

MATLAB Quick Reference

HELP

help	lists 20 categories of help.
help <i>category</i>	lists functions in that category with a brief description.
help <i>function name</i>	lists information on that function. Use this for more complete information and additional options for everything listed on this handout!
lookfor <i>word</i>	search all help files for a particular word.

Some help categories

general	General purpose commands.
ops	Operators and special characters.
elmat	Elementary matrices and matrix manipulation.
elfun	Elementary math functions.
datafun	Data analysis and Fourier transforms.
polyfun	Interpolation and polynomials.
funfun	Function functions and ODE solvers.
graph2d	Two dimensional graphs.

SOME BASIC COMMANDS (Note command syntax is case-sensitive!)

matlab	issued from UNIX, this starts MATLAB.
quit	quits MATLAB, returning you to the operating system.
exit	same as quit.
who	lists all of the variables in your MATLAB workspace.
whos	list the variables and describes their matrix size.
clear	deletes all variables from active workspace. clear x deletes just the variable x from active workspace.
...	to continue a command to the next line, type the ellipsis (three successive periods) at the end of the line
save filename	saves the contents of workspace into filename.mat
save filename x y	saves the matrices x and y into the file titled filename.mat.
load filename	loads the contents of filename into current workspace; the file can be a binary (.mat) file or an ASCII file.
!	Use ! preceding a Unix command to execute the Unix command from MATLAB.

PLOTTING

plot(x,y)	creates a Cartesian plot of the vectors x & y.
plot(y)	creates a plot of y vs. the indices of the elements in the y-vector.
semilogx(x,y)	plots x vs y with a logarithmic x axis
semilogy(x,y)	plots x vs y with a logarithmic y axis
loglog(x,y)	plots x vs y with logarithmic axes
grid	creates a grid on the graphics plot.
title('text')	places a title at top of graphics plot.
xlabel('text')	writes 'text' beneath the x-axis of a plot.
ylabel('text')	writes 'text' beside the y-axis of a plot.
text(x,y,'text')	writes 'text' at the location (x,y) .
gtext('text')	writes text according to placement of mouse
print	sends the contents of graphics window to printer. print filename -dps writes the contents of current graphics to 'filename' in PostScript format.
hold on	maintains the current plot in the graphics window while subsequent plotting commands are executed. hold off turns off the 'hold on' option.
axis([xmin, xmax, ymin, ymax])	sets the axes for the current plot to the specified range.
figure	opens a new plot window; figure(n) opens a new or existing plot window numbered n.

Special Plot Types:

polar(theta,r)	creates a polar plot of the vectors r & θ where θ is in radians.
bar(x)	creates a bar graph of the vector x . (Note also the command stairs(y).)
bar(x,y)	creates a bar-graph of the elements of the vector y , locating the bars according to the vector elements of ' x '. (Note also the command stairs(x,y).)
hist(x)	creates a histogram. This differs from the bargraph in that frequency is plotted on the vertical axis.
mesh(z)	creates a surface in xyz space where z is a matrix of the values of the function $z(x,y)$. z can be interpreted to be the height of the surface above some xy reference plane.
contour(z)	draws a contour map in xy space of the function or surface z .

SPECIAL NUMBERS

i, j	square root of -1
pi	$\pi = 3.1415926\dots$

SPECIAL MATRICES

a:d:b	array of linearly spaced elements between a and b , spaced by d . $a:b$ defaults to $d=1$.
linspace(a,b,n)	array of n linearly spaced elements between a and b , inclusive
logspace(a,b,n)	array of n logarithmically spaced elements between 10^a and 10^b , inclusive.
eye(n)	creates the $n \times n$ identity matrix (1's on the diagonal).
ones(n)	creates a $n \times n$ (square) matrix whose elements are 1's
ones(m,n)	creates a $m \times n$ (m-row, n-column) matrix whose elements are 1's.
ones(A)	creates a matrix of 1's the same size as an existing matrix, A .
zeros(n)	creates an $n \times n$ (square) matrix whose elements are zero.
zeros(m,n)	creates a $m \times n$ (m-row, n-column) matrix of zeros.
zeros(A)	creates a matrix of 0's the same size as an existing matrix, A .

MATRIX MANIPULATION

x(m:n)	returns the subvector $[x(m), x(m+1), \dots, x(n)]$.
x(end)	returns the last element of x ; $x(\text{end}-1)$ is the next-to-last element, etc.
x(:)	convert x to a column vector
A(m:n, p:q)	returns the submatrix of A consisting of rows m through n of columns p through q .
A(m:n, :)	returns the submatrix of A consisting of rows m through n of <i>all</i> columns.
A(:, p:q)	returns the submatrix of A consisting of columns p through q of <i>all</i> rows.
A(:, :)	same as A
length(x)	returns the number elements in vector x .
size(A)	returns a vector $[m, n]$ containing the size m (rows) and n (columns) of matrix A .
fliplr(x)	Reverse the order of the columns in a matrix. If x is a row vector, flips it end-for-end.
flipud(x)	Reverse the order of the rows in a matrix. If x is a column vector, flips it end-for-end.
reshape(A,m,n)	reshapes the matrix A into an $m \times n$ matrix from element (1,1) working column-wise.
A', A.'	transpose matrix A (interchange rows and columns). For real-valued matrices, $'$ and $'$ are equivalent, but with complex elements, $'$ alone gives the conjugate transpose, whereas $'$ gives the simple transpose.
find(x)	returns the indices of the nonzero elements of vector x . $\text{find}(x>1)$ returns the indices of those elements of x which are greater than 1. $x(\text{find}(x>1))$ returns the subvector of x whose elements are all greater than 1.

PRINTING

To print a figure, select "Print" from the file menu in the figure window.

To print text from UNIX/linux: Save the file. Then type 'lpr -Plw225-1 filename.m' in the UNIX shell (not the MATLAB command window). The -P... can be skipped to use the default printer. To print the command window, copy (by highlighting) and paste (middle mouse button) into the editor.

To print text from a PC: Select "Print" from the file menu in the text editor or the command window.

ARITHMETIC OPERATIONS*Ordinary matrix (and scalar) arithmetic*

+	addition
-	subtraction
*	multiplication; for matrices, $A*B$ is not usually equal to $B*A$.
/	right division (a/b means $a \div b$)
\	left division ($a \backslash b$ means $b \div a$)
^	exponentiation

The sum or product of a scalar and a matrix is a matrix in which every element of the matrix has been added or multiplied by the scalar.

The precedence or order of the calculations included in a single line of code follows the usual order:

- 1 parentheses
- 2 exponentiation, left to right
- 3 multiplication and division, left right
- 4 addition and subtraction, left right

Array operations

Array or element-by-element operations are executed when the operator is preceded by a '.' (period). Thus

$a .* b$	multiplies each element of a by the respective element of b
$a ./ b$	divides each element of a by the respective element of b
$a .^ b$	raises each element of a to the respective element of b

Solution to simultaneous equations

$\text{inv}(A)$ returns the inverse of the (square) matrix A . $A*\text{inv}(A)$ results in an identity matrix. If $Ax=b$ is a matrix equation and A is the coefficient matrix, the solution x is $x=\text{inv}(A)*b$. You can also type $x=A \backslash b$ to do the same thing.

COMPARISON OPERATORS

$x == y$	returns 1 if $x=y$, 0 otherwise. If x and y are vectors, $x==y$ returns a vector with 1s where $x_i = y_i$ and 0s elsewhere.
$\sim =$	not equal
$<, >$	less than, greater than
$<=, >=$	less than or equal, greater than or equal

BOOLEAN OPERATORS

A variable x is true if it is nonzero, else it is false.

$x \& y$	Logical AND of vectors x and y . Returns a vector with 1s where x_i and y_i are both nonzero, and 0s elsewhere.
$x y$	Logical OR. Returns a vector with 1s where either x_i or y_i is nonzero, and 0s where they are both zero.
$\sim x$	Logical NOT. Returns a vector with 1s where $x_i=0$ and 0s elsewhere.
$\text{xor}(x,y)$	Logical EXCLUSIVE OR. Returns a vector with 1s where either x_i or y_i is nonzero (but not both), and 0s where they are both zero or both one.
$\text{any}(x)$	True if any element of vector x is nonzero
all	True if all elements of vector x are nonzero

CONDITIONAL EXECUTION

```
if <expression>
    <statements>
elseif <expression>
    <statements>
else
    <statements>
end
```

POLYNOMIALS

Polynomials can be represented by vectors of coefficients, e.g., $a_0x^2 + a_1x + a_2 \leftrightarrow [a_0, a_1, a_2]$. The following functions work with that representation.

roots(p) computes the roots of the polynomial whose coefficients are in the vector p
poly(r) computes the coefficients of the polynomial whose roots are in the vector r
polyval(p,x) evaluates the polynomial p at values x

MATHEMATICAL FUNCTIONS

cos(x) returns a vector containing the cosines of the elements of x. Type “help elfun” to see all elementary mathematical functions (sin, tan, log, log10, exp, abs, real, imag, angle, ...)
atan2(y,x) computes the inverse tangent of y./x, placing the angle in the proper quadrant.

DATA ANALYSIS COMMANDS

max(x) returns the maximum value of the elements in a vector or if x is a matrix, returns a row vector whose elements are the maximum values from each respective column of the matrix.
min(x) returns the minimum of x (see max(x) for details).
mean(x) returns the mean value of the elements of a vector or if x is a matrix, returns a row vector whose elements are the mean value of the elements from each column of the matrix.
median(x) same as mean(x), only returns the median value.
polyfit(x, y, n) finds the coefficients of a polynomial $p(x)$ of degree n that fits the data points (x_i, y_i) in vectors x and y, $p(x_i) \sim y_i$, in a least-squares sense. Use polyfit(x,y,1) to fit a line, polyfit(x,y,2) to fit a parabola, etc.
std(x) same as mean(x), only returns the standard deviation.
sum(x) returns the sum of the elements of a vector or if x is a matrix, returns the sum of the elements from each respective column of the matrix.

FUNCTIONS

To create a function funname(), create a file funname.m The first line of the file must be:

```
function [output1,output2,...] = funname(input1,input2,...)
```

Variables within the function are local variables.

SOLVING DIFFERENTIAL EQUATIONS

Use ode45, as described in detail in another handout.

This handout compiled by Charlie Sullivan 6/99, largely following the US Naval Academy Mechanical Engineering Department's guide, <http://eng.usna.navy.mil/~mecheng/DESIGN/CAD/MATLAB/matlab/matlab.html>. Rev 1/01 by E. Hansen, 6/02 C. Sullivan.