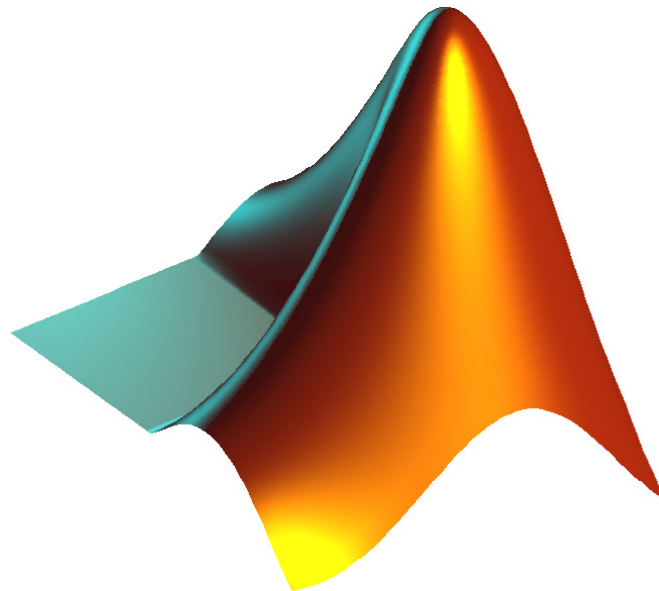


ZUSAMMENFASSUNG UND KURZBESCHREIBUNG ALLER MATLAB-BEFEHLE VERSION 2007A

(und wichtiger Toolboxen)

Stefan Wicki



Versionen:

MATLAB	Version 7.4	(R2007a)
Simulink	Version 6.6	(R2007a)
Optimization Toolbox	Version 3.1.1	(R2007a)
Signal Processing Toolbox	Version 6.7	(R2007a)
Symbolic Math Toolbox	Version 3.2	(R2007a)

Filename: Befehlssatz MATLAB Version 2007a_1_0.doc	Version: 1.0	Author: Stefan Wicki
Created: 20.02.2008	Last modified: 21.02.2008 21:30	Page: 1 / 64

Inhaltsverzeichnis:

1.	MATLAB	4
1.1	Audio and Video support	4
1.2	Commands for creating and debugging code	5
1.3	Data analysis and Fourier transforms	6
1.4	Data types and structures	7
1.5	Elementary math functions	10
1.6	Elementary matrices and matrix manipulation	12
1.7	Function functions and ODE solvers	14
1.8	General purpose commands	16
1.9	Two dimensional graphs	18
1.10	Three dimensional graphs	19
1.11	Handle Graphics	22
1.12	Help commands	24
1.13	Image and scientific data input/output	24
1.14	File input and output	26
1.15	Programming language constructs	28
1.16	Matrix functions - numerical linear algebra	30
1.17	Operators and special characters	31
1.18	Interpolation and polynomials	33
1.19	Sparse matrices	34
1.20	Specialized math functions	36
1.21	Specialized math functions	38
1.22	Specialized graphs	39
1.23	Character strings	42
1.24	Time and dates	43
1.25	Graphical User Interface Tools	44
1.26	Windows Operating System Interface Files (COM/DDE)	46
2.	Simulink	47
2.1	Simulink block library	47
2.2	Simulink components	47
2.3	Simulink Fixed Point utilities	48
2.4	Simulink demonstrations and examples	48
2.5	Simulink	49

3.	Optimization Toolbox	51
4.	Signal Processing Toolbox	52
4.1	Discrete-time filter design, analysis, and implementation	52
4.2	Analog filter design, transformation, and discretization	54
4.3	Linear system transformations	55
4.4	Windows	55
4.5	Transforms	56
4.6	Cepstral analysis	56
4.7	Statistical signal processing and spectral analysis	57
4.8	Parametric modeling	57
4.9	Linear prediction	58
4.10	Multirate signal processing	58
4.11	Waveform generation	58
4.12	Specialized operations	59
4.13	Graphical User Interfaces	59
5.	Symbolic Math Toolbox	60

1. MATLAB

1.1 Audio and Video support

Audio input/output objects.

- `audioplayer` - Audio player object.
- `audiorecorder` - Audio recorder object.

Audio hardware drivers.

- `sound` - Play vector as sound.
- `soundsc` - Autoscale and play vector as sound.
- `wavplay` - Play sound using Windows audio output device.
- `wavrecord` - Record sound using Windows audio input device.

Audio file import and export.

- `aufinfo` - Return information about AU file.
- `auread` - Read NeXT/SUN (".au") sound file.
- `auwrite` - Write NeXT/SUN (".au") sound file.
- `wavfinfo` - Return information about WAV file.
- `wavread` - Read Microsoft WAVE (".wav") sound file.
- `wavwrite` - Write Microsoft WAVE (".wav") sound file.

Video file import/export.

- `aviread` - Read movie (AVI) file.
- `aviinfo` - Return information about AVI file.
- `avifile` - Create a new AVI file.
- `mmfileinfo` - Return information for a Windows multimedia file.
- `movie2avi` - Create AVI movie from MATLAB movie.

Utilities.

- `lin2mu` - Convert linear signal to mu-law encoding.
- `mu2lin` - Convert mu-law encoding to linear signal.

Example audio data (MAT files).

- `chirp` - Frequency sweeps (1.6 sec, 8192 Hz)
- `gong` - Gong (5.1 sec, 8192 Hz)
- `handel` - Hallelujah chorus (8.9 sec, 8192 Hz)
- `laughter` - Laughter from a crowd (6.4 sec, 8192 Hz)
- `splat` - Chirp followed by a splat (1.2 sec, 8192 Hz)
- `train` - Train whistle (1.5 sec, 8192 Hz)

See also `imagesci`, `iofun`.

1.2 Commands for creating and debugging code

Writing and managing M-files.

<code>edit</code>	- Edit M-file.
<code>notebook</code>	- Open an m-book in Microsoft Word (Windows only).
<code>mlint</code>	- Display inconsistencies and suspicious constructs in M-files.
<code>publish</code>	- Run a script and save the results.
<code>grabcode</code>	- Pull M-code from MATLAB-generated HTML demo files.

Directory tools

<code>contentsrpt</code>	- Audit the Contents.m for the given directory
<code>coveragerpt</code>	- Scan a directory for profiler line coverage.
<code>deprpt</code>	- Scan a file or directory for dependencies.
<code>diffprt</code>	- Visual directory browser
<code>dofixrpt</code>	- Scan a file or directory for all TODO, FIXME, or NOTE messages.
<code>helpprt</code>	- Scan a file or directory for help.
<code>mlintrpt</code>	- Scan a file or directory for all M-Lint messages.
<code>standardrpt</code>	- Visual directory browser

Profiling M-files.

<code>profile</code>	- Profile function execution time.
<code>profview</code>	- Display HTML profiler interface.
<code>profsave</code>	- Save a static version of the HTML profile report
<code>profreport</code>	- Generate profile report.
<code>profviewgateway</code>	- Profiler HTML gateway function.
<code>opentoline</code>	- Open to specified line in function file in the editor.
<code>stripanchors</code>	- Remove anchors that evaluate MATLAB code from Profiler HTML.

Debugging M-files.

<code>debug</code>	- Debugging commands.
<code>dbstop</code>	- Set breakpoint.
<code>dbclear</code>	- Remove breakpoint.
<code>dbcont</code>	- Continue execution.
<code>dbdown</code>	- Change local workspace context.
<code>dbstack</code>	- Display function call stack.
<code>dbstatus</code>	- List all breakpoints.
<code>dbstep</code>	- Execute one or more lines.
<code>dbtype</code>	- List M-file with line numbers.
<code>dbup</code>	- Change local workspace context.
<code>dbquit</code>	- Quit debug mode.
<code>dbmex</code>	- Debug MEX-files (UNIX only).

Managing, watching, and editing variables.

<code>openvar</code>	- Open a workspace variable for graphical editing.
<code>workspace</code>	- View the contents of a workspace.

Managing the file system and search path.

- `filebrowser` - Open the Current Directory browser or bring it to the front.
- `pathtool` - View, modify, and save the MATLAB search path.

Command Window and Command History window.

- `commandwindow` - Open the Command Window or bring it to the front.
- `commandhistory` - Open the Command History window or bring it to the front.

1.3 Data analysis and Fourier transforms

Basic operations.

- `max` - Largest component.
- `min` - Smallest component.
- `mean` - Average or mean value.
- `median` - Median value.
- `std` - Standard deviation.
- `var` - Variance.
- `sort` - Sort in ascending order.
- `sortrows` - Sort rows in ascending order.
- `sum` - Sum of elements.
- `prod` - Product of elements.
- `hist` - Histogram.
- `histc` - Histogram count.
- `trapz` - Trapezoidal numerical integration.
- `cumsum` - Cumulative sum of elements.
- `cumprod` - Cumulative product of elements.
- `cumtrapz` - Cumulative trapezoidal numerical integration.

Finite differences.

- `diff` - Difference and approximate derivative.
- `gradient` - Approximate gradient.
- `del2` - Discrete Laplacian.

Correlation.

- `corrcoef` - Correlation coefficients.
- `cov` - Covariance matrix.
- `subspace` - Angle between subspaces.

Filtering and convolution.

- `filter` - One-dimensional digital filter.
- `filter2` - Two-dimensional digital filter.
- `conv` - Convolution and polynomial multiplication.
- `conv2` - Two-dimensional convolution.
- `convn` - N-dimensional convolution.
- `deconv` - Deconvolution and polynomial division.
- `detrend` - Linear trend removal.

Fourier transforms.

<code>fft</code>	- Discrete Fourier transform.
<code>fft2</code>	- Two-dimensional discrete Fourier transform.
<code>fftn</code>	- N-dimensional discrete Fourier Transform.
<code>ifft</code>	- Inverse discrete Fourier transform.
<code>ifft2</code>	- Two-dimensional inverse discrete Fourier transform.
<code>ifftn</code>	- N-dimensional inverse discrete Fourier Transform.
<code>fftshift</code>	- Shift zero-frequency component to center of spectrum.
<code>ifftshift</code>	- Inverse FFTSHIFT.

1.4 Data types and structures

Data types (classes)

<code>double</code>	- Convert to double precision.
<code>logical</code>	- Convert numeric values to logical.
<code>cell</code>	- Create cell array.
<code>struct</code>	- Create or convert to structure array.
<code>single</code>	- Convert to single precision.
<code>uint8</code>	- Convert to unsigned 8-bit integer.
<code>uint16</code>	- Convert to unsigned 16-bit integer.
<code>uint32</code>	- Convert to unsigned 32-bit integer.
<code>uint64</code>	- Convert to unsigned 64-bit integer.
<code>int8</code>	- Convert to signed 8-bit integer.
<code>int16</code>	- Convert to signed 16-bit integer.
<code>int32</code>	- Convert to signed 32-bit integer.
<code>int64</code>	- Convert to signed 64-bit integer.
<code>inline</code>	- Construct <code>INLINE</code> object.
<code>function_handle</code>	- Function handle array.
<code>javaArray</code>	- Construct a Java Array object.
<code>javaMethod</code>	- Invoke a Java method.
<code>javaObject</code>	- Invoke a Java object constructor.
<code>cast</code>	- Cast a variable to a different data type or class.

Class determination functions.

<code>isnumeric</code>	- True for numeric arrays.
<code>isfloat</code>	- True for floating point arrays, both single and double.
<code>isinteger</code>	- True for arrays of integer data type.
<code>islogical</code>	- True for logical array.
<code>iscom</code>	- true for COM/ActiveX objects.
<code>isinterface</code>	- true for COM Interfaces.

Cell array functions.

<code>cell</code>	- Create cell array.
<code>celldisp</code>	- Display cell array contents.
<code>cellplot</code>	- Display graphical depiction of cell array.
<code>cell2mat</code>	- Convert the contents of a cell array into a single matrix.
<code>mat2cell</code>	- Break matrix up into a cell array of matrices.
<code>num2cell</code>	- Convert numeric array into cell array.
<code>deal</code>	- Deal inputs to outputs.
<code>cell2struct</code>	- Convert cell array into structure array.
<code>struct2cell</code>	- Convert structure array into cell array.
<code>iscell</code>	- True for cell array.

Array functions.

<code>arrayfun</code>	- Apply a function to each element of an array.
<code>cellfun</code>	- Apply a function to each cell of a cell array.
<code>structfun</code>	- Apply a function to each field of a scalar structure.

Structure functions.

<code>struct</code>	- Create or convert to structure array.
<code>fieldnames</code>	- Get structure field names.
<code>getfield</code>	- Get structure field contents.
<code>setfield</code>	- Set structure field contents.
<code>rmfield</code>	- Remove fields from a structure array.
<code>isfield</code>	- True if field is in structure array.
<code>isstruct</code>	- True for structures.
<code>orderfields</code>	- Order fields of a structure array.

Function handle functions.

<code>@</code>	- Create <code>function_handle</code> ; use "help <code>function_handle</code> ".
<code>func2str</code>	- Construct a string from a function handle.
<code>str2func</code>	- Construct a <code>function_handle</code> from a function name string.
<code>functions</code>	- List functions associated with a <code>function_handle</code> .

Byte manipulation functions.

<code>swapbytes</code>	- Swap byte ordering, changing endianness.
<code>typecast</code>	- Convert datatypes without changing underlying data.

Object oriented programming functions.

<code>class</code>	- Create object or return object class.
<code>struct</code>	- Convert object to structure array.
<code>methods</code>	- Display class method names.
<code>methodsview</code>	- View names and properties of class methods.
<code>isa</code>	- True if object is a given class.
<code>isjava</code>	- True for Java object arrays
<code>isobject</code>	- True for MATLAB objects.
<code>inferiorto</code>	- Inferior class relationship.
<code>superiorto</code>	- Superior class relationship.
<code>substruct</code>	- Create structure argument for <code>SUBSREF</code> or <code>SUBSASGN</code> .
<code>ismethod</code>	- True if method of an object.
<code>isprop</code>	- Returns true if the property exists
<code>loadobj</code>	- Called when loading an object from a <code>.MAT</code> file.
<code>saveobj</code>	- Called when saving an object to a <code>.MAT</code> file.

1.5 Elementary math functions

Trigonometric.

<code>sin</code>	- Sine.
<code>sind</code>	- Sine of argument in degrees.
<code>sinh</code>	- Hyperbolic sine.
<code>asin</code>	- Inverse sine.
<code>asind</code>	- Inverse sine, result in degrees.
<code>asinh</code>	- Inverse hyperbolic sine.
<code>cos</code>	- Cosine.
<code>cosd</code>	- Cosine of argument in degrees.
<code>cosh</code>	- Hyperbolic cosine.
<code>acos</code>	- Inverse cosine.
<code>acosd</code>	- Inverse cosine, result in degrees.
<code>acosh</code>	- Inverse hyperbolic cosine.
<code>tan</code>	- Tangent.
<code>tand</code>	- Tangent of argument in degrees.
<code>tanh</code>	- Hyperbolic tangent.
<code>atan</code>	- Inverse tangent.
<code>atand</code>	- Inverse tangent, result in degrees.
<code>atan2</code>	- Four quadrant inverse tangent.
<code>atanh</code>	- Inverse hyperbolic tangent.
<code>sec</code>	- Secant.
<code>secd</code>	- Secant of argument in degrees.
<code>sech</code>	- Hyperbolic secant.
<code>asec</code>	- Inverse secant.
<code>asecd</code>	- Inverse secant, result in degrees.
<code>asech</code>	- Inverse hyperbolic secant.
<code>csc</code>	- Cosecant.
<code>cscd</code>	- Cosecant of argument in degrees.
<code>csch</code>	- Hyperbolic cosecant.
<code>acsc</code>	- Inverse cosecant.
<code>acscd</code>	- Inverse cosecant, result in degrees.
<code>acsch</code>	- Inverse hyperbolic cosecant.
<code>cot</code>	- Cotangent.
<code>cotd</code>	- Cotangent of argument in degrees.
<code>coth</code>	- Hyperbolic cotangent.
<code>acot</code>	- Inverse cotangent.
<code>acotd</code>	- Inverse cotangent, result in degrees.
<code>acoth</code>	- Inverse hyperbolic cotangent.
<code>hypot</code>	- Square root of sum of squares.

Exponential.

<code>exp</code>	- Exponential.
<code>expm1</code>	- Compute $\exp(x)-1$ accurately.
<code>log</code>	- Natural logarithm.
<code>log1p</code>	- Compute $\log(1+x)$ accurately.
<code>log10</code>	- Common (base 10) logarithm.
<code>log2</code>	- Base 2 logarithm and dissect floating point number.
<code>pow2</code>	- Base 2 power and scale floating point number.
<code>realpow</code>	- Power that will error out on complex result.
<code>reallog</code>	- Natural logarithm of real number.
<code>realsqrt</code>	- Square root of number greater than or equal to zero.
<code>sqrt</code>	- Square root.
<code>nthroot</code>	- Real n-th root of real numbers.
<code>nextpow2</code>	- Next higher power of 2.

Complex.

<code>abs</code>	- Absolute value.
<code>angle</code>	- Phase angle.
<code>complex</code>	- Construct complex data from real and imaginary parts.
<code>conj</code>	- Complex conjugate.
<code>imag</code>	- Complex imaginary part.
<code>real</code>	- Complex real part.
<code>unwrap</code>	- Unwrap phase angle.
<code>isreal</code>	- True for real array.
<code>cplxpair</code>	- Sort numbers into complex conjugate pairs.

Rounding and remainder.

<code>fix</code>	- Round towards zero.
<code>floor</code>	- Round towards minus infinity.
<code>ceil</code>	- Round towards plus infinity.
<code>round</code>	- Round towards nearest integer.
<code>mod</code>	- Modulus (signed remainder after division).
<code>rem</code>	- Remainder after division.
<code>sign</code>	- Signum.

1.6 Elementary matrices and matrix manipulation

Elementary matrices.

<code>zeros</code>	- Zeros array.
<code>ones</code>	- Ones array.
<code>eye</code>	- Identity matrix.
<code> repmat</code>	- Replicate and tile array.
<code>rand</code>	- Uniformly distributed random numbers.
<code>randn</code>	- Normally distributed random numbers.
<code>linspace</code>	- Linearly spaced vector.
<code>logspace</code>	- Logarithmically spaced vector.
<code>freqspace</code>	- Frequency spacing for frequency response.
<code>meshgrid</code>	- X and Y arrays for 3-D plots.
<code>accumarray</code>	- Construct an array with accumulation.
<code>:</code>	- Regularly spaced vector and index into matrix.

Basic array information.

<code>size</code>	- Size of array.
<code>length</code>	- Length of vector.
<code>ndims</code>	- Number of dimensions.
<code> numel</code>	- Number of elements.
<code>disp</code>	- Display matrix or text.
<code>isempty</code>	- True for empty array.
<code>isequal</code>	- True if arrays are numerically equal.
<code>isequalwithequalnans</code>	- True if arrays are numerically equal.

Matrix manipulation.

<code>cat</code>	- Concatenate arrays.
<code>reshape</code>	- Change size.
<code>diag</code>	- Diagonal matrices and diagonals of matrix.
<code>blkdiag</code>	- Block diagonal concatenation.
<code>tril</code>	- Extract lower triangular part.
<code>triu</code>	- Extract upper triangular part.
<code>fliplr</code>	- Flip matrix in left/right direction.
<code>flipud</code>	- Flip matrix in up/down direction.
<code>flipdim</code>	- Flip matrix along specified dimension.
<code>rot90</code>	- Rotate matrix 90 degrees.
<code>:</code>	- Regularly spaced vector and index into matrix.
<code>find</code>	- Find indices of nonzero elements.
<code>end</code>	- Last index.
<code>sub2ind</code>	- Linear index from multiple subscripts.
<code>ind2sub</code>	- Multiple subscripts from linear index.
<code>bsxfun</code>	- Binary singleton expansion function.

Multi-dimensional array functions.

- ndgrid - Generate arrays for N-D functions and interpolation.
- permute - Permute array dimensions.
- ipermute - Inverse permute array dimensions.
- shiftdim - Shift dimensions.
- circshift - Shift array circularly.
- squeeze - Remove singleton dimensions.

Array utility functions.

- isscalar - True for scalar.
- isvector - True for vector.

Special variables and constants.

- ans - Most recent answer.
- eps - Floating point relative accuracy.
- realmax - Largest positive floating point number.
- realmin - Smallest positive floating point number.
- pi - 3.1415926535897....
- i - Imaginary unit.
- inf - Infinity.
- nan - Not-a-Number.
- isnan - True for Not-a-Number.
- isinf - True for infinite elements.
- isfinite - True for finite elements.
- j - Imaginary unit.
- why - Succinct answer.

Specialized matrices.

- compan - Companion matrix.
- gallery - Higham test matrices.
- hadamard - Hadamard matrix.
- hankel - Hankel matrix.
- hilb - Hilbert matrix.
- invhilb - Inverse Hilbert matrix.
- magic - Magic square.
- pascal - Pascal matrix.
- rosser - Classic symmetric eigenvalue test problem.
- toeplitz - Toeplitz matrix.
- vander - Vandermonde matrix.
- wilkinson - Wilkinson's eigenvalue test matrix.

1.7 Function functions and ODE solvers

Optimization and root finding.

- `fminbnd` - Scalar bounded nonlinear function minimization.
- `fminsearch` - Multidimensional unconstrained nonlinear minimization, by Nelder-Mead direct search method.
- `fzero` - Scalar nonlinear zero finding.

Optimization Option handling

- `optimset` - Create or alter optimization OPTIONS structure.
- `optimget` - Get optimization parameters from OPTIONS structure.

Numerical integration (quadrature).

- `quad` - Numerically evaluate integral, low order method.
- `quadl` - Numerically evaluate integral, higher order method.
- `quadv` - Vectorized QUAD.
- `dblquad` - Numerically evaluate double integral.
- `triplequad` - Numerically evaluate triple integral.

Plotting.

- `ezplot` - Easy to use function plotter.
- `ezplot3` - Easy to use 3-D parametric curve plotter.
- `ezpolar` - Easy to use polar coordinate plotter.
- `ezcontour` - Easy to use contour plotter.
- `ezcontourf` - Easy to use filled contour plotter.
- `ezmesh` - Easy to use 3-D mesh plotter.
- `ezmeshc` - Easy to use combination mesh/contour plotter.
- `ezsurf` - Easy to use 3-D colored surface plotter.
- `ezsurfz` - Easy to use combination surf/contour plotter.
- `fplot` - Plot function.

Inline function object.

- `inline` - Construct INLINE function object.
- `argnames` - Argument names.
- `formula` - Function formula.
- `char` - Convert INLINE object to character array.

Differential equation solvers.

Initial value problem solvers for ODEs. (If unsure about stiffness, try ODE45 first, then ODE15S.)

- `ode45` - Solve non-stiff differential equations, medium order method.
- `ode23` - Solve non-stiff differential equations, low order method.
- `ode113` - Solve non-stiff differential equations, variable order method.
- `ode23t` - Solve moderately stiff ODEs and DAEs Index 1, trapezoidal rule.
- `ode15s` - Solve stiff ODEs and DAEs Index 1, variable order method.
- `ode23s` - Solve stiff differential equations, low order method.
- `ode23tb` - Solve stiff differential equations, low order method.

Initial value problem solver for fully implicit ODEs/DAEs $F(t, y, y')=0$.

- decic - Compute consistent initial conditions.
- ode15i - Solve implicit ODEs or DAEs Index 1.

Initial value problem solver for delay differential equations (DDEs).

- dde23 - Solve delay differential equations (DDEs) with constant delays.
- dodes - Solve delay differential equations (DDEs) with variable delays.

Boundary value problem solver for ODEs.

- bvp4c - Solve boundary value problems for ODEs by collocation.

1D Partial differential equation solver.

- pdepe - Solve initial-boundary value problems for parabolic-elliptic PDEs.

Option handling.

- odeset - Create/alter ODE OPTIONS structure.
- odeget - Get ODE OPTIONS parameters.
- dodeset - Create/alter DDE OPTIONS structure.
- dodeget - Get DDE OPTIONS parameters.
- bvpset - Create/alter BVP OPTIONS structure.
- bvpget - Get BVP OPTIONS parameters.

Input and Output functions.

- deval - Evaluates the solution of a differential equation problem.
- odextend - Extends the solutions of a differential equation problem.
- odeplot - Time series ODE output function.
- odephas2 - 2-D phase plane ODE output function.
- odephas3 - 3-D phase plane ODE output function.
- odeprint - Command window printing ODE output function.
- bvpinit - Forms the initial guess for BVP4C.
- bvpextend - Forms a guess structure for extending BVP solution.
- pdeval - Evaluates by interpolation the solution computed by PDEPE.
- odefile - MATLAB v5 ODE file syntax (obsolete).

1.8 General purpose commands

General information.

<code>syntax</code>	- Help on MATLAB command syntax.
<code>demo</code>	- Run demonstrations.
<code>ver</code>	- MATLAB, Simulink and toolbox version information.
<code>version</code>	- MATLAB version information.
<code>verLessThan</code>	- Compare version of toolbox to specified version string.

Managing the workspace.

<code>who</code>	- List current variables.
<code>whos</code>	- List current variables, long form.
<code>clear</code>	- Clear variables and functions from memory.
<code>pack</code>	- Consolidate workspace memory.
<code>load</code>	- Load workspace variables from disk.
<code>save</code>	- Save workspace variables to disk.
<code>saveas</code>	- Save Figure or model to desired output format.
<code>memory</code>	- Help for memory limitations.
<code>recycle</code>	- Set option to move deleted files to recycle folder.
<code>quit</code>	- Quit MATLAB session.
<code>exit</code>	- Exit from MATLAB.

Managing commands and functions.

<code>what</code>	- List MATLAB-specific files in directory.
<code>type</code>	- List M-file.
<code>open</code>	- Open files by extension.
<code>which</code>	- Locate functions and files.
<code>pcode</code>	- Create pre-parsed pseudo-code file (P-file).
<code>mex</code>	- Compile MEX-function.
<code>inmem</code>	- List functions in memory.
<code>namelengthmax</code>	- Maximum length of MATLAB function or variable name.

Managing the search path.

<code>path</code>	- Get/set search path.
<code>addpath</code>	- Add directory to search path.
<code>rmpath</code>	- Remove directory from search path.
<code>rehash</code>	- Refresh function and file system caches.
<code>import</code>	- Import Java packages into the current scope.
<code>finfo</code>	- Identify file type against standard file handlers on path.
<code>genpath</code>	- Generate recursive toolbox path.
<code>savepath</code>	- Save the current MATLAB path in the pathdef.m file.

Managing the java search path.

<code>javaaddpath</code>	- Add directories to the dynamic java path.
<code>javaclasspath</code>	- Get and set java path.
<code>javarmppath</code>	- Remove directory from dynamic java path.

Controlling the command window.

echo	- Echo commands in M-files.
more	- Control paged output in command window.
diary	- Save text of MATLAB session.
format	- Set output format.
beep	- Produce beep sound.
desktop	- Start and query the MATLAB Desktop.
preferences	- Bring up MATLAB user settable preferences dialog.

Operating system commands.

cd	- Change current working directory.
copyfile	- Copy file or directory.
movefile	- Move file or directory.
delete	- Delete file or graphics object.
pwd	- Show (print) current working directory.
dir	- List directory.
ls	- List directory.
fileattrib	- Set or get attributes of files and directories.
isdir	- True if argument is a directory.
mkdir	- Make new directory.
rmdir	- Remove directory.
getenv	- Get environment variable.
!	- Execute operating system command (see PUNCT).
dos	- Execute DOS command and return result.
unix	- Execute UNIX command and return result.
system	- Execute system command and return result.
perl	- Execute Perl command and return the result.
computer	- Computer type.
isunix	- True for the UNIX version of MATLAB.
ispc	- True for the PC (Windows) version of MATLAB.

Debugging.

debug	- List debugging commands.
-------	----------------------------

Tools to locate dependent functions of an M-file.

depfun	- Locate dependent functions of an M-file or P-file.
depcdir	- Locate dependent directories of an M-file or P-file.

Loading and calling shared libraries.

<code>calllib</code>	- Call a function in an external library.
<code>libpointer</code>	- Creates a pointer object for use with external libraries.
<code>libstruct</code>	- Creates a structure pointer for use with external libraries.
<code>libisloaded</code>	- True if the specified shared library is loaded.
<code>loadlibrary</code>	- Load a shared library into MATLAB.
<code>libfunctions</code>	- Return information on functions in an external library.
<code>libfunctionsview</code>	- View the functions in an external library.
<code>unloadlibrary</code>	- Unload a shared library loaded with <code>LOADLIBRARY</code> .
<code>java</code>	- Using Java from within MATLAB.
<code>usejava</code>	- True if the specified Java feature is supported in MATLAB.

See also `lang`, `datatypes`, `iofun`, `graphics`, `ops`, `strfun`, `timefun`,
`matfun`, `demos`, `graphics`, `datafun`, `uitools`, `doc`, `punct`, `arith`.

1.9 Two dimensional graphs

Elementary X-Y graphs.

<code>plot</code>	- Linear plot.
<code>loglog</code>	- Log-log scale plot.
<code>semilogx</code>	- Semi-log scale plot.
<code>semilogy</code>	- Semi-log scale plot.
<code>polar</code>	- Polar coordinate plot.
<code>plotyy</code>	- Graphs with y tick labels on the left and right.

Axis control.

<code>axis</code>	- Control axis scaling and appearance.
<code>zoom</code>	- Zoom in and out on a 2-D plot.
<code>grid</code>	- Grid lines.
<code>box</code>	- Axis box.
<code>rbbox</code>	- Rubberband box.
<code>hold</code>	- Hold current graph.
<code>axes</code>	- Create axes in arbitrary positions.
<code>subplot</code>	- Create axes in tiled positions.

Graph annotation.

<code>plotedit</code>	- Tools for editing and annotating plots.
<code>title</code>	- Graph title.
<code>xlabel</code>	- X-axis label.
<code>ylabel</code>	- Y-axis label.
<code>texlabel</code>	- Produces the TeX format from a character string.
<code>text</code>	- Text annotation.
<code>gtext</code>	- Place text with mouse.

Hardcopy and printing.

- `print` - Print graph or Simulink system; or save graph to M-file.
- `printopt` - Printer defaults.
- `orient` - Set paper orientation.

See also `graph3d`, `specgraph`.

1.10 Three dimensional graphs

Elementary 3-D plots.

- `plot3` - Plot lines and points in 3-D space.
- `mesh` - 3-D mesh surface.
- `surf` - 3-D colored surface.
- `fill3` - Filled 3-D polygons.

Color control.

- `colormap` - Color look-up table.
- `caxis` - Pseudocolor axis scaling.
- `shading` - Color shading mode.
- `hidden` - Mesh hidden line removal mode.
- `brighten` - Brighten or darken color map.
- `colordef` - Set color defaults.
- `graymon` - Set graphics defaults for gray-scale monitors.

Lighting.

- `surfl` - 3-D shaded surface with lighting.
- `lighting` - Lighting mode.
- `material` - Material reflectance mode.
- `specular` - Specular reflectance.
- `diffuse` - Diffuse reflectance.
- `surfnorm` - Surface normals.

Color maps.

hsv	- Hue-saturation-value color map.
hot	- Black-red-yellow-white color map.
gray	- Linear gray-scale color map.
bone	- Gray-scale with tinge of blue color map.
copper	- Linear copper-tone color map.
pink	- Pastel shades of pink color map.
white	- All white color map.
flag	- Alternating red, white, blue, and black color map.
lines	- Color map with the line colors.
colorcube	- Enhanced color-cube color map.
vga	- Windows colormap for 16 colors.
jet	- Variant of HSV.
prism	- Prism color map.
cool	- Shades of cyan and magenta color map.
autumn	- Shades of red and yellow color map.
spring	- Shades of magenta and yellow color map.
winter	- Shades of blue and green color map.
summer	- Shades of green and yellow color map.

Transparency.

alpha	- Transparency (alpha) mode.
alphamap	- Transparency (alpha) look-up table.
alim	- Transparency (alpha) scaling

Axis control.

axis	- Control axis scaling and appearance.
zoom	- Zoom in and out on a 2-D plot.
grid	- Grid lines.
box	- Axis box.
hold	- Hold current graph.
axes	- Create axes in arbitrary positions.
subplot	- Create axes in tiled positions.
daspect	- Data aspect ratio.
pbaspect	- Plot box aspect ratio.
xlim	- X limits.
ylim	- Y limits.
zlim	- Z limits.

Viewpoint control.

view	- 3-D graph viewpoint specification.
viewmtx	- View transformation matrix.
rotate3d	- Interactively rotate view of 3-D plot.

Camera control.

- `campos` - Camera position.
- `camtarget` - Camera target.
- `camva` - Camera view angle.
- `camup` - Camera up vector.
- `camproj` - Camera projection.

High level camera control.

- `camorbit` - Orbit camera.
- `campan` - Pan camera.
- `camdolly` - Dolly camera.
- `camzoom` - Zoom camera.
- `camroll` - Roll camera.
- `camlookat` - Move camera and target to view specified objects.
- `cameratoolbar` - Interactively manipulate camera.

High level light control.

- `camlight` - Creates or sets position of a light.
- `lightangle` - Spherical position of a light.

Graph annotation.

- `title` - Graph title.
- `xlabel` - X-axis label.
- `ylabel` - Y-axis label.
- `zlabel` - Z-axis label.
- `text` - Text annotation.
- `gtext` - Mouse placement of text.
- `plottedit` - Experimental graph editing and annotation tools.

Hardcopy and printing.

- `print` - Print graph or Simulink system; or save graph to M-file.
- `printopt` - Printer defaults.
- `orient` - Set paper orientation.
- `vrml` - Save graphics to VRML 2.0 file.

See also `graph2d`, `specgraph`.

1.11 Handle Graphics

Figure window creation and control.

<code>figure</code>	- Create figure window.
<code>gcf</code>	- Get handle to current figure.
<code>clf</code>	- Clear current figure.
<code>shg</code>	- Show graph window.
<code>close</code>	- Close figure.
<code>refresh</code>	- Refresh figure.
<code>refreshdata</code>	- Refresh data in plot
<code>openfig</code>	- Open new copy or raise existing copy of saved figure.

Axis creation and control.

<code>subplot</code>	- Create axes in tiled positions.
<code>axes</code>	- Create axes in arbitrary positions.
<code>gca</code>	- Get handle to current axis.
<code>cla</code>	- Clear current axis.
<code>axis</code>	- Control axis scaling and appearance.
<code>box</code>	- Axis box.
<code>caxis</code>	- Control pseudocolor axis scaling.
<code>hold</code>	- Hold current graph
<code>ishold</code>	- Return hold state.

Handle Graphics objects.

<code>figure</code>	- Create figure window.
<code>axes</code>	- Create axes in arbitrary positions.
<code>line</code>	- Create line.
<code>text</code>	- Create text.
<code>patch</code>	- Create patch.
<code>rectangle</code>	- Create rectangle, rounded-rectangle, or ellipse.
<code>surface</code>	- Create surface.
<code>image</code>	- Create image.
<code>light</code>	- Create light.
<code>uicontrol</code>	- Create user interface control.
<code>uimenu</code>	- Create user interface menu.
<code>uicontextmenu</code>	- Create user interface context menu.

Handle Graphics operations.

<code>set</code>	- Set object properties.
<code>get</code>	- Get object properties.
<code>reset</code>	- Reset graphics object properties to their defaults.
<code>delete</code>	- Delete object.
<code>gco</code>	- Get handle to current object.
<code>gcbo</code>	- Get handle to current callback object.
<code>gcbf</code>	- Get handle to current callback figure.
<code>drawnow</code>	- Flush pending graphics events.
<code>findobj</code>	- Find objects with specified property values.
<code>copyobj</code>	- Make copy of graphics object and its children.
<code>isappdata</code>	- True if application-defined data exists.
<code>getappdata</code>	- Get value of application-defined data.
<code>setappdata</code>	- Set application-defined data.
<code>rmappdata</code>	- Remove application-defined data.

Hardcopy and printing.

<code>print</code>	- Print figure or model. Save to disk as image or M-file.
<code>printopt</code>	- Printer defaults.
<code>orient</code>	- Set paper orientation for printing.
<code>pagesetupdlg</code>	- Page setup dialog
<code>ppreview</code>	- Display preview of figure to be printed
<code>printdlg</code>	- Print dialog box.
<code>printpreview</code>	- Display preview of figure to be printed
<code>figureheaderdlg</code>	- Show figure header dialog
<code>copyoptionsfcn</code>	- brings up the preferences dialog with Copy Options selected.
<code>exportsetupdlg</code>	- Figure style editor

Utilities.

<code>closereq</code>	- Figure close request function.
<code>newplot</code>	- M-file preamble for NextPlot property.
<code>ishandle</code>	- True for graphics handles.
<code>findall</code>	- find all objects.

ActiveX Client Functions (PC Only).

<code>actxcontrol</code>	- Create an ActiveX control.
<code>actxserver</code>	- Create an ActiveX server.

See also `graph2d`, `graph3d`, `specgraph`, `winfun`.

1.12 Help commands

Accessing on-line HTML help.

- helpbrowser - Bring up the help browser.
- helpdesk - Comprehensive hypertext documentation and troubleshooting.
- helpview - Display HTML file in help viewer.
- doc - Display HTML documentation in the Help browser.
- docsearch - Search HTML documentation in the Help browser.

Accessing M-file help.

- help - M-file help, displayed at the command line.
- helpwin - M-file help, displayed in the help browser.
- lookfor - Search all M-files for keyword.

Accessing information about MathWorks products and technical support.

- info - Information about MATLAB and The MathWorks.
- support - Open MathWorks Technical Support Web Page.
- whatsnew - Access Release Notes.

Display HTML files inside internal or system Web browser.

- web - Open Web browser on site or files.

1.13 Image and scientific data input/output

Image file import/export.

- informats - List details about supported file formats.
- iminfo - Return information about graphics file.
- imread - Read image from graphics file.
- imwrite - Write image to graphics file.
- im2java - Convert image to Java image.
- multibandread - Read band-interleaved data from a binary file.
- multibandwrite - Write multiband data to a file.
- exifread - Read the EXIF information from JPEG and TIFF images.

CDF file handling.

- cdfread - Read data from a CDF file.
- cdfinfo - Get information from a CDF file.
- cdfwrite - Write data to a CDF file.
- cdfepoch - Construct cdfepoch object.

FITS file handling.

- fitsinfo - Get information from a FITS file.
- fitsread - Read data from a FITS file.

HDF version 4 file handling.

- hdfinfo - Get information about an HDF4 file.
- hdfread - Extract from an HDF4 file.
- hdftool - Browse and import from HDF4 or HDF-EOS files.

HDF version 5 file handling.

- hdf5info - Get information about an HDF5 file.
- hdf5read - Read datasets and attributes from an HDF5 file.
- hdf5write - Write datasets and attributes to an HDF5 file.

HDF version 5 data objects.

- hdf5.h5array - Construct HDF5 array datatype object.
- hdf5.h5compound - Construct HDF5 compound datatype object.
- hdf5.h5enum - Construct HDF5 enumeration datatype object.
- hdf5.h5string - Construct HDF5 string datatype object.
- hdf5.h5vlen - Construct HDF5 variable length array datatype object.

HDF version 4 library interface.

- hdf - MEX-file interface to the HDF library.
- hdfan - MATLAB Gateway to HDF multifile annotation interface.
- hdfdf24 - MATLAB Gateway to HDF raster image interface.
- hdfdf8 - MATLAB Gateway to HDF 8-bit raster image interface.
- hdfh - MATLAB Gateway to HDF H interface.
- hdfhd - MATLAB Gateway to HDF HD interface.
- hdfhe - MATLAB Gateway to HDF HE interface.
- hdfhx - MATLAB Gateway to HDF HX interface.
- hdfml - MATLAB-HDF gateway utilities.
- hdfsd - MATLAB Gateway to HDF multifile scientific dataset interface.
- hdfv - MATLAB Gateway to HDF V (Vgroup) interface.
- hdfvf - MATLAB Gateway to HDF VF (Vdata) interface.
- hdfvh - MATLAB Gateway to HDF VH (Vdata) interface.
- hdfvs - MATLAB Gateway to HDF VS (Vdata) interface.

HDF-EOS library interface help.

- hdfgd - MATLAB Gateway to HDF-EOS grid interface.
- hdfpt - MATLAB Gateway to HDF-EOS point interface.
- hdfsw - MATLAB Gateway to HDF-EOS swath interface.

HDF version 5 library interface.

- H5 - MATLAB Gateway to HDF5 Library Interface.
- H5A - MATLAB Gateway to HDF5 Attribute Interface.
- H5D - MATLAB Gateway to HDF5 Dataset Interface.
- H5E - MATLAB Gateway to HDF5 Error Interface.
- H5F - MATLAB Gateway to HDF5 File Interface.
- H5G - MATLAB Gateway to HDF5 Group Interface.
- H5I - MATLAB Gateway to HDF5 Identifier Interface.
- H5ML - MATLAB-HDF5 Gateway utilities.
- H5P - MATLAB Gateway to HDF5 Property Interface.
- H5R - MATLAB Gateway to HDF5 Reference Interface.
- H5S - MATLAB Gateway to HDF5 Dataspace Interface.
- H5T - MATLAB Gateway to HDF5 Datatype Interface.
- H5Z - MATLAB Gateway to HDF5 Compression Interface.

See also `audiovideo`, `iofun`.

1.14 File input and output

File import/export functions.

<code>dlmread</code>	- Read ASCII delimited file.
<code>dlmwrite</code>	- Write ASCII delimited file.
<code>importdata</code>	- Load data from a file into MATLAB.
<code>daqread</code>	- Read Data Acquisition Toolbox (.daq) data file.
<code>matfinfo</code>	- Text description of MAT-file contents.

Spreadsheet support.

<code>xlsread</code>	- Get data and text from a spreadsheet in an Excel workbook.
<code>xlswrite</code>	- Stores numeric array or cell array in Excel workbook.
<code>xlsfinfo</code>	- Determine if file contains Microsoft Excel spreadsheet.
<code>wk1read</code>	- Read spreadsheet (WK1) file.
<code>wk1write</code>	- Write spreadsheet (WK1) file.
<code>wk1finfo</code>	- Determine if file contains Lotus WK1 worksheet.
<code>str2rng</code>	- Convert spreadsheet range string to numeric array.
<code>wk1wrec</code>	- Write a WK1 record header.

Internet resource.

<code>urlread</code>	- Returns the contents of a URL as a string.
<code>urlwrite</code>	- Save the contents of a URL to a file.
<code>ftp</code>	- Create an FTP object.
<code>sendmail</code>	- Send e-mail.

Zip file access.

<code>zip</code>	- Compress files into zip file.
<code>unzip</code>	- Extract contents of zip file.

Tar file access.

<code>tar</code>	- Compress files into tar file.
<code>untar</code>	- Extract contents of tar file.

Gzip file access.

<code>gzip</code>	- Compress files into GNU zip files.
<code>gunzip</code>	- Uncompress GNU zip files.

Formatted file I/O.

<code>fgetl</code>	- Read line from file, discard newline character.
<code>fgets</code>	- Read line from file, keep newline character.
<code>fprintf</code>	- Write formatted data to file.
<code>fscanf</code>	- Read formatted data from file.
<code>textscan</code>	- Read formatted data from text file.
<code>textread</code>	- Read formatted data from text file.

File opening and closing.

<code>fopen</code>	- Open file.
<code>fclose</code>	- Close file.

Binary file I/O.

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

File positioning.

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

Memory-mapped file support.

- `memmapfile` - Construct memory-mapped file object.

File name handling.

- `fileparts` - Filename parts.
- `filesep` - Directory separator for this platform.
- `fullfile` - Build full filename from parts.
- `matlabroot` - Root directory of MATLAB installation.
- `mexext` - MEX filename extension for this platform.
- `partialpath` - Partial pathnames.
- `pathsep` - Path separator for this platform.
- `prefdir` - Preference directory name.
- `tempdir` - Get temporary directory.
- `tempname` - Get temporary file.

XML file handling.

- `xmlread` - Parse an XML document and return a Document Object Model node.
- `xmlwrite` - Serialize an XML Document Object Model node.
- `xslt` - Transform an XML document using an XSLT engine.

Serial port support.

- `serial` - Construct serial port object.
- `instrfindall` - Find all serial port objects with specified property values.
- `freeserial` - Release MATLAB's hold on serial port.
- `instrfind` - Find serial port objects with specified property values.

Timer support.

- `timer` - Construct timer object.
- `timerfindall` - Find all timer objects with specified property values.
- `timerfind` - Find visible timer objects with specified property values.

Command window I/O.

- `clc` - Clear command window.
- `home` - Send the cursor home.

SOAP support.

- `createClassFromWsd1` - Create a MATLAB object based on a WSDL-file.
- `callSoapService` - Send a SOAP message off to an endpoint.
- `createSoapMessage` - Create the SOAP message, ready to send to the server.
- `parseSoapResponse` - Convert the response from a SOAP server into MATLAB types.

See also `general`, `lang`, `audiovideo`, `imagesci`, `graphics`, `uitools`.

1.15 Programming language constructs

Control flow.

- `if` - Conditionally execute statements.
- `else` - Execute statement if previous IF condition failed.
- `elseif` - Execute if previous IF failed and condition is true.
- `end` - Terminate scope of control statements.
- `for` - Repeat statements a specific number of times.
- `while` - Repeat statements an indefinite number of times.
- `break` - Terminate execution of WHILE or FOR loop.
- `continue` - Pass control to the next iteration of a loop.
- `switch` - Switch among several cases based on expression.
- `case` - SWITCH statement case.
- `otherwise` - Default SWITCH statement case.
- `try` - Begin TRY block.
- `catch` - Begin CATCH block.
- `return` - Return to invoking function.
- `error` - Display message and abort function.
- `assert` - Generate an error when a condition is violated.
- `rethrow` - Reissue error.

Evaluation and execution.

- `eval` - Execute string with MATLAB expression.
- `evalc` - Evaluate MATLAB expression with capture.
- `feval` - Execute the specified function.
- `evalin` - Evaluate expression in workspace.
- `builtin` - Execute built-in function from overloaded method.
- `assignin` - Assign variable in workspace.
- `run` - Run script.

Scripts, functions, and variables.

<code>script</code>	- About MATLAB scripts and M-files.
<code>function</code>	- Add new function.
<code>global</code>	- Define global variable.
<code>persistent</code>	- Define persistent variable.
<code>mfilename</code>	- Name of currently executing M-file.
<code>lists</code>	- Comma separated lists.
<code>exist</code>	- Check if variables or functions are defined.
<code>mlock</code>	- Prevent M-file from being cleared.
<code>munlock</code>	- Allow M-file to be cleared.
<code>mislocked</code>	- True if M-file cannot be cleared.
<code>precedence</code>	- Operator Precedence in MATLAB.
<code>isvarname</code>	- True for valid variable name.
<code>iskeyword</code>	- Check if input is a keyword.
<code>javachk</code>	- Validate level of Java support.
<code>genvarname</code>	- Construct a valid MATLAB variable name from a string.

Argument handling.

<code>nargchk</code>	- Validate number of input arguments.
<code>nargoutchk</code>	- Validate number of output arguments.
<code>nargin</code>	- Number of function input arguments.
<code>nargout</code>	- Number of function output arguments.
<code>varargin</code>	- Variable length input argument list.
<code>varargout</code>	- Variable length output argument list.
<code>inputname</code>	- Input argument name.

Message display.

<code>warning</code>	- Display warning message.
<code>lasterr</code>	- Last error message.
<code>lasterror</code>	- Last error message and related information.
<code>lastwarn</code>	- Last warning message.
<code>disp</code>	- Display array.
<code>display</code>	- Display array.
<code>intwarning</code>	- Controls the state of the 4 integer warnings.

Interactive input.

<code>input</code>	- Prompt for user input.
<code>keyboard</code>	- Invoke keyboard from M-file.

See also `general`, `iofun`, `ops`, `datatypes`, `matfun`, `funfun`, `elfun`, `polyfun`.

1.16 Matrix functions - numerical linear algebra

Matrix analysis.

<code>norm</code>	- Matrix or vector norm.
<code>normest</code>	- Estimate the matrix 2-norm.
<code>rank</code>	- Matrix rank.
<code>det</code>	- Determinant.
<code>trace</code>	- Sum of diagonal elements.
<code>null</code>	- Null space.
<code>orth</code>	- Orthogonalization.
<code>rref</code>	- Reduced row echelon form.
<code>subspace</code>	- Angle between two subspaces.

Linear equations.

<code>\</code> and <code>/</code>	- Linear equation solution; use "help slash".
<code>linsolve</code>	- Linear equation solution with extra control.
<code>inv</code>	- Matrix inverse.
<code>rcond</code>	- LAPACK reciprocal condition estimator
<code>cond</code>	- Condition number with respect to inversion.
<code>condest</code>	- 1-norm condition number estimate.
<code>normest1</code>	- 1-norm estimate.
<code>chol</code>	- Cholesky factorization.
<code>cholinc</code>	- Incomplete Cholesky factorization.
<code>ldl</code>	- Block LDL' factorization.
<code>lu</code>	- LU factorization.
<code>luinc</code>	- Incomplete LU factorization.
<code>qr</code>	- Orthogonal-triangular decomposition.
<code>lsqnonneg</code>	- Linear least squares with nonnegativity constraints.
<code>pinv</code>	- Pseudoinverse.
<code>lscov</code>	- Least squares with known covariance.

Eigenvalues and singular values.

<code>eig</code>	- Eigenvalues and eigenvectors.
<code>svd</code>	- Singular value decomposition.
<code>gsvd</code>	- Generalized singular value decomposition.
<code>eigs</code>	- A few eigenvalues.
<code>svds</code>	- A few singular values.
<code>poly</code>	- Characteristic polynomial.
<code>polyeig</code>	- Polynomial eigenvalue problem.
<code>condeig</code>	- Condition number with respect to eigenvalues.
<code>hess</code>	- Hessenberg form.
<code>schur</code>	- Schur decomposition.
<code>qz</code>	- QZ factorization for generalized eigenvalues.
<code>ordschur</code>	- Reordering of eigenvalues in Schur decomposition.
<code>ordqz</code>	- Reordering of eigenvalues in QZ factorization.
<code>ordeig</code>	- Eigenvalues of quasitriangular matrices.

Matrix functions.

`expm` - Matrix exponential.
`logm` - Matrix logarithm.
`sqrtm` - Matrix square root.
`funm` - Evaluate general matrix function.

Factorization utilities

`qrdelete` - Delete a column or row from QR factorization.
`qrinsert` - Insert a column or row into QR factorization.
`rsf2csf` - Real block diagonal form to complex diagonal form.
`cdf2rdf` - Complex diagonal form to real block diagonal form.
`balance` - Diagonal scaling to improve eigenvalue accuracy.
`planerot` - Givens plane rotation.
`cholupdate` - rank 1 update to Cholesky factorization.
`qrupdate` - rank 1 update to QR factorization.

1.17 Operators and special characters

Arithmetic operators.

`plus` - Plus +
`uplus` - Unary plus +
`minus` - Minus -
`uminus` - Unary minus -
`mtimes` - Matrix multiply *
`times` - Array multiply .*
`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 ./
`kron` - Kronecker tensor product kron

Relational operators.

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

Logical operators.

relop	- Short-circuit logical AND	&&
relop	- Short-circuit logical OR	
and	- Element-wise logical AND	&
or	- Element-wise logical OR	
not	- Logical NOT	~
xor	- Logical EXCLUSIVE OR	
any	- True if any element of vector is nonzero	
all	- True if all elements of vector are nonzero	

Special characters.

colon	- Colon	:
paren	- Parentheses and subscripting	()
paren	- Brackets	[]
paren	- Braces and subscripting	{ }
punct	- Function handle creation	@
punct	- Decimal point	.
punct	- Structure field access	.
punct	- Parent directory	..
punct	- Continuation	...
punct	- Separator	,
punct	- Semicolon	;
punct	- Comment	%
punct	- Invoke operating system command	!
punct	- Assignment	=
punct	- Quote	'
transpose	- Transpose	.'
ctranspose	- Complex conjugate transpose	'
horzcat	- Horizontal concatenation	[,]
vertcat	- Vertical concatenation	[:]
subasgn	- Subscripted assignment	(), { }, .
subsref	- Subscripted reference	(), { }, .
subsindex	- Subscript index	

Bitwise operators.

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

Set operators.

- `union` - Set union.
- `unique` - Set unique.
- `intersect` - Set intersection.
- `setdiff` - Set difference.
- `setxor` - Set exclusive-or.
- `ismember` - True for set member.

See also `arith`, `relop`, `slash`, `function_handle`.

1.18 Interpolation and polynomials

Data interpolation.

- `pchip` - Piecewise cubic Hermite interpolating polynomial.
- `interp1` - 1-D interpolation (table lookup).
- `interp1q` - Quick 1-D linear interpolation.
- `interpft` - 1-D interpolation using FFT method.
- `interp2` - 2-D interpolation (table lookup).
- `interp3` - 3-D interpolation (table lookup).
- `interpn` - N-D interpolation (table lookup).
- `griddata` - Data gridding and surface fitting.
- `griddata3` - Data gridding and hyper-surface fitting for 3-dimensional data.
- `griddatan` - Data gridding and hyper-surface fitting (dimension ≥ 2).

Spline interpolation.

- `spline` - Cubic spline interpolation.
- `ppval` - Evaluate piecewise polynomial.

Geometric analysis.

- `delaunay` - Delaunay triangulation.
- `delaunay3` - 3-D Delaunay tessellation.
- `delaunayn` - N-D Delaunay tessellation.
- `dsearch` - Search Delaunay triangulation for nearest point.
- `dsearchn` - Search N-D Delaunay tessellation for nearest point.
- `tsearch` - Closest triangle search.
- `tsearchn` - N-D closest triangle search.
- `convhull` - Convex hull.
- `convhulln` - N-D convex hull.
- `voronoi` - Voronoi diagram.
- `voronoin` - N-D Voronoi diagram.
- `inpolygon` - True for points inside polygonal region.
- `rectint` - Rectangle intersection area.
- `polyarea` - Area of polygon.

Polynomials.

roots	- Find polynomial roots.
poly	- Convert roots to polynomial.
polyval	- Evaluate polynomial.
polyvalm	- Evaluate polynomial with matrix argument.
residue	- Partial-fraction expansion (residues).
polyfit	- Fit polynomial to data.
polyder	- Differentiate polynomial.
polyint	- Integrate polynomial analytically.
conv	- Multiply polynomials.
deconv	- Divide polynomials.

1.19 Sparse matrices

Elementary sparse matrices.

speye	- Sparse identity matrix.
sprand	- Sparse uniformly distributed random matrix.
sprandn	- Sparse normally distributed random matrix.
sprandsym	- Sparse random symmetric matrix.
spdiags	- Sparse matrix formed from diagonals.

Full to sparse conversion.

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

Working with sparse matrices.

nnz	- Number of nonzero matrix elements.
nonzeros	- Nonzero matrix elements.
nzmax	- Amount of storage allocated for nonzero matrix elements.
spones	- Replace nonzero sparse matrix elements with ones.
spalloc	- Allocate space for sparse matrix.
issparse	- True for sparse matrix.
spfun	- Apply function to nonzero matrix elements.
spy	- Visualize sparsity pattern.

Reordering algorithms.

amd	- Approximate minimum degree permutation.
colamd	- Column approximate minimum degree permutation.
symamd	- Symmetric approximate minimum degree permutation.
colmmd	- Column minimum degree permutation.
symmmd	- Symmetric minimum degree permutation.
symrcm	- Symmetric reverse Cuthill-McKee permutation.
colperm	- Column permutation.
randperm	- Random permutation.
dmperm	- Dulmage-Mendelsohn permutation.

Linear algebra.

- `eigs` - A few eigenvalues, using ARPACK.
- `svds` - A few singular values, using `eigs`.
- `ilu` - Incomplete LU factorization.
- `luinc` - Incomplete LU factorization.
- `cholinc` - Incomplete Cholesky factorization.
- `normest` - Estimate the matrix 2-norm.
- `condest` - 1-norm condition number estimate.
- `sprank` - Structural rank.

Linear Equations (iterative methods).

- `pcg` - Preconditioned Conjugate Gradients Method.
- `bicg` - BiConjugate Gradients Method.
- `bicgstab` - BiConjugate Gradients Stabilized Method.
- `cgs` - Conjugate Gradients Squared Method.
- `gmres` - Generalized Minimum Residual Method.
- `lsqr` - LSQR Method.
- `minres` - Minimum Residual Method.
- `qmr` - Quasi-Minimal Residual Method.
- `symmlq` - Symmetric LQ Method.

Operations on graphs (trees).

- `treelayout` - Lay out tree or forest.
- `treepplot` - Plot picture of tree.
- `etree` - Elimination tree.
- `etreeplot` - Plot elimination tree.
- `gplot` - Plot graph, as in "graph theory".

Miscellaneous.

- `sympfact` - Symbolic factorization analysis.
- `spparms` - Set parameters for sparse matrix routines.
- `spsaugment` - Form least squares augmented system.

1.20 Specialized math functions

Elementary sparse matrices.

- speye - Sparse identity matrix.
- sprand - Sparse uniformly distributed random matrix.
- sprandn - Sparse normally distributed random matrix.
- sprandsym - Sparse random symmetric matrix.
- spdiags - Sparse matrix formed from diagonals.

Full to sparse conversion.

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

Working with sparse matrices.

- nnz - Number of nonzero matrix elements.
- nonzeros - Nonzero matrix elements.
- nzmax - Amount of storage allocated for nonzero matrix elements.
- spones - Replace nonzero sparse matrix elements with ones.
- spalloc - Allocate space for sparse matrix.
- issparse - True for sparse matrix.
- spfun - Apply function to nonzero matrix elements.
- spy - Visualize sparsity pattern.

Reordering algorithms.

- amd - Approximate minimum degree permutation.
- colamd - Column approximate minimum degree permutation.
- symamd - Symmetric approximate minimum degree permutation.
- colmmd - Column minimum degree permutation.
- symmmd - Symmetric minimum degree permutation.
- symrcm - Symmetric reverse Cuthill-McKee permutation.
- colperm - Column permutation.
- randperm - Random permutation.
- dmperm - Dulmage-Mendelsohn permutation.

Linear algebra.

- eigs - A few eigenvalues, using ARPACK.
- svds - A few singular values, using eigs.
- ilu - Incomplete LU factorization.
- luinc - Incomplete LU factorization.
- cholinc - Incomplete Cholesky factorization.
- normest - Estimate the matrix 2-norm.
- condest - 1-norm condition number estimate.
- sprank - Structural rank.

Linear Equations (iterative methods).

- pcg** - Preconditioned Conjugate Gradients Method.
- bicg** - BiConjugate Gradients Method.
- bicgstab** - BiConjugate Gradients Stabilized Method.
- cgs** - Conjugate Gradients Squared Method.
- gmres** - Generalized Minimum Residual Method.
- lsqr** - LSQR Method.
- minres** - Minimum Residual Method.
- qmr** - Quasi-Minimal Residual Method.
- symmlq** - Symmetric LQ Method.

Operations on graphs (trees).

- treelayout** - Lay out tree or forest.
- treeplot** - Plot picture of tree.
- etree** - Elimination tree.
- etreeplot** - Plot elimination tree.
- gplot** - Plot graph, as in "graph theory".

Miscellaneous.

- symbfact** - Symbolic factorization analysis.
- spparms** - Set parameters for sparse matrix routines.
- spaugment** - Form least squares augmented system.

1.21 Specialized math functions

Specialized math functions.

<code>airy</code>	- Airy functions.
<code>besselj</code>	- Bessel function of the first kind.
<code>bessely</code>	- Bessel function of the second kind.
<code>besselh</code>	- Bessel functions of the third kind (Hankel function).
<code>besseli</code>	- Modified Bessel function of the first kind.
<code>besselk</code>	- Modified Bessel function of the second kind.
<code>beta</code>	- Beta function.
<code>betainc</code>	- Incomplete beta function.
<code>betaln</code>	- Logarithm of beta function.
<code>ellipj</code>	- Jacobi elliptic functions.
<code>ellipke</code>	- Complete elliptic integral.
<code>erf</code>	- Error function.
<code>erfc</code>	- Complementary error function.
<code>erfcx</code>	- Scaled complementary error function.
<code>erfinv</code>	- Inverse error function.
<code>expint</code>	- Exponential integral function.
<code>gamma</code>	- Gamma function.
<code>gammainc</code>	- Incomplete gamma function.
<code>gammaln</code>	- Logarithm of gamma function.
<code>psi</code>	- Psi (polygamma) function.
<code>legendre</code>	- Associated Legendre function.
<code>cross</code>	- Vector cross product.
<code>dot</code>	- Vector dot product.

Number theoretic functions.

<code>factor</code>	- Prime factors.
<code>isprime</code>	- True for prime numbers.
<code>primes</code>	- Generate list of prime numbers.
<code>gcd</code>	- Greatest common divisor.
<code>lcm</code>	- Least common multiple.
<code>rat</code>	- Rational approximation.
<code>rats</code>	- Rational output.
<code>perms</code>	- All possible permutations.
<code>nchoosek</code>	- All combinations of N elements taken K at a time.
<code>factorial</code>	- Factorial function.

Coordinate transforms.

<code>cart2sph</code>	- Transform Cartesian to spherical coordinates.
<code>cart2pol</code>	- Transform Cartesian to polar coordinates.
<code>pol2cart</code>	- Transform polar to Cartesian coordinates.
<code>sph2cart</code>	- Transform spherical to Cartesian coordinates.
<code>hsv2rgb</code>	- Convert hue-saturation-value colors to red-green-blue.
<code>rgb2hsv</code>	- Convert red-green-blue colors to hue-saturation-value.

1.22 Specialized graphs

Specialized 2-D graphs.

<code>area</code>	- Filled area plot.
<code>bar</code>	- Bar graph.
<code>barh</code>	- Horizontal bar graph.
<code>comet</code>	- Comet-like trajectory.
<code>compass</code>	- Compass plot.
<code>errorbar</code>	- Error bar plot.
<code>ezplot</code>	- Easy to use function plotter.
<code>ezpolar</code>	- Easy to use polar coordinate plotter.
<code>feather</code>	- Feather plot.
<code>fill</code>	- Filled 2-D polygons.
<code>fplot</code>	- Plot function.
<code>hist</code>	- Histogram.
<code>pareto</code>	- Pareto chart.
<code>pie</code>	- Pie chart.
<code>plotmatrix</code>	- Scatter plot matrix.
<code>rose</code>	- Angle histogram plot.
<code>scatter</code>	- Scatter plot.
<code>stem</code>	- Discrete sequence or "stem" plot.
<code>stairs</code>	- Stairstep plot.

Contour and 2-1/2 D graphs.

<code>contour</code>	- Contour plot.
<code>contourf</code>	- Filled contour plot.
<code>contour3</code>	- 3-D Contour plot.
<code>clabel</code>	- Contour plot elevation labels.
<code>ezcontour</code>	- Easy to use contour plotter.
<code>ezcontourf</code>	- Easy to use filled contour plotter.
<code>pcolor</code>	- Pseudocolor (checkerboard) plot.
<code>voronoi</code>	- Voronoi diagram.

Specialized 3-D graphs.

<code>bar3</code>	- 3-D bar graph.
<code>bar3h</code>	- Horizontal 3-D bar graph.
<code>comet3</code>	- 3-D comet-like trajectories.
<code>ezgraph3</code>	- General purpose surface plotter.
<code>ezmesh</code>	- Easy to use 3-D mesh plotter.
<code>ezmeshc</code>	- Easy to use combination mesh/contour plotter.
<code>ezplot3</code>	- Easy to use 3-D parametric curve plotter.
<code>ezsurf</code>	- Easy to use 3-D colored surface plotter.
<code>ezsurf c</code>	- Easy to use combination surf/contour plotter.
<code>meshc</code>	- Combination mesh/contour plot.
<code>meshz</code>	- 3-D mesh with curtain.
<code>pie3</code>	- 3-D pie chart.
<code>ribbon</code>	- Draw 2-D lines as ribbons in 3-D.
<code>scatter3</code>	- 3-D scatter plot.
<code>stem3</code>	- 3-D stem plot.
<code>surf c</code>	- Combination surf/contour plot.
<code>trisurf</code>	- Triangular surface plot.
<code>trimesh</code>	- Triangular mesh plot.
<code>waterfall</code>	- Waterfall plot.

Volume and vector visualization.

<code>vissuite</code>	- Visualization suite.
<code>isosurface</code>	- Isosurface extractor.
<code>isonormals</code>	- Isosurface normals.
<code>isocaps</code>	- Isosurface end caps.
<code>isocolors</code>	- Isosurface and patch colors.
<code>contourslice</code>	- Contours in slice planes.
<code>slice</code>	- Volumetric slice plot.
<code>streamline</code>	- Streamlines from 2D or 3D vector data.
<code>stream3</code>	- 3D streamlines.
<code>stream2</code>	- 2D streamlines.
<code>quiver3</code>	- 3D quiver plot.
<code>quiver</code>	- 2D quiver plot.
<code>divergence</code>	- Divergence of a vector field.
<code>curl</code>	- Curl and angular velocity of a vector field.
<code>coneplot</code>	- 3D cone plot.
<code>streamtube</code>	- 3D stream tube.
<code>streamribbon</code>	- 3D stream ribbon.
<code>streamslice</code>	- Streamlines in slice planes.
<code>streamparticles</code>	- Display stream particles.
<code>interpstreamspeed</code>	- Interpolate streamline vertices from speed.
<code>subvolume</code>	- Extract subset of volume dataset.
<code>reducevolume</code>	- Reduce volume dataset.
<code>volumebounds</code>	- Returns x,y,z and color limits for volume data.
<code>smooth3</code>	- Smooth 3D data.
<code>reducepatch</code>	- Reduce number of patch faces.
<code>shrinkfaces</code>	- Reduce size of patch faces.

Images display and file I/O.

<code>image</code>	- Display image.
<code>imagesc</code>	- Scale data and display as image.
<code>colormap</code>	- Color look-up table.
<code>gray</code>	- Linear gray-scale color map.
<code>contrast</code>	- Gray scale color map to enhance image contrast.
<code>brighten</code>	- Brighten or darken color map.
<code>colorbar</code>	- Display color bar (color scale).
<code>imread</code>	- Read image from graphics file.
<code>imwrite</code>	- Write image to graphics file.
<code>imfinfo</code>	- Information about graphics file.
<code>im2java</code>	- Convert image to Java image.

Movies and animation.

<code>getframe</code>	- Get movie frame.
<code>movie</code>	- Play recorded movie frames.
<code>rotate</code>	- Rotate object about specified origin and direction.
<code>frame2im</code>	- Convert movie frame to indexed image.
<code>im2frame</code>	- Convert index image into movie format.

Color related functions.

<code>spinmap</code>	- Spin color map.
<code>rgbplot</code>	- Plot color map.
<code>colstyle</code>	- Parse color and style from string.
<code>ind2rgb</code>	- Convert indexed image to RGB image.

Solid modeling.

<code>cylinder</code>	- Generate cylinder.
<code>sphere</code>	- Generate sphere.
<code>ellipsoid</code>	- Generate ellipsoid.
<code>patch</code>	- Create patch.
<code>surf2patch</code>	- Convert surface data to patch data.

See also `graph2d`, `graph3d`.

1.23 Character strings

General.

- `char` - Create character array (string).
- `strings` - Help for strings.
- `cellstr` - Create cell array of strings from character array.
- `blanks` - String of blanks.
- `deblank` - Remove trailing blanks.

String tests.

- `iscellstr` - True for cell array of strings.
- `ischar` - True for character array (string).
- `isspace` - True for white space characters.
- `isstrprop` - Check if string elements are of a specified category.

String operations.

- `regex` - Match regular expression.
- `regexpi` - Match regular expression, ignoring case.
- `regexprep` - Replace string using regular expression.
- `strcat` - Concatenate strings.
- `strvcat` - Vertically concatenate strings.
- `strcmp` - Compare strings.
- `strncmp` - Compare first N characters of strings.
- `strcmpi` - Compare strings ignoring case.
- `strncmpi` - Compare first N characters of strings ignoring case.
- `strread` - Read formatted data from string.
- `findstr` - Find one string within another.
- `strfind` - Find one string within another.
- `strjust` - Justify character array.
- `strmatch` - Find possible matches for string.
- `strep` - Replace string with another.
- `strtok` - Find token in string.
- `strtrim` - Remove insignificant whitespace.
- `upper` - Convert string to uppercase.
- `lower` - Convert string to lowercase.

Character set conversion.

- `native2unicode` - Convert bytes to Unicode characters.
- `unicode2native` - Convert Unicode characters to bytes.

String to number conversion.

- `num2str` - Convert numbers to a string.
- `int2str` - Convert integer to string.
- `mat2str` - Convert a 2-D matrix to a string in MATLAB syntax.
- `str2double` - Convert string to double precision value.
- `str2num` - Convert string matrix to numeric array.
- `sprintf` - Write formatted data to string.
- `sscanf` - Read string under format control.

Base number conversion.

- hex2num - Convert hexadecimal string to double precision number.
- hex2dec - Convert hexadecimal string to decimal integer.
- dec2hex - Convert decimal integer to hexadecimal string.
- bin2dec - Convert binary string to decimal integer.
- dec2bin - Convert decimal integer to a binary string.
- base2dec - Convert base B string to decimal integer.
- dec2base - Convert decimal integer to base B string.
- num2hex - Convert singles and doubles to IEEE hexadecimal strings.

See also `general`, `lang`, `iofun`, `ops`, `datatypes`.

1.24 Time and dates

Current date and time.

- `now` - Current date and time as date number.
- `date` - Current date as date string.
- `clock` - Current date and time as date vector.

Basic functions.

- `datenum` - Serial date number.
- `datestr` - String representation of date.
- `datevec` - Date components.

Date functions.

- `calendar` - Calendar.
- `weekday` - Day of week.
- `eomday` - End of month.
- `datetick` - Date formatted tick labels.

Timing functions.

- `cputime` - CPU time in seconds.
- `tic` - Start stopwatch timer.
- `toc` - Stop stopwatch timer.
- `etime` - Elapsed time.
- `pause` - Wait in seconds.

1.25 Graphical User Interface Tools

GUI functions.

<code>uicontrol</code>	- Create user interface control.
<code>uimenu</code>	- Create user interface menu.
<code>dragrect</code>	- Drag XOR rectangles with mouse.
<code>ginput</code>	- Graphical input from mouse.
<code>selectmoveresize</code>	- Interactively select, move, resize, or copy objects.
<code>uipanel</code>	- Uipanel container object.
<code>uirestore</code>	- Restores the interactive functionality figure window.
<code>uiresume</code>	- Resume execution of blocked M-file.
<code>uistack</code>	- Reorder the visual stacking order of objects.
<code>uisuspend</code>	- suspends all interactive properties of the figure.
<code>uiwait</code>	- Block execution and wait for resume.
<code>waitfor</code>	- Block execution and wait for event.
<code>waitforbuttonpress</code>	- Wait for key/buttonpress over figure.

GUI design tools.

<code>align</code>	- Align uicontrols and axes.
<code>inspect</code>	- Open the inspector and inspect object properties
<code>propedit</code>	- Graphical property editor

Dialog boxes.

<code>dialog</code>	- Create dialog figure.
<code>errordlg</code>	- Error dialog box.
<code>helpdlg</code>	- Help dialog box.
<code>inputdlg</code>	- Input dialog box.
<code>listdlg</code>	- List selection dialog box.
<code>makemenu</code>	- Create menu structure.
<code>menu</code>	- Generate a menu of choices for user input.
<code>msgbox</code>	- Message box.
<code>questdlg</code>	- Question dialog box.
<code>tabdlg</code>	- Create and manage tabbed dialog box.
<code>uigetdir</code>	- Standard open directory dialog box
<code>uigetfile</code>	- Standard open file dialog box.
<code>uigetpref</code>	- question dialog box with preference support
<code>uiputfile</code>	- Standard save file dialog box.
<code>uiload</code>	- Present file selection dialog and load result using LOAD
<code>uiopen</code>	- Present file selection dialog with appropriate file filters.
<code>uisave</code>	- GUI Helper function for SAVE
<code>uisetcolor</code>	- Color selection dialog box.
<code>uisetfont</code>	- Font selection dialog box.
<code>waitbar</code>	- Display wait bar.
<code>warndlg</code>	- Warning dialog box.

Preferences.

- addpref - Add preference.
- getpref - Get preference.
- ispref - Test for existence of preference.
- rmpref - Remove preference.
- setpref - Set preference.

Miscellaneous utilities.

- allchild - Get all object children
- clipboard - Copy and paste strings to and from system clipboard.
- figname - Chooses next available figure name.
- findfigs - Position figures on to screen.
- getpixelposition - Get the position of an HG object in pixel units.
- getptr - Get figure pointer.
- listfonts - Get list of available system fonts in cell array.
- movegui - Move a figure window to a specified position on the screen.
- guidata - Store or retrieve application data.
- guihandles - Return a structure of handles.
- overobj - Get handle of object the pointer is over.
- remapfig - Transform figure objects' positions.
- setpixelposition - Set position HG object in pixels.
- setptr - Set figure pointer.
- textwrap - Return wrapped string matrix for given UI Control.
- uibuttongroup - Component to exclusively manage radiobuttons/togglebuttons.
- uiclearmode - Clears the current interactive figure mode;
- uipushtool - Create a pushbutton in the toolbar of a figure window.
- uisetpref - manages preferences used in UIGETPREF
- uitoggletool - Create a togglebutton in the toolbar of a figure window.
- uitoolbar - Create a toolbar in a figure window.

1.26 Windows Operating System Interface Files (COM/DDE)

COM Automation Client Functions.

actxcontrol	- Create an ActiveX control.
actxserver	- Create an ActiveX server.
eventlisteners	- Lists all events that are registered.
isevent	- True if event of object.
registerevent	- Registers events for a specified control at runtime.
unregisterallevents	- Unregister all events for a specified control at runtime.
unregisterevent	- Unregister events for a specified control at runtime.
iscom	- True if input handle is a COM/ActiveX object.
isinterface	- True if input handle is a COM Interface.
COM/set	- Set a property value on a COM object.
COM/get	- Get COM object properties.
COM/invoke	- Invoke method on object or interface, or display methods.
COM/events	- Return list of events the COM object can trigger.
COM/interfaces	- List custom interfaces supported by a COM server.
COM/addproperty	- Add custom property to an object.
COM/deleteproperty	- Remove custom property from object.
COM/delete	- Delete a COM object.
COM/release	- Release a COM interface.
COM/move	- Move and/or resize an ActiveX control in its parent window.
COM/propedit	- Invoke the property page.
COM/save	- Serialize a COM object to a file.
COM/load	- Initialize a COM object from a file.

COM Sample code.

mwsamp	- Sample Activex control creation.
sampev	- Sample event handler for ActiveX server.

DDE Client Functions.

ddeadv	- Set up advisory link.
ddeexec	- Send string for execution.
ddeinit	- Initiate DDE conversation.
ddepoke	- Send data to application.
ddereq	- Request data from application.
ddeterm	- Terminate DDE conversation.
ddeunadv	- Release advisory link.

Other

winopen	- Open a file using the appropriate Windows command.
winqueryreg	- Get information from the Windows registry.

2. SIMULINK

2.1 Simulink block library

Block libraries.

```
simulink      - Open main block library.  
simulink_extras - Simulink Extras block library.
```

Example S-function models and blocks.

```
simo          - simo model, block diagram form.  
simom         - simo model, M-file form.  
simom2        - simo model, M-file form #2.  
simosys       - simo model, S-function block form.  
vdpm          - Van der Pol model, M-file form.  
mixed         - Mixed continuous/discrete model, block diagram form.  
mixedm        - Mixed continuous/discrete model, M-file form.  
limintm       - Limited integrator block, M-file form.  
vlimintm      - Vectorized limited integrator, M-file form.  
vdlmintm      - Discrete-time vectorized limited integrator, M-file.  
sfuntmpl      - M-file S-function template.  
csfunc        - Continuous-time model, M-file form.  
dsfunc        - Discrete-time model, M-file form.  
vsfunc        - Variable sample-time model, M-file form.  
sfuncont      - M-file S-function template, continuous-time model.  
sfundsc1      - M-file S-function template, discrete real-time model.  
sfundsc2      - M-file S-function template, discrete sampled model.  
timestwo      - M-file S-function example.
```

The `MATLABROOT/simulink` directory contains the source code for examples of C, Fortran, and Ada implementations of the above blocks.

2.2 Simulink components

```
slmldiscui    - Launch Simulink Model Discretizer UI  
sldiscmdl     - Discretize Simulink model block by block
```

2.3 Simulink Fixed Point utilities

Utilities for creating fixed-point data:

```

fixdt - Create an object describing a fixed-point or floating-point data type.
sfix  - Create structure describing Signed FIXEd point data type.
sfrac - Create structure describing Signed FRACTIONal data type.
sint  - Create structure describing Signed INTeger data type.
ufix  - Create structure describing Unsigned FIXEd point data type.
ufrac - Create structure describing Unsigned FRACTIONal data type.
uint  - Create structure describing Unsigned INTeger data type.
float - Create structure describing a floating point data type.

```

Utilities for manipulating and displaying fixed-point data:

```

fixptbestexp - Determine the exponent that gives the best precision.
fixptbestprec - Determine the maximum precision that can be used in the fixed-
point
                representation of a value.
fxptdlg      - Graphical interface for fixed-point simulation log.
fxptplt      - Graphical interface for plotting blocks in a fixed-point system.
num2fixpt    - Quantize a value using a fixed point representation.
fixpt_interp1 - Fixed-point 1-D interpolation (table lookup).
fixpt_look1_func_approx - Find points for lookup table approximate to a function.
fixpt_look1_func_plot - Plot an ideal function and its lookup approximation.

```

Run HELP on any of these commands for usage information. For example, typing

```
help fixdt
```

at the MATLAB prompt will display the usage information related to the FIXDT function.

2.4 Simulink demonstrations and examples

Models and demos that correspond to the demos included in the MATLAB "demo" facility are located in the subdirectories corresponding to their demo categories:

```

simgeneral - General model examples.
automotive - Automotive examples.
aerospace  - Aerospace examples.
simfeatures - Demos of specific features.

```

Type "demo simulink" to bring up the demos section of the Help system.

2.5 Simulink

Model analysis and construction functions.

Simulation

<code>sim</code>	- Simulate a Simulink model.
<code>sldebug</code>	- Debug a Simulink model.
<code>simset</code>	- Define options to SIM Options structure.
<code>simget</code>	- Get SIM Options structure

Linearization and trimming.

<code>linmod</code>	- Extract linear model from continuous-time system.
<code>linmod2</code>	- Extract linear model, advanced method.
<code>dlinmod</code>	- Extract linear model from discrete-time system.
<code>trim</code>	- Find steady-state operating point.

Model Construction.

<code>close_system</code>	- Close open model or block.
<code>new_system</code>	- Create new empty model window.
<code>open_system</code>	- Open existing model or block.
<code>load_system</code>	- Load existing model without making model visible.
<code>save_system</code>	- Save an open model.
<code>add_block</code>	- Add new block.
<code>add_line</code>	- Add new line.
<code>delete_block</code>	- Remove block.
<code>delete_line</code>	- Remove line.
<code>find_system</code>	- Search a model.
<code>hilite_system</code>	- Hilite objects within a model.
<code>replace_block</code>	- Replace existing blocks with a new block.
<code>set_param</code>	- Set parameter values for model or block.
<code>get_param</code>	- Get simulation parameter values from model.
<code>add_param</code>	- Add a user-defined string parameter to a model.
<code>delete_param</code>	- Delete a user-defined parameter from a model.
<code>bdclose</code>	- Close a Simulink window.
<code>bdroot</code>	- Root level model name.
<code>gcb</code>	- Get the name of the current block.
<code>gcbh</code>	- Get the handle of the current block.
<code>gcs</code>	- Get the name of the current system.
<code>getfullname</code>	- get the full path name of a block
<code>slupdate</code>	- Update older 1.x models to 3.x.
<code>addterms</code>	- Add terminators to unconnected ports.
<code>boolean</code>	- Convert numeric array to boolean.
<code>slhelp</code>	- Simulink user's guide or block help.

Masking.

- `hasmask` - Check for mask.
- `hasmaskdlg` - Check for mask dialog.
- `hasmaskicon` - Check for mask icon.
- `iconedit` - Design block icons using `ginput` function.
- `maskpopups` - Return and change masked block's popup menu items.
- `movemask` - Restructure masked built-in blocks as masked subsystems.

Library.

- `libinfo` - Get library information for a system.

Diagnostics.

- `sllastdiagnostic` - Last diagnostic array.
- `sllasterror` - Last error array.
- `sllastwarning` - Last warning array.
- `sldiagnostics` - Get block count and compile stats for a model.

Hardcopy and printing.

- `frameedit` - Edit print frames for annotated model printouts.
- `print` - Print graph or Simulink system; or save graph to M-file.
- `printopt` - Printer defaults.
- `orient` - Set paper orientation.

See also `blocks` and `simdemos`.

3. OPTIMIZATION TOOLBOX

Nonlinear minimization of functions.

- `fminbnd` - Scalar bounded nonlinear function minimization.
- `fmincon` - Multidimensional constrained nonlinear minimization.
- `fminsearch` - Multidimensional unconstrained nonlinear minimization, by Nelder-Mead direct search method.
- `fminunc` - Multidimensional unconstrained nonlinear minimization.
- `fseminf` - Multidimensional constrained minimization, semi-infinite constraints.

Nonlinear minimization of multi-objective functions.

- `fgoalattain` - Multidimensional goal attainment optimization
- `fminimax` - Multidimensional minimax optimization.

Linear least squares (of matrix problems).

- `lsqlin` - Linear least squares with linear constraints.
- `lsqnonneg` - Linear least squares with nonnegativity constraints.

Nonlinear least squares (of functions).

- `lsqcurvefit` - Nonlinear curvefitting via least squares (with bounds).
- `lsqnonlin` - Nonlinear least squares with upper and lower bounds.

Nonlinear zero finding (equation solving).

- `fzero` - Scalar nonlinear zero finding.
- `fsolve` - Nonlinear system of equations solve (function solve).

Minimization of matrix problems.

- `bintprog` - Binary integer (linear) programming.
- `linprog` - Linear programming.
- `quadprog` - Quadratic programming.

Controlling defaults and options.

- `optimset` - Create or alter optimization OPTIONS structure.
- `optimget` - Get optimization parameters from OPTIONS structure.

Graphical user interface and plot routines

- `optimtool` - Optimization Toolbox Graphical User Interface
- `optimplotconstrviolation` - Plot max. constraint violation at each iteration
- `optimplotfirstorderopt` - Plot first-order optimality at each iteration
- `optimplotresnorm` - Plot value of the norm of residuals at each iteration
- `optimplotstepsize` - Plot step size at each iteration

4. SIGNAL PROCESSING TOOLBOX

4.1 Discrete-time filter design, analysis, and implementation

FIR filter design

<code>cfirpm</code>	- Complex and nonlinear phase equiripple FIR filter design
<code>fir1</code>	- Window based FIR filter design - low, high, band, stop, multi
<code>fir2</code>	- FIR arbitrary shape filter design using the frequency sampling method
<code>fircls</code>	- Constrained Least Squares filter design - arbitrary response
<code>fircls1</code>	- Constrained Least Squares FIR filter design - low and highpass
<code>firls</code>	- Optimal least-squares FIR filter design
<code>firpm</code>	- Parks-McClellan optimal equiripple FIR filter design
<code>firpmord</code>	- Parks-McClellan optimal equiripple FIR order estimator
<code>intfilt</code>	- Interpolation FIR filter design
<code>kaiserord</code>	- Kaiser window design based filter order estimation
<code>sgolay</code>	- Savitzky-Golay FIR smoothing filter design

Communications filters

<code>firrcos</code>	- Raised cosine FIR filter design
<code>gaussfir</code>	- Gaussian FIR Pulse-Shaping Filter Design

IIR digital filter design

<code>butter</code>	- Butterworth filter design
<code>cheby1</code>	- Chebyshev Type I filter design (passband ripple)
<code>cheby2</code>	- Chebyshev Type II filter design (stopband ripple)
<code>ellip</code>	- Elliptic filter design
<code>maxflat</code>	- Generalized Butterworth lowpass filter design
<code>yulewalk</code>	- Yule-Walker filter design

IIR filter order estimation

<code>buttord</code>	- Butterworth filter order estimation
<code>cheblord</code>	- Chebyshev Type I filter order estimation
<code>cheb2ord</code>	- Chebyshev Type II filter order estimation
<code>ellipord</code>	- Elliptic filter order estimation

Filter analysis

<code>abs</code>	- Magnitude
<code>angle</code>	- Phase angle
<code>filternorm</code>	- Compute the 2-norm or inf-norm of a digital filter
<code>freqz</code>	- Z-transform frequency response
<code>fvtool</code>	- Filter Visualization Tool
<code>grpdelay</code>	- Group delay
<code>impz</code>	- Discrete impulse response
<code>phasedelay</code>	- Phase delay of a digital filter
<code>phasez</code>	- Digital filter phase response (unwrapped)
<code>stepz</code>	- Digital filter step response
<code>unwrap</code>	- Unwrap phase angle
<code>zerophase</code>	- Zero-phase response of a real filter
<code>zplane</code>	- Discrete pole-zero plot

Filter implementation

<code>conv</code>	- Convolution
<code>conv2</code>	- 2-D convolution
<code>convmtx</code>	- Convolution matrix
<code>deconv</code>	- Deconvolution
<code>fftfilt</code>	- Overlap-add filter implementation
<code>filter</code>	- Filter implementation
<code>filter2</code>	- Two-dimensional digital filtering
<code>filtfilt</code>	- Zero-phase version of filter
<code>filtic</code>	- Determine filter initial conditions
<code>latcfilt</code>	- Lattice filter implementation
<code>medfilt1</code>	- 1-Dimensional median filtering
<code>sgolayfilt</code>	- Savitzky-Golay filter implementation
<code>sosfilt</code>	- Second-order sections (biquad) filter implementation
<code>upfirdn</code>	- Upsample, FIR filter, downsample

4.2 Analog filter design, transformation, and discretization

Analog lowpass filter prototypes

- besselap - Bessel filter prototype
- buttap - Butterworth filter prototype
- cheblap - Chebyshev Type I filter prototype (passband ripple)
- cheb2ap - Chebyshev Type II filter prototype (stopband ripple)
- ellipap - Elliptic filter prototype

Analog filter design

- besself - Bessel analog filter design
- butter - Butterworth filter design
- cheby1 - Chebyshev Type I filter design
- cheby2 - Chebyshev Type II filter design
- ellip - Elliptic filter design

Filter analysis

- abs - Magnitude
- angle - Phase angle
- freqs - Laplace transform frequency response
- freqspace - Frequency spacing for frequency response

Analog filter transformation

- lp2bp - Lowpass to bandpass analog filter transformation
- lp2bs - Lowpass to bandstop analog filter transformation
- lp2hp - Lowpass to highpass analog filter transformation
- lp2lp - Lowpass to lowpass analog filter transformation

Filter discretization

- bilinear - Bilinear transformation with optional prewarping
- impinvar - Impulse invariance analog to digital conversion

4.3 Linear system transformations

latc2tf	- Lattice or lattice ladder to transfer function conversion
polyscale	- Scale roots of polynomial
polystab	- Polynomial stabilization
residuez	- Z-transform partial fraction expansion
sos2ss	- Second-order sections to state-space conversion
sos2tf	- Second-order sections to transfer function conversion
sos2zp	- Second-order sections to zero-pole conversion
ss2sos	- State-space to second-order sections conversion
ss2tf	- State-space to transfer function conversion
ss2zp	- State-space to zero-pole conversion
tf2latc	- Transfer function to lattice or lattice ladder conversion
tf2sos	- Transfer Function to second-order sections conversion
tf2ss	- Transfer function to state-space conversion
tf2zpk	- Discrete-time transfer function to zero-pole conversion
zp2sos	- Zero-pole to second-order sections conversion
zp2ss	- Zero-pole to state-space conversion
zp2tf	- Zero-pole to transfer function conversion

4.4 Windows

barthannwin	- Modified Bartlett-Hanning window
bartlett	- Bartlett window
blackman	- Blackman window
blackmanharris	- Minimum 4-term Blackman-Harris window
bohmanwin	- Bohman window
chebwin	- Chebyshev window
dpss	- Discrete prolate spheroidal sequences (Slepian sequences)
dpsscldir	- Remove discrete prolate spheroidal sequences from database
dpssdir	- Discrete prolate spheroidal sequence database directory
dpssload	- Load discrete prolate spheroidal sequences from database
dpsssave	- Save discrete prolate spheroidal sequences in database
flattopwin	- Flat Top window
gausswin	- Gaussian window
hamming	- Hamming window
hann	- Hann window
kaiser	- Kaiser window
nuttallwin	- Nuttall defined minimum 4-term Blackman-Harris window
parzenwin	- Parzen (de la Valle-Poussin) window
rectwin	- Rectangular window
taylorwin	- Taylor window
triang	- Triangular window
tukeywin	- Tukey window
wvtool	- Window Visualization Tool
window	- Window function gateway

4.5 Transforms

<code>bitrevorder</code>	- Permute input into bit-reversed order
<code>czft</code>	- Chirp-z transform
<code>dct</code>	- Discrete cosine transform
<code>dftmtx</code>	- Discrete Fourier transform matrix
<code>digitrevorder</code>	- Permute input into digit-reversed order
<code>fft</code>	- Fast Fourier transform
<code>fft2</code>	- 2-D fast Fourier transform
<code>fftshift</code>	- Swap vector halves
<code>goertzel</code>	- Second-order Goertzel algorithm
<code>hilbert</code>	- Discrete-time analytic signal via Hilbert transform
<code>idct</code>	- Inverse discrete cosine transform
<code>ifft</code>	- Inverse fast Fourier transform
<code>ifft2</code>	- Inverse 2-D fast Fourier transform

4.6 Cepstral analysis

<code>cceps</code>	- Complex cepstrum
<code>icceps</code>	- Inverse Complex cepstrum
<code>rceps</code>	- Real cepstrum and minimum phase reconstruction

4.7 Statistical signal processing and spectral analysis

<code>corrcoef</code>	- Correlation coefficients.
<code>corrmtx</code>	- Autocorrelation matrix
<code>cov</code>	- Covariance matrix
<code>cpsd</code>	- Cross Power Spectral Density
<code>mscohere</code>	- Magnitude squared coherence estimate
<code>pburg</code>	- Burg's PSD estimation method
<code>pcov</code>	- Covariance PSD estimation method
<code>peig</code>	- Eigenvector PSD estimation method
<code>periodogram</code>	- Periodogram PSD estimation method
<code>pmcov</code>	- Modified Covariance PSD estimation method
<code>pmtm</code>	- Thomson multitaper PSD estimation method
<code>pmusic</code>	- Multiple Signal Classification PSD estimation method
<code>pwelch</code>	- Welch's PSD estimation method
<code>pyulear</code>	- Yule-Walker AR PSD estimation method
<code>rooteig</code>	- Sinusoid frequency and power estimation via the eigenvector algorithm
<code>rootmusic</code>	- Sinusoid frequency and power estimation via the MUSIC algorithm
<code>spectrogram</code>	- Spectrogram using a Short-Time Fourier Transform (STFT)
<code>tfestimate</code>	- Transfer function estimate
<code>xcorr</code>	- Cross-correlation function
<code>xcorr2</code>	- 2-D cross-correlation
<code>xcov</code>	- Covariance function
Spectral estimators	
<code>spectrum</code>	- Family of PSD, mean-square (power), and pseudo spectrum estimators
DSP data objects	
<code>dspdata</code>	- PSD, mean-square (power) spectrum, and pseudo spectrum data container

4.8 Parametric modeling

<code>arburg</code>	- AR parametric modeling via Burg's method
<code>arconv</code>	- AR parametric modeling via covariance method
<code>armcov</code>	- AR parametric modeling via modified covariance method
<code>aryule</code>	- AR parametric modeling via the Yule-Walker method
<code>ident</code>	- See the System Identification Toolbox
<code>invfreqs</code>	- Analog filter fit to frequency response
<code>invfreqz</code>	- Discrete filter fit to frequency response
<code>prony</code>	- Prony's discrete filter fit to time response
<code>stmcb</code>	- Steiglitz-McBride iteration for ARMA modeling

4.9 Linear prediction

<code>ac2poly</code>	- Autocorrelation sequence to prediction polynomial conversion
<code>ac2rc</code>	- Autocorrelation sequence to reflection coefficients conversion
<code>is2rc</code>	- Inverse sine parameters to reflection coefficients conversion
<code>lar2rc</code>	- Log area ratios to reflection coefficients conversion
<code>levinson</code>	- Levinson-Durbin recursion
<code>lpc</code>	- Linear Predictive Coefficients using autocorrelation method
<code>lsf2poly</code>	- Line spectral frequencies to prediction polynomial conversion
<code>poly2ac</code>	- Prediction polynomial to autocorrelation sequence conversion
<code>poly2lsf</code>	- Prediction polynomial to line spectral frequencies conversion
<code>poly2rc</code>	- Prediction polynomial to reflection coefficients conversion
<code>rc2ac</code>	- Reflection coefficients to autocorrelation sequence conversion
<code>rc2is</code>	- Reflection coefficients to inverse sine parameters conversion
<code>rc2lar</code>	- Reflection coefficients to log area ratios conversion
<code>rc2poly</code>	- Reflection coefficients to prediction polynomial conversion
<code>rlevinson</code>	- Reverse Levinson-Durbin recursion
<code>schurrc</code>	- Schur algorithm

4.10 Multirate signal processing

<code>decimate</code>	- Resample data at a lower sample rate
<code>downsample</code>	- Downsample input signal
<code>interp</code>	- Resample data at a higher sample rate
<code>interp1</code>	- General 1-D interpolation
<code>resample</code>	- Resample sequence with new sampling rate
<code>spline</code>	- Cubic spline interpolation
<code>upfirdn</code>	- Up sample, FIR filter, down sample
<code>upsample</code>	- Upsample input signal

4.11 Waveform generation

<code>chirp</code>	- Swept-frequency cosine generator
<code>diric</code>	- Dirichlet (periodic sinc) function
<code>gauspuls</code>	- Gaussian RF pulse generator
<code>gmonopuls</code>	- Gaussian monopulse generator
<code>pulstran</code>	- Pulse train generator
<code>rectpuls</code>	- Sampled aperiodic rectangle generator
<code>sawtooth</code>	- Sawtooth function
<code>sinc</code>	- Sinc or $\sin(\pi*x)/(\pi*x)$ function
<code>square</code>	- Square wave function
<code>tripuls</code>	- Sampled aperiodic triangle generator
<code>vco</code>	- Voltage controlled oscillator

4.12 Specialized operations

<code>buffer</code>	- Buffer a signal vector into a matrix of data frames
<code>cell2sos</code>	- Convert cell array to second-order-section matrix
<code>cplxpair</code>	- Order vector into complex conjugate pairs
<code>demod</code>	- Demodulation for communications simulation
<code>eqtflength</code>	- Equalize the length of a discrete-time transfer function
<code>modulate</code>	- Modulation for communications simulation
<code>seqperiod</code>	- Find minimum-length repeating sequence in a vector
<code>sos2cell</code>	- Convert second-order-section matrix to cell array
<code>stem</code>	- Plot discrete data sequence
<code>strips</code>	- Strip plot
<code>udocode</code>	- Uniform decoding of the input
<code>uencode</code>	- Uniform quantization and encoding of the input into N-bits

4.13 Graphical User Interfaces

<code>fdatool</code>	- Filter Design and Analysis Tool
<code>fvtool</code>	- Filter Visualization Tool
<code>sptool</code>	- Signal Processing Tool
<code>wintool</code>	- Window Design and Analysis Tool
<code>wvtool</code>	- Window Visualization Tool

5. SYMBOLIC MATH TOOLBOX

Calculus.

<code>diff</code>	- Differentiate.
<code>int</code>	- Integrate.
<code>limit</code>	- Limit.
<code>taylor</code>	- Taylor series.
<code>jacobian</code>	- Jacobian matrix.
<code>symsum</code>	- Summation of series.

Linear Algebra.

<code>diag</code>	- Create or extract diagonals.
<code>triu</code>	- Upper triangle.
<code>tril</code>	- Lower triangle.
<code>inv</code>	- Matrix inverse.
<code>det</code>	- Determinant.
<code>rank</code>	- Rank.
<code>rref</code>	- Reduced row echelon form.
<code>null</code>	- Basis for null space.
<code>colspace</code>	- Basis for column space.
<code>eig</code>	- Eigenvalues and eigenvectors.
<code>svd</code>	- Singular values and singular vectors.
<code>jordan</code>	- Jordan canonical (normal) form.
<code>poly</code>	- Characteristic polynomial.
<code>expm</code>	- Matrix exponential.
<code>mldivide</code>	- \ matrix left division.
<code>mpower</code>	- ^ matrix power.
<code>mrdivide</code>	- / matrix right division.
<code>mtimes</code>	- * matrix multiplication.
<code>transpose</code>	- .' matrix transpose.
<code>ctranspose</code>	- ' matrix complex conjugate transpose.

Simplification.

<code>simplify</code>	- Simplify.
<code>expand</code>	- Expand.
<code>factor</code>	- Factor.
<code>collect</code>	- Collect.
<code>simple</code>	- Search for shortest form.
<code>numden</code>	- Numerator and denominator.
<code>horner</code>	- Nested polynomial representation.
<code>subexpr</code>	- Rewrite in terms of subexpressions.
<code>coeffs</code>	- Coefficients of a multivariate polynomial.
<code>sort</code>	- Sort symbolic vectors or polynomials.
<code>subs</code>	- Symbolic substitution.

Solution of Equations.

- `solve` - Symbolic solution of algebraic equations.
- `dsolve` - Symbolic solution of differential equations.
- `finverse` - Functional inverse.
- `compose` - Functional composition.

Variable Precision Arithmetic.

- `vpa` - Variable precision arithmetic.
- `digits` - Set variable precision accuracy.

Integral Transforms.

- `fourier` - Fourier transform.
- `laplace` - Laplace transform.
- `ztrans` - Z transform.
- `ifourier` - Inverse Fourier transform.
- `ilaplace` - Inverse Laplace transform.
- `iztrans` - Inverse Z transform.

Conversions.

- `double` - Convert symbolic matrix to double.
- `single` - Convert symbolic matrix to single precision.
- `poly2sym` - Coefficient vector to symbolic polynomial.
- `sym2poly` - Symbolic polynomial to coefficient vector.
- `char` - Convert sym object to string.
- `int8` - Convert to signed 8-bit integers.
- `int16` - Convert to signed 16-bit integers.
- `int32` - Convert to signed 32-bit integers.
- `int64` - Convert to signed 64-bit integers.
- `uint8` - Convert to unsigned 8-bit integers.
- `uint16` - Convert to unsigned 16-bit integers.
- `uint32` - Convert to unsigned 32-bit integers.
- `uint64` - Convert to unsigned 64-bit integers.

Symbolic Operations.

- `sym` - Create symbolic object.
- `syms` - Short-cut for constructing symbolic objects.
- `findsym` - Determine symbolic variables.
- `pretty` - Pretty print a symbolic expression.
- `latex` - LaTeX representation of a symbolic expression.
- `texlabel` - Produces the TeX format from a character string.
- `ccode` - C code representation of a symbolic expression.
- `fortran` - FORTRAN representation of a symbolic expression.

Arithmetic and Algebraic Operations.

plus	- +	addition.
minus	- -	subtraction.
uminus	- -	negation.
times	- .*	array multiplication.
ldivide	- \	left division.
rdivide	- /	right division.
power	- .^	array power.
abs	-	Absolute value.
ceil	-	Ceiling.
conj	-	Conjugate.
colon	-	Colon operator.
fix	-	Integer part.
floor	-	Floor.
frac	-	Fractional part.
mod	-	Mod.
round	-	Round.
quorem	-	Quotient and remainder.
imag	-	Imaginary part.
real	-	real part.
exp	-	Exponential.
log	-	Natural logarithm.
log10	-	Common logarithm.
log2	-	Base-2 logarithm.
sqrt	-	Square root.
prod	-	Product of the elements.
sum	-	Sum of the elements.

Logical Operations.

isreal	-	True for real array
eq	-	Equality test.
ne	-	Inequality test.

Special Functions.

besseli	-	Bessel function, I.
besselj	-	Bessel function, J.
besselk	-	Bessel function, K.
bessely	-	Bessel function, Y.
erf	-	Error function.
sinint	-	Sine integral.
cosint	-	Cosine integral.
zeta	-	Riemann zeta function.
gamma	-	Symbolic gamma function.
gcd	-	Greatest common divisor.
lcm	-	Least common multiple.
hypergeom	-	Generalized hypergeometric function.
lambertw	-	Lambert W function.
dirac	-	Delta function.
heaviside	-	Step function.

Trigonometric Functions.

<code>acos</code>	- Inverse cosine.
<code>acosh</code>	- Inverse hyperbolic cosine.
<code>acot</code>	- Inverse cotangent.
<code>acoth</code>	- Inverse hyperbolic cotangent.
<code>acsc</code>	- Inverse cosecant.
<code>acsch</code>	- Inverse hyperbolic cosecant.
<code>asec</code>	- Inverse secant.
<code>asech</code>	- Inverse hyperbolic secant.
<code>asin</code>	- Inverse sine.
<code>asinh</code>	- Inverse hyperbolic sine.
<code>atan</code>	- Inverse tangent.
<code>atanh</code>	- Inverse hyperbolic tangent.
<code>cos</code>	- Cosine function.
<code>cosh</code>	- Hyperbolic cosine.
<code>cot</code>	- Cotangent.
<code>coth</code>	- Hyperbolic cotangent.
<code>csc</code>	- Cosecant.
<code>csch</code>	- Hyperbolic cosecant.
<code>sec</code>	- Secant.
<code>sech</code>	- Hyperbolic secant.
<code>sin</code>	- Sine function.
<code>sinh</code>	- Hyperbolic sine.
<code>tan</code>	- Tangent function.
<code>tanh</code>	- Hyperbolic tangent.

String handling utilities.

<code>isvarname</code>	- Check for a valid variable name (MATLAB Toolbox).
<code>vectorize</code>	- Vectorize a symbolic expression.
<code>disp</code>	- Displays a sym as text.
<code>display</code>	- Display function for syms.
<code>eval</code>	- Evaluate a symbolic expression.

Pedagogical and Graphical Applications.

<code>rsums</code>	- Riemann sums.
<code>ezcontour</code>	- Easy to use contour plotter.
<code>ezcontourf</code>	- Easy to use filled contour plotter.
<code>ezmesh</code>	- Easy to use mesh (surface) plotter.
<code>ezmeshc</code>	- Easy to use combined mesh/contour plotter.
<code>ezplot</code>	- Easy to use function, implicit, and parametric curve plotter.
<code>ezplot3</code>	- Easy to use spatial curve plotter.
<code>ezpolar</code>	- Easy to use polar coordinates plotter.
<code>ezsurf</code>	- Easy to use surface plotter.
<code>ezsurfz</code>	- Easy to use combined surface/contour plotter.
<code>funtool</code>	- Function calculator.
<code>taylortool</code>	- Taylor series calculator.

Demonstrations.

- symintro** - Introduction to the Symbolic Toolbox.
- symcalcdemo** - Calculus demonstration.
- symlindemo** - Demonstrate symbolic linear algebra.
- symvpademo** - Demonstrate variable precision arithmetic
- symrotdemo** - Study plane rotations.
- symeqndemo** - Demonstrate symbolic equation solving.

Access to Maple.

- maple** - Access Maple kernel. (Not available with Student Version.)
- mfun** - Numeric evaluation of Maple functions.
- mfunlist** - List of functions for MFUN.
- mhhelp** - Maple help. (Not available with Student Version.)
- procread** - Install a Maple procedure. (Requires Extended Toolbox.)