ENG101 Reference Card for MATLAB, Revision 11/20/2001

MATLAB Workspace and Variables

The basic variable type in MATLAB is a two-dimensional array of doubles (64-bit representation).
A *scalar* is a 1×1 array.
A *row vector* of length N is a $1 \times N$ array.
A *column vector* of length M is an $M \times 1$ array.
A *matrix* of dimensions M rows and M columns is an $M \times N$ array.

Variable name conventions

MATLAB is case sensitive
Starts with A-Z, a-z
Up to 31 letters, digits and underscores

Default MATLAB variables

ans	result of last unassigned calculation
eps	smallest number that can be added to 1.0 and still be different
flops	count of floating point operations
Inf	infinity *e.g.* 1/0 = Inf
NaN	Not a Number *e.g.* 0/0 = NaN
pi	Value of Pi (3.1415...)
realmax	Largest real number MATLAB can represent
realmin	Smallest real number MATLAB can represent

User defined variables

clear	clear all or selected variables (or functions) from the current workspace
length	length of a vector or maximum dimension of an array
size	display dimensions of a particular array
who	display current workspace variable names
whos	display current workspace variable names, types and associated sizes

Command/Figure Window control

clc	clear the command window
clf	clear the figure window
figure	start a new figure window
figure(N)	Make figure with index N active. If N is an integer and figure(N) does not exist, create it
close	close current figure window
close(N)	close figure with index N

Useful Workspace functions

help	obtain help generally or for a specific function
lookfor	obtain one-line help if it exists
more	toggles pagination, useful for longs "helps"
load	read variables in from a disk file
save	save all or selected variables to a disk file

MATLAB path and environment

cd	change to s specific directory
dir	list files in the current directory
path	display or modify the function search path

MATLAB punctuation

.	decimal point 325/100, 3.25 and .325e1 are all the same
...	three or more decimal points at the end of a line cause the following line to be a continuation
,	comma is used to separate matrix subscripts and arguments to functions, also used to separate statements in multi-statement lines
;	used inside brackets to indicate the ends of the rows of a matrix, also used after an expression or statement to suppress printing
%	begins comments
'	Quote. 'ANY TEXT' is a vector whose components are the ASCII codes for the characters. A quote within the text is indicated by two quotes. For example: 'Don''t forget.'

Assignment

Individual elements with a row can be delimited by a comma or a space.

Explicit assignment using ;'s to end rows

```
a = [1,2,3;4,5,6;7,8,9]
```

Explicit assignment using "newline" to end rows

```
a = [1,2,3
4,5,6
7,8,9]
```

Explicit assignment using continuation lines

```
b = [1 2 3 4 5 6 ...
7 8 9 10]
```

Useful ways to initialize variables

linspace(a,b,N)	Linearly spaced intervals between a and b (inclusive) comprised of N points
zeros(m,n)	An m by n array of zeroes
zeros(n)	An n by n array of zeroes
ones(m,n)	An m by n array of ones
ones(n)	An n by n array of ones
eye(m,n)	An m by n array with ones on the diagonal
eye(n)	An n by n identity matrix
ones(n)	An n by n array of ones
rand(m,n)	An m by n array of random numbers
rand(n)	An n by n array of random numbers

Vector/Matrix operators

These are Vector and Matrix operators

+	addition
−	subtraction
*	multiplication
/	left division
\	right division
^	exponentiation
'	transpose

Point-by-point operators

These operate on matrix elements in point-wise fashion

.*	point-wise multiplication
./	point-wise left division
.\	point-wise right division
.^	point-wise exponentiation

Logical operators

<	less than
<=	less than or equal
>	greater than
>=	greater than or equal
==	equal
~=	not equal
&	logical AND
|	logical OR
~	logical NOT

List generation/variable indexing

i:k:l	List generation syntax = 1stValue:Stride:LatValue
v(1)	1st element of vector v
v(end)	Last element of vector v
v(1:2:9)	1st, 3rd, 5th, 7th, 9th elements of vector v
v(2:3:9)	2nd, 5th, 8th elements of vector v
a(2,3)	2'nd row, 3'rd column of matrix a

```
a(:,3)        all elements in column 3
a(1,:)        all elements in row 1
a(1:2:end,:)  all odd rows of matrix a
a(1:2,2:4)    sub-matrix of rows 1 and 2, columns 2 through 4
a(1,end)      last element in 1'st row
```

Script M-files

Sequences of MATLAB commands can be stored in text files with the extension .m. The commands can be executed with typing the name of the files (without the extension) or through the file management tools provided by the Command Window menu.

Useful M-file functions

Function M-files

Function definition

Define a separate file called functionName.m with the following form:

```
function [out1,...,outN] = functionName(in1,...inM)
%FUNCTIONNAME A brief one line description (optional)
% .
% .
% More description (optional)
% .
% .
% .
% First executable statement follows this line
.
.
.
% Valid executable MATLAB statements and comments
.
.
.
% Last line in functionName
```

Function call

The function call is made with the following statement:

```
[out1,out2,...,outN] = functionName(in1,in2,...inM)
```

Useful M-file functions

```
disp       display a string
fprintf    write data to screen of file
echo       toggle command echo
error      display message and abort
input      prompt for input
keyboard   transfer control to keyboard
pause      wait for time or user response
return     return to caller
warning    display warning messages
```

Performance monitoring

```
tic,toc    stopwatch timer functions
flops      counts floating point operations
```

Formatting

```
format short    Scaled fixed point format with 5 digits
format long     Scaled fixed point format with 15 digits
format compact  Suppress extra line-feeds
format loose    Puts extra line-feeds in the output
```

Program Flow control

for loops

```
for k = vectorOrColumnList
    % MATLAB statements
end
```

while loops

```
while logicalExpression
    % MATLAB statements
end
```

if/elseif/else construct

```
if logicalExpression1 % Mandatory
    % MATLAB statements
elseif logicalExpression2 % Optional
    % MATLAB statements
elseif logicalExpression3 % Optional
.
.
.
elseif logicalExpressionN % Optional
    % MATLAB statements
else % Optional
    % MATLAB statements
end % Mandatory
```

Plotting

```
contour    Contour plot on a plane
contour3   3-D contour plot with displayed depth
mesh       3-D mesh surface
meshc      Combination mesh/contour plot
meshz      3-D mesh with curtain
pcolor     Pseudocolor (checkerboard) plot
plot       Basic 2D plots
plot3      Plot lines and points in 3-D space
surf       3-D colored surface
surfc      Combination surf/contour plot
surfl      3-D shaded surface with lighting
```

Plotting annotation

```
clabel     Contour plot elevation labels
colorbar   Display color bar (color scale
legend     Graph legend
title      Graph title
xlabel     X-axis label
ylabel     Y-axis label
```

Additional plotting functions

```
box        Toggle the box display
colormap   color look-up table
grid       Toggle the grid state
hold       Control multiple plots on a single figure
shading    Color shading mode e.g. flat, interp
subplot    control multiple plots in one window
zoom       Enable mouse-based zooming
```