List of built-in functions

abs(x)
absolute value of x, $|x|$

acos(x)
inverse cosinus of x, $\cos^{-1}(x)$

acosd(x)
inverse cosinus of x with angle in degrees

acot(x)
inverse cotangent of x, $\cot^{-1}(x)$

asin(x)
inverse sinus of x, $\sin^{-1}(x)$

asind(x)
inverse sinus of x with angle in degrees

atan(x)
inverse tangent of x, $\tan^{-1}(x)$

atand(x)
inverse tangent of x with angle in degrees

atan2(x,y)
element-wise inverse tangent of x/y

bar(x,y), bar(y)
produce bar graph of two vectors x and y
use indices of y as x-axis if one argument

cd *dir*
change current working directory

ceil(x)
return smallest integer $\geq$ x

char(x)
convert integer(s) into character(s)

clear var . . .
delete variable(s) from the symbol table

cos(x)
cosinus of x with argument in radians

cosd(x)
cosinus of x with argument in degrees

cot(x)
cotangent of x with argument in radians

cotd(x)
cotangent of x with argument in degrees

cputime()
return total CPU time spent so far

disp(x)
print the value of x with a newline

eps, eps(N,M)
machine precision as a number or a N×M-matrix

exp(x)
compute the exponential of x, e^x

eye(N), eye(N,M)
return $N \times N$ or $N \times M$ identity matrix

false
return logical 0 (false)

status = fclose(fid)
close file with file-id *fid*
return 0 on success, -1 on failure

fix(x)
round x to nearest integer towards zero

floor(x)
return largest integer $\leq$ x

fid *or* **[fid msg] = fopen(name,mode)**
open file with pathname *name*, mode character is
'r':read, 'w':write, 'a':append, 'r+':read and write
return file-id *fid* > 0 on success, -1 on error
optionally return error message *msg*

fprintf(fid,format,variable, . . .)
print variable(s) with specified formatting
%mf, %m.nf or %f:fixed-form float-point number
%me, %m.ne or %e:float-point with exponent
%md or %d:integer, %s:string, %%:%, \n:newline
m is field width, n is number of digits in fraction

val *or* **[val count] = fread(fid,sz, `'double'`)**
read *sz* elements from file *fid* to *val*
optionally return number of elements read in *count*
sz is any of inf:as much as possible, N:n elements,
[N M]:N×M matrix, [N inf]:M as large as possible

status = fseek(fid,offset,origin)
set file position to *offset* within open file *fid*
origin is 'bof', 'cof' or 'eof'
return 0 on success, -1 on failure

position = ftell(fid)
return file-pointer position of *fid*, -1 on failure

count = fwrite(fid,var, `'double'`)
write variable *var* to file *fid*
return number of elements written in *count*

grid on *or* **off**
turns grid on or off on a 2D-plot

hold off *or* **on**
do or do not erase previos plot before plotting next

val = input(msg) *or* input(msg, 's')
output *msg*, then read keyboard input to *val*
last form reads input as a string (does not evaluate)

intmax
return largest (32-bit) integer available

intmin
return smallest (32-bit) integer available

A_{inv} = inv(A)
return inverse of matrix A, the matrix A^{-1}

isfinite(x)
return 1 if x is a finite number, 0 otherwise

length(A)
return largest dimension of matrix A

val = load(`'-ascii'`, name)
load contents of text file *name* into *val*

log(x)
compute the natural logarithm, $\ln x$

log2(x)
compute the base-2 logarithm, $\log_2 x$

var *or* [var ix] = max(x)
find largest element in x, optionally with index *ix*

mesh(X,Y,Z)
plot 3-D mesh grid Z = f(X,Y)
use meshgrid to compute arrays X and Y

[X Y] = meshgrid(x,y) *or* meshgrid(x)
transforms domain specified by vectors (x,y) into arrays X and Y for use with 3-D plots
meshgrid(x) equals meshgrid(x,x)

var *or* [var ix] = min(x)
find smallest element in x, optionally with index *ix*

mod(x,y)
remainder of x/y, $x - \lfloor x/y \rfloor \cdot y$

nargin
number of arguments passed to the function

nargout
number of values the caller expects to receive

norm(x)
compute the 2-norm of x, `sqrt(sum(x.^2))`

str = num2str(x) *or* num2str(x,n)
convert input into text and store in *str*,
last form use a maximum precision of n digits

ones(N), ones(N,M)
return N×N or N×M matrix of ones

pause(secs), pause
pause execution *secs* seconds or until any key hit

pi, pi(N,M)
π as a number or a N×M-matrix

plot(x,y)
2-D plot of vector x versus vector y

prod(x)
product of elements in x, $\prod x_i$

pwd, string = pwd
print or return working directory as a string

rand, rand(N), rand(N,M)
return a random number on the open interval (0,1),
a N×N or N×M matrix of random numbers

realmax
return largest real (floating-point) number

realmin
return smallest real (floating-point) number

round(x)
round x towards nearest integer

save(`'-ascii'`, name, 'var', ...)
save variables *var, ...* on text file *name*

sign(x)
sign of x, -1 if negative, 0 if zero and 1 if positive

sin(x)
sine of x with argument in radians

sind(x)
sine of x with argument in degrees

size(A), size(A,n)
return all dimensions or n^th dimension of A

sort(X), sort(X,n), sort(X,n,mode)
sort X in ascending order, sort along n^th dimension
or sort with mode `'ascend'` or `'descend'`

sqrt(x)
compute square root of x, $\sqrt{x}$

x = str2double(string)
convert character string to (floating-point) number

A = str2num(string)
convert character string matrix to number
use str2double to convert a single number

y = sum(x)
compute sum of elements, $\sum x_i$

surf(X,Y,Z)
plot 3-D surface Z = f(X,Y)
use meshgrid to compute arrays X and Y

tan(x)
tangent of x with argument in radians

tand(x)
tangent of x with argument in degrees

tic, toc
set and check a wall-clock timer

true
return logical 1 (true)

type name
return the function or built-in matching *name*

version
return Matlab interpreter version string

who, who var, . . .
display all or specified variables *var,* . . .

whos, whos var, . . .
long form of who; more detailed listing

xlabel(str)
print x-axis label *str* onto 2D plot

ylabel(str)
print y-axis label *str* onto 2D plot

zeros(N)
return N×N or N×M matrix of zeroes