified roots
<code>polyder</code>	Polynomial derivative
<code>polyeig</code>	Polynomial eigenvalue problem
<code>polyfit</code>	Polynomial curve fitting
<code>polyint</code>	Analytic polynomial integration
<code>polyval</code>	Polynomial evaluation
<code>polyvalm</code>	Matrix polynomial evaluation
<code>residue</code>	Convert between partial fraction expansion and polynomial coefficients
<code>roots</code>	Polynomial roots

**Data Interpolation**

<code>convhull</code>	Convex hull
<code>convhulln</code>	Multidimensional convex hull
<code>delaunay</code>	Delaunay triangulation
<code>delaunay3</code>	3-D Delaunay tessellation
<code>delaunayn</code>	Multidimensional Delaunay tessellation
<code>dsearch</code>	Search for nearest point
<code>dsearchn</code>	Multidimensional closest point search
<code>griddata</code>	Data gridding
<code>griddata3</code>	Data gridding and hypersurface fitting for three-dimensional data
<code>griddata_n</code>	Data gridding and hypersurface fitting (dimension $\geq 2$ )
<code>interp1</code>	One-dimensional data interpolation (table lookup)
<code>interp2</code>	Two-dimensional data interpolation (table lookup)

<b>Data Interpolation (Continued)</b>	
<code>interp3</code>	Three-dimensional data interpolation (table lookup)
<code>interpft</code>	One-dimensional interpolation using the FFT method
<code>interpn</code>	Multidimensional data interpolation (table lookup)
<code>meshgrid</code>	Generate X and Y matrices for three-dimensional plots
<code>ndgrid</code>	Generate arrays for multidimensional functions and interpolation
<code>pchip</code>	Piecewise Cubic Hermite Interpolating Polynomial (PCHIP)
<code>ppval</code>	Piecewise polynomial evaluation
<code>spline</code>	Cubic spline interpolation
<code>tsearch</code>	Search for enclosing Delaunay triangle
<code>tsearchn</code>	Multidimensional closest simplex search
<code>voronoi</code>	Voronoi diagram
<code>voronoin</code>	Multidimensional Voronoi diagrams

## Function Functions - Nonlinear Numerical Methods

Using these functions you can solve differential equations, perform numerical evaluation of integrals, and optimize functions.

<b>Function Functions - Nonlinear Numerical Methods</b>	
<code>bvp4c</code>	Solve two-point boundary value problems (BVPs) for ordinary differential equations (ODEs)
<code>bvpget</code>	Extract parameters from BVP options structure
<code>bvpinit</code>	Form the initial guess for <code>bvp4c</code>

<b>Function Functions - Nonlinear Numerical Methods (Continued)</b>	
<code>bvpset</code>	Create/alter BVP options structure
<code>bvpval</code>	Evaluate the solution computed by <code>bvp4c</code>
<code>dblquad</code>	Numerical evaluation of double integrals
<code>fminbnd</code>	Minimize a function of one variable
<code>fminsearch</code>	Minimize a function of several variables
<code>fzero</code>	Find zero of a function of one variable
<code>ode45</code> , <code>ode23</code> , <code>ode113</code> , <code>ode15s</code> , <code>ode23s</code> , <code>ode23t</code> , <code>ode23tb</code>	Solve initial value problems for ODEs
<code>odeget</code>	Extract parameters from ODE options structure
<code>odeset</code>	Create/alter ODE options structure
<code>optimget</code>	Get optimization options structure parameter values
<code>optimset</code>	Create or edit optimization options parameter structure
<code>pdepe</code>	Solve initial-boundary value problems
<code>pdeval</code>	Evaluate the solution computed by <code>pdepe</code>
<code>quad</code>	Numerical evaluation of integrals, adaptive Simpson quadrature
<code>quadl</code>	Numerical evaluation of integrals, adaptive Lobatto quadrature
<code>vectorize</code>	Vectorize expression

## Sparse Matrix Functions

These functions allow you to operate on a special type of matrix, sparse. Using these functions you can convert full to sparse, visualize, and operate on these matrices.

### Elementary Sparse Matrices

spdiags	Extract and create sparse band and diagonal matrices
speye	Sparse identity matrix
sprand	Sparse uniformly distributed random matrix
sprandn	Sparse normally distributed random matrix
sprandsym	Sparse symmetric random matrix

### Full to Sparse Conversion

find	Find indices and values of nonzero elements
full	Convert sparse matrix to full matrix
sparse	Create sparse matrix
spconvert	Import matrix from sparse matrix external format

### Working with Nonzero Entries of Sparse Matrices

nnz	Number of nonzero matrix elements
nonzeros	Nonzero matrix elements
nzmax	Amount of storage allocated for nonzero matrix elements
spalloc	Allocate space for sparse matrix
spfun	Apply function to nonzero sparse matrix elements
spones	Replace nonzero sparse matrix elements with ones

### Visualizing Sparse Matrices

spy	Visualize sparsity pattern
-----	----------------------------

### Reordering Algorithms

colamd	Column approximate minimum degree permutation
colmmd	Sparse column minimum degree permutation
colperm	Sparse column permutation based on nonzero count
dmperm	Dulmage-Mendelsohn decomposition
randperm	Random permutation
symamd	Symmetric approximate minimum degree permutation
symmmd	Sparse symmetric minimum degree ordering
symrcm	Sparse reverse Cuthill-McKee ordering

### Norm, Condition Number, and Rank

condest	1-norm matrix condition number estimate
normest	2-norm estimate

### Sparse Systems of Linear Equations

bicg	BiConjugate Gradients method
bicgstab	BiConjugate Gradients Stabilized method
cgs	Conjugate Gradients Squared method
cholinc	Sparse Incomplete Cholesky and Cholesky-Infinity factorizations
cholupdate	Rank 1 update to Cholesky factorization
gmres	Generalized Minimum Residual method (with restarts)

**Sparse Systems of Linear Equations (Continued)**

lsqr	LSQR implementation of Conjugate Gradients on the normal equations
luis	Incomplete LU matrix factorizations
pcg	Preconditioned Conjugate Gradients method
qmr	Quasi-Minimal Residual method
qr	Orthogonal-triangular decomposition
qrdelete	Delete column from QR factorization
qrintert	Insert column in QR factorization
qrupdate	Rank 1 update to QR factorization

**Sparse Eigenvalues and Singular Values**

eigs	Find eigenvalues and eigenvectors
svds	Find singular values

**Miscellaneous**

spparms	Set parameters for sparse matrix routines
---------	---

**Sound Processing Functions**

The sound processing functions let you convert signals, and read and write . au and . wav sound files.

**General Sound Functions**

lin2mu	Convert linear audio signal to mu-law
mu2lin	Convert mu-law audio signal to linear
sound	Convert vector into sound
soundsc	Scale data and play as sound

**SPARCstation-Specific Sound Functions**

auread	Read NeXT/SUN (. au) sound file
auwrite	Write NeXT/SUN (. au) sound file

**.WAV Sound Functions**

wavplay	Play recorded sound on a PC-based audio output device
wavread	Read Microsoft WAVE (. wav) sound file
wavrecord	Record sound using a PC-based audio input device
wavwrite	Write Microsoft WAVE (. wav) sound file

## Character String Functions

This set of functions lets you manipulate strings such as comparison, concatenation, search, and conversion.

### General

abs	Absolute value and complex magnitude
eval	Interpret strings containing MATLAB expressions
real	Real part of complex number
strings	MATLAB string handling

### String to Function Handle Conversion

func2str	Constructs a function name string from a function handle
str2func	Constructs a function handle from a function name string

### String Manipulation

deblank	Strip trailing blanks from the end of a string
findstr	Find one string within another
lower	Convert string to lower case
strcat	String concatenation
strcmp	Compare strings
strcmpi	Compare strings ignoring case
strjust	Justify a character array
strmatch	Find possible matches for a string
strncmp	Compare the first n characters of two strings
strncmpi	Compare the first n characters of strings, ignoring case
strrep	String search and replace
strtok	First token in string
strvcat	Vertical concatenation of strings

### String Manipulation (Continued)

symvar	Determine symbolic variables in an expression
texlab	Produce the TeX format from a character string
upper	Convert string to upper case

### String to Number Conversion

char	Create character array (string)
int2str	Integer to string conversion
mat2str	Convert a matrix into a string
num2str	Number to string conversion
sprintf	Write formatted data to a string
sscanf	Read string under format control
str2double	Convert string to double-precision value
str2mat	String to matrix conversion
str2num	String to number conversion

### Radix Conversion

bin2dec	Binary to decimal number conversion
dec2bin	Decimal to binary number conversion
dec2hex	Decimal to hexadecimal number conversion
hex2dec	Hexadecimal to decimal number conversion
hex2num	Hexadecimal to double number conversion



## File I/O Functions

The file I/O functions allow you to open and close files, read and write formatted and unformatted data, operate on files, and perform other specialized file I/O such as reading and writing images and spreadsheets.

### File Opening and Closing

<code>fclose</code>	Close one or more open files
<code>fopen</code>	Open a file or obtain information about open files

### Unformatted I/O

<code>fread</code>	Read binary data from file
<code>fwrite</code>	Write binary data to a file

### Formatted I/O

<code>fgetl</code>	Return the next line of a file as a string without line terminator(s)
<code>fgets</code>	Return the next line of a file as a string with line terminator(s)
<code>fprintf</code>	Write formatted data to file
<code>fscanf</code>	Read formatted data from file

### File Positioning

<code>feof</code>	Test for end-of-file
<code>ferror</code>	Query MATLAB about errors in file input or output
<code>frewind</code>	Rewind an open file
<code>fseek</code>	Set file position indicator
<code>ftell</code>	Get file position indicator

### String Conversion

<code>sprintf</code>	Write formatted data to a string
<code>sscanf</code>	Read string under format control

### Specialized File I/O

<code>dlmread</code>	Read an ASCII delimited file into a matrix
<code>dlmwrite</code>	Write a matrix to an ASCII delimited file
<code>hdf</code>	HDF interface
<code>imfinfo</code>	Return information about a graphics file
<code>imread</code>	Read image from graphics file
<code>imwrite</code>	Write an image to a graphics file
<code>strread</code>	Read formatted data from a string
<code>textread</code>	Read formatted data from text file
<code>wk1read</code>	Read a Lotus123 WK1 spreadsheet file into a matrix
<code>wk1write</code>	Write a matrix to a Lotus123 WK1 spreadsheet file

## Bitwise Functions

These functions let you operate at the bit level such as shifting and complementing.

### Bitwise Functions

<code>bitand</code>	Bit-wise AND
<code>bitcmp</code>	Complement bits
<code>bitor</code>	Bit-wise OR
<code>bitmax</code>	Maximum floating-point integer
<code>bitset</code>	Set bit
<code>bitshift</code>	Bit-wise shift
<code>bitget</code>	Get bit
<code>bitxor</code>	Bit-wise XOR

## Structure Functions

Structures are arrays whose elements can hold any MATLAB data type such as text, numeric arrays, or other structures. You access structure elements by name. Use the structure functions to create and operate on this array type.

---

### Structure Functions

<code>deal</code>	Deal inputs to outputs
<code>fieldnames</code>	Field names of a structure
<code>getfield</code>	Get field of structure array
<code>rmfield</code>	Remove structure fields
<code>setfield</code>	Set field of structure array
<code>struct</code>	Create structure array
<code>struct2cell</code>	Structure to cell array conversion

## MATLAB Object Functions

Using the object functions you can create objects, detect objects of a given class, and return the class of an object.

---

### Object Functions

<code>class</code>	Create object or return class of object
<code>isa</code>	Detect an object of a given class
<code>methods</code>	Display method names
<code>methodsview</code>	Displays information on all methods implemented by a class
<code>subsasgn</code>	Overloaded method for $A(I)=B$ , $A\{I\}=B$ , and $A.field=B$
<code>subsindex</code>	Overloaded method for $X(A)$
<code>subsref</code>	Overloaded method for $A(I)$ , $A\{I\}$ and $A.field$

## MATLAB Interface to Java Functions

These functions allow you to bring Java classes into MATLAB, construct objects, and call and save methods.

---

### Interface to Java Functions

<code>class</code>	Create object or return class of object
<code>import</code>	Add a package or class to the current Java import list
<code>isa</code>	Detect an object of a given class
<code>isjava</code>	Test whether an object is a Java object
<code>javaArray</code>	Constructs a Java array
<code>javaMethod</code>	Invokes a Java method
<code>javaObject</code>	Constructs a Java object
<code>methods</code>	Display method names
<code>methodsview</code>	Display information on all methods implemented by a class

## Cell Array Functions

Cell arrays are arrays comprised of cells, which can hold any MATLAB data type such as text, numeric arrays, or other cell arrays. Unlike structures, you access these cells by number. Use the cell array functions to create and operate on these arrays.

Cell Array Functions	
<code>cell</code>	Create cell array
<code>cellfun</code>	Apply a function to each element in a cell array
<code>cellstr</code>	Create cell array of strings from character array
<code>cell2struct</code>	Cell array to structure array conversion
<code>celldisp</code>	Display cell array contents
<code>cellplot</code>	Graphically display the structure of cell arrays
<code>num2cell</code>	Convert a numeric array into a cell array

## Multidimensional Array Functions

These functions provide a mechanism for working with arrays of dimension greater than 2.

Multidimensional Array Functions	
<code>cat</code>	Concatenate arrays
<code>flipdim</code>	Flip array along a specified dimension
<code>ind2sub</code>	Subscripts from linear index
<code>ipermute</code>	Inverse permute the dimensions of a multidimensional array
<code>ndgrid</code>	Generate arrays for multidimensional functions and interpolation
<code>ndims</code>	Number of array dimensions

## Multidimensional Array Functions (Continued)

<code>permute</code>	Rearrange the dimensions of a multidimensional array
<code>reshape</code>	Reshape array
<code>shiftdim</code>	Shift dimensions
<code>squeeze</code>	Remove singleton dimensions
<code>sub2ind</code>	Single index from subscripts

## Data Visualization

This extensive set of functions gives you the ability to create basic graphs such as bar, pie, polar, and three-dimensional plots, and advanced graphs such as surface, mesh, contour, and volume visualization plots. In addition, you can use these functions to control lighting, color, view, and many other fine manipulations.

### Basic Plots and Graphs

<code>bar</code>	Vertical bar chart
<code>barh</code>	Horizontal bar chart
<code>hist</code>	Plot histograms
<code>histc</code>	Histogram count
<code>hold</code>	Hold current graph
<code>loglog</code>	Plot using log-log scales
<code>pie</code>	Pie plot
<code>plot</code>	Plot vectors or matrices.
<code>polar</code>	Polar coordinate plot
<code>semilogx</code>	Semi-log scale plot
<code>semilogy</code>	Semi-log scale plot
<code>subplot</code>	Create axes in tiled positions

### Three-Dimensional Plotting

<code>bar3</code>	Vertical 3-D bar chart
<code>bar3h</code>	Horizontal 3-D bar chart
<code>comet3</code>	3-D comet plot
<code>cylinder</code>	Generate cylinder

<b>Three-Dimensional Plotting (Continued)</b>	
<code>fill3</code>	Draw filled 3-D polygons in 3-space
<code>plot3</code>	Plot lines and points in 3-D space
<code>quiver3</code>	Three-dimensional quiver (or velocity) plot
<code>slice</code>	Volumetric slice plot
<code>sphere</code>	Generate sphere
<code>stem3</code>	Plot discrete surface data
<code>waterfall</code>	Waterfall plot

### Plot Annotation and Grids

<code>clabel</code>	Add contour labels to a contour plot
<code>datetick</code>	Date formatted tick labels
<code>grid</code>	Grid lines for 2-D and 3-D plots
<code>gtext</code>	Place text on a 2-D graph using a mouse
<code>legend</code>	Graph legend for lines and patches
<code>plotedit</code>	Start plot edit mode to edit and annotate plots
<code>plotyy</code>	Plot graphs with Y tick labels on the left and right
<code>title</code>	Titles for 2-D and 3-D plots
<code>xlabel</code>	X-axis labels for 2-D and 3-D plots
<code>ylabel</code>	Y-axis labels for 2-D and 3-D plots
<code>zlabel</code>	Z-axis labels for 3-D plots

### Surface, Mesh, and Contour Plots

<code>contour</code>	Contour (level curves) plot
<code>contourc</code>	Contour computation
<code>contourf</code>	Filled contour plot
<code>hidden</code>	Mesh hidden line removal mode
<code>meshc</code>	Combination mesh/contourplot

<b>Surface, Mesh, and Contour Plots (Continued)</b>	
<code>mesh</code>	3-D mesh with reference plane
<code>peaks</code>	A sample function of two variables
<code>surf</code>	3-D shaded surface graph
<code>surface</code>	Create surface low-level objects
<code>surfc</code>	Combination surf/contourplot
<code>surfll</code>	3-D shaded surface with lighting
<code>trimesh</code>	Triangular mesh plot
<code>trisurf</code>	Triangular surface plot

### Volume Visualization

<code>coneplot</code>	Plot velocity vectors as cones in 3-D vector field
<code>contourslice</code>	Draw contours in volume slice plane
<code>curl</code>	Compute the curl and angular velocity of a vector field
<code>divergence</code>	Compute the divergence of a vector field
<code>flow</code>	Generate scalar volume data
<code>interpstreamspeed</code>	Interpolate streamline vertices from vector-field magnitudes
<code>isocaps</code>	Compute isosurface end-cap geometry
<code>isocolors</code>	Compute the colors of isosurface vertices
<code>isonormals</code>	Compute normals of isosurface vertices
<code>isosurface</code>	Extract isosurface data from volume data
<code>reducepatch</code>	Reduce the number of patch faces
<code>reducevolume</code>	Reduce number of elements in volume data set
<code>shrinkfaces</code>	Reduce the size of patch faces
<code>slice</code>	Draw slice planes in volume
<code>smooth3</code>	Smooth 3-D data

<b>Volume Visualization (Continued)</b>	
stream2	Compute 2-D stream line data
stream3	Compute 3-D stream line data
streamline	Draw stream lines from 2- or 3-D vector data
streamparticles	Draw stream particles from vector volume data
streamribbon	Draw stream ribbons from vector volume data
streamslice	Draw well-spaced stream lines from vector volume data
streamtube	Draw stream tubes from vector volume data
surf2patch	Convert surface data to patch data
subvolume	Extract subset of volume data set

### **Domain Generation**

griddata	Data gridding and surface fitting
meshgrid	Generation of X and Y arrays for 3-D plots

### **Specialized Plotting**

area	Area plot
box	Axis box for 2-D and 3-D plots
comet	Comet plot
compass	Compass plot
convhull	Convex hull
delaunay	Delaunay triangulation
dsearch	Search Delaunay triangulation for nearest point
errorbar	Plot graph with error bars
ezcontour	Easy to use contour plotter
ezcontourf	Easy to use filled contour plotter
ezmesh	Easy to use 3-D mesh plotter

<b>Specialized Plotting (Continued)</b>	
ezmeshc	Easy to use combination mesh/contour plotter
ezplot	Easy to use function plotter
ezplot3	Easy to use 3-D parametric curve plotter
ezpolar	Easy to use polar coordinate plotter
ezsurf	Easy to use 3-D colored surface plotter
ezsurf c	Easy to use combination surface/contour plotter
feather	Feather plot
fill	Draw filled 2-D polygons
fplot	Plot a function
inpolygon	True for points inside a polygonal region
pareto	Pareto char
pcolor	Pseudocolor (checkerboard) plot
pie3	3-D pie plot
plotmatrix	Scatter plot matrix
pol yarea	Area of polygon
quiver	Quiver (or velocity) plot
ribbon	Ribbon plot
rose	Plot rose or angle histogram
scatter	Scatter plot
scatter3	3-D scatter plot
stairs	Stairstep graph
stem	Plot discrete sequence data
tsearch	Search for enclosing Delaunay triangle
voronoi	Voronoi diagram

<b>View Control</b>	
<code>camdolly</code>	Move camera position and target
<code>camlookat</code>	View specific objects
<code>camorbit</code>	Orbit about camera target
<code>campan</code>	Rotate camera target about camera position
<code>campos</code>	Set or get camera position
<code>camproj</code>	Set or get projection type
<code>camroll</code>	Rotate camera about viewing axis
<code>camtarget</code>	Set or get camera target
<code>camup</code>	Set or get camera up-vector
<code>camva</code>	Set or get camera view angle
<code>camzoom</code>	Zoom camera in or out
<code>daspect</code>	Set or get data aspect ratio
<code>pbaspect</code>	Set or get plot box aspect ratio
<code>view</code>	3-D graph viewpoint specification.
<code>viewmtx</code>	Generate view transformation matrices
<code>xlim</code>	Set or get the current x-axis limits
<code>ylim</code>	Set or get the current y-axis limits
<code>zlim</code>	Set or get the current z-axis limits

<b>Lighting</b>	
<code>camlight</code>	Create or position a light
<code>light</code>	Light object creation function
<code>lightangle</code>	Spherical position of a light
<code>lighting</code>	Lighting mode
<code>material</code>	Material reflectance mode

<b>Transparency</b>	
<code>alpha</code>	Set or query transparency properties for objects in current axes
<code>alphamap</code>	Specify the figure alphamap
<code>alpha</code>	Set or query the axes alpha limits

<b>Color Operations</b>	
<code>brighten</code>	Brighten or darken color map
<code>caxis</code>	Pseudocolor axis scaling
<code>colorbar</code>	Display color bar (color scale)
<code>colordf</code>	Set up color defaults
<code>colormap</code>	Set the color look-up table (list of colormaps)
<code>graymon</code>	Graphics figure defaults set for grayscale monitor
<code>hsv2rgb</code>	Hue-saturation-value to red-green-blue conversion
<code>rgb2hsv</code>	RGB to HSV conversion
<code>rgbplot</code>	Plot color map
<code>shading</code>	Color shading mode
<code>spinmap</code>	Spin the colormap
<code>surfnorm</code>	3-D surface normals
<code>whitbg</code>	Change axes background color for plots

<b>Colormaps</b>	
<code>autumn</code>	Shades of red and yellow color map
<code>bone</code>	Gray-scale with a tinge of blue color map
<code>contrast</code>	Gray color map to enhance image contrast
<code>cool</code>	Shades of cyan and magenta color map
<code>copper</code>	Linear copper-tone color map

**Colormaps (Continued)**

flag	Alternating red, white, blue, and black color map
gray	Linear gray-scale color map
hot	Black-red-yellow-white color map
hsv	Hue-saturation-value (HSV) color map
jet	Variant of HSV
lines	Line color colormap
prism	Colormap of prism colors
spring	Shades of magenta and yellow color map
summer	Shades of green and yellow colormap
winter	Shades of blue and green color map

**Printing**

orient	Hardcopy paper orientation
pagesetupdlg	Page position dialog box
print	Print graph or save graph to file
printdlg	Print dialog box
printopt	Configure local printer defaults
saveas	Save figure to graphic file

**Handle Graphics, General**

allchild	Find all children of specified objects
copyobj	Make a copy of a graphics object and its children
findall	Find all graphics objects (including hidden handles)
findobj	Find objects with specified property values
gcbo	Return object whose callback is currently executing
gco	Return handle of current object

**Handle Graphics, General (Continued)**

get	Get object properties
ishandle	True for graphics objects
rotate	Rotate objects about specified origin and direction
set	Set object properties

**Working with Application Data**

getappdata	Get value of application data
isappdata	True if application data exists
rmappdata	Remove application data
setappdata	Specify application data

**Handle Graphics, Object Creation**

axes	Create axes object
figure	Create figure (graph) windows
image	Create image (2-D matrix)
light	Create light object (illuminates patch and surface)
line	Create line object (3-D polylines)
patch	Create patch object (polygons)
rectangle	Create rectangle object (2-D rectangle)
surface	Create surface (quadrilaterals)
text	Create text object (character strings)
uicontextmenu	Create context menu (pop-up associated with object)

**Handle Graphics, Figure Windows**

capture	Screen capture of the current figure
clc	Clear figure window
clf	Clear figure
close	Close specified window
closereq	Default close request function
gcf	Get current figure handle

**Handle Graphics, Figure Windows (Continued)**

newplot	Graphics M-file preamble for NextPlot property
refresh	Refresh figure
saveas	Save figure or model to desired output format

**Handle Graphics, Axes**

axis	Plot axis scaling and appearance
cla	Clear axes
gca	Get current axes handle

**Object Manipulation**

reset	Reset axis or figure
rotate3d	Interactively rotate the view of a 3-D plot
selectmoveresize	Interactively select, move, or resize objects

**Interactive User Input**

ginput	Graphical input from a mouse or cursor
zoom	Zoom in and out on a 2-D plot

**Region of Interest**

dragrect	Drag XOR rectangles with mouse
drawnow	Complete any pending drawing
rbbox	Rubberband box

**Graphical User Interfaces**

The graphical user interface functions let you build your own interfaces for your applications.

**Dialog Boxes**

dialog	Create a dialog box
errordlg	Create error dialog box
helpdlg	Display help dialog box
inputdlg	Create input dialog box
listdlg	Create list selection dialog box
msgbox	Create message dialog box
pagedlg	Display page layout dialog box
printdlg	Display print dialog box
questdlg	Create question dialog box
uigetfile	Display dialog box to retrieve name of file for reading
uiputfile	Display dialog box to retrieve name of file for writing
uisetcolor	Interactively set a ColorSpec using a dialog box
uisetfont	Interactively set a font using a dialog box
warndlg	Create warning dialog box

**User Interface Deployment**

guidata	Store or retrieve application data
guihandles	Create a structure of handles
movegui	Move GUI figure onscreen
openfig	Open or raise GUI figure

**User Interface Development**

guide	Open the GUI Layout Editor
inspect	Display Property Inspector



**User Interface Objects**

menu	Generate a menu of choices for user input
ui context menu	Create context menu
ui control	Create user interface control
ui menu	Create user interface menu

**Other Functions**

dragrect	Drag rectangles with mouse
findfigs	Display off-screen visible figure windows
gcbf	Return handle of figure containing callback object
gcbo	Return handle of object whose callback is executing
rbbox	Create rubberband box for area selection
selectmoveresize	Select, move, resize, or copy axes and uicontrol graphics objects
textwrap	Return wrapped string matrix for given uicontrol
ui resume	Used with ui wait, controls program execution
ui wait	Used with ui resume, controls program execution
waitbar	Display wait bar
waitforbuttonpress	Wait for key/buttonpress over figure

**Serial Port I/O**

These functions provides direct access to peripheral devices that you connect to your computer's serial port.

**Creating a Serial Port Object**

serial	Create a serial port object
--------	-----------------------------

**Writing and Reading Data**

fgetl	Read one line of text from the device and discard the terminator
fgets	Read one line of text from the device and include the terminator
fpri ntf	Write text to the device
fread	Read binary data from the device
fscanf	Read data from the device, and format as text
fwri te	Write binary data to the device
readasync	Read data asynchronously from the device
stopasync	Stop asynchronous and write operations

**Configuring and Returning Properties**

get	Return serial port object properties
set	Configure or display serial port object properties

**State Change**

fcl ose	Disconnect a serial port object from the device
fopen	Connect a serial port object to the device
record	Record data and event information to a file

---

**General Purpose**

---

<code>clear</code>	Remove a serial port object from the MATLAB workspace
<code>delete</code>	Remove a serial port object from memory
<code>disp</code>	Display serial port object summary information
<code>instraction</code>	Display event information when an event occurs
<code>instrfind</code>	Return serial port objects from memory to the MATLAB workspace
<code>isvalid</code>	Determine if serial port objects are valid
<code>length</code>	Length of serial port object array
<code>load</code>	Load serial port objects and variables into the MATLAB workspace
<code>save</code>	Save serial port objects and variables to a MAT-file
<code>serialbreak</code>	Send a break to the device connected to the serial port
<code>size</code>	Size of serial port object array

---