MATLAB Summary (taken from http://www.math.ufl.edu/help/matlab-tutorial/matlab-tutorial.html)

MATLAB (matrix algebra)

Online help is available from the Matlab prompt (a double arrow)

>> help [a long list of help topics follows]

or for specific commands:

>> help fft [a help message on the fft function follows].

MATLAB Tutorial

Building Matrices

Matlab has many types of matrices which are built into the system. A 7 by 7 matrix with random entries is produced by typing

>> rand(7)

You can generate random matrices of other sizes:

>> rand(2,5)
>> help rand

Some of the standard matrices from linear algebra:

>> eye(6)
>> zeros(4,7)
>> ones(5)

Build matrices with any entries that you want.

>> [1 2 3 5 7 9]
>> [1, 2, 3; 4, 5, 6; 7, 8, 9]
>> [1 2 <RETURN> 3 4 <RETURN> 5 6]

Variables

Matlab has built-in variables like pi, eps, and ans. You can learn their values from the Matlab interpreter.

>> pi
>> help pi

At any time you want to know the active variables you can use who:

>> who
>> help who

The variable ans will keep track of the last output which was not assigned to another variable.

>> [1, 2, 3; 4, 5, 6; 7, 8, 9]
>> ans
>> x = ans
>> x

Since you have created a new variable, x, it should appear as an active variable.

>> who

To remove a variable, try this:

>> clear x
>> x
>> who [x has now disappeared from the list of active variables]

Functions

>> a = [1 2; 5 7]

Take the transpose of a:

>> a'

Note that if the matrix A has complex numbers as entries then the Matlab function taking A to A' will compute the transpose of the conjugate of A rather than the transpose of A.

Other arithmetic operations are easy to perform.

```
>> 3*a
>> -a
>> a+(-a)
>> b = max(a)
>> max(b)
```

Matlab has a convention in which a dot in front of an operation usually changes the operation. In the case of multiplication, a.*b will perform entry-by-entry multiplication instead of the usual matrix multiplication.

```
>> a = [1 2; 5 7]
>> b = [1 2; 1 3]
>> a.*b  (there is a dot there!)
>> x = 5
>> x^2
>> a*a
>> a^2
>> a.^2   (another dot)
>> a
>> diag(a)
>> diag(diag(a))
>> c=rand(4,5)
>> size(c)
>> [m,n] = size(c)
>> m
>> d=.5-c
```

There are many functions which we apply to scalars which Matlab can apply to both scalars and matrices.

```
>> sin(d)
>> exp(d)
>> log(d)
>> abs(d)
```

Matlab has functions to round floating point numbers to integers. These are round, fix, ceil, and floor. The next few examples work through this set of commands and a couple more arithmetic operations.

```
>> f=[-.5 .1 .5]
>> round(f)
>> ceil(f)
>> floor(f)
>> sum(f)
>> prod(f)
```

Relations and Logical Operations

In this section you should think of 1 as "true" and 0 as "false." The notations &, |, ~ stand for "and,""or," and "not," respectively. The notation == is a check for equality.

```
>> a=[1 0 1 0]
>> b=[1  1  0  0]
>> a==b
>> a<=b
>> ~a
>> a&b
>> a & ~a
>> a | b
>> a | ~a
```

Colon Notation

Matlab offers some powerful methods for creating arrays and for taking them apart.

```
>> x=-2:1
>> length(x)
>> -2:.5:1
>> -2:.2:1
>> a= [1 2 3 4 5; 4 5 6 7 8; 9 8 7 6 5; 3 4 5 6 7; 6 5 4 3 2]
>> a(2,3)
```

Now we will use the colon notation to select a column of a.

```
>> a(2,:)
>> a(:,3)
>> a
>> a(2:4,:)
>> a(:,3:5)
>> a(2:4,3:5)
>> a(1:2:5,:)
```

The MATLAB display only shows 5 digits in the default mode. The command

```
>> format long
```

will switch to display all 16 digits and

```
>> format short
```

will return to the shorter display.

It is not always necessary for MATLAB to display the results of a command to the screen. If you do not want the matrix A displayed, put a semicolon

after it, A;. When MATLAB is ready to proceed, the prompt >> will appear. Try this on a matrix right now.

Saving variables

You can save variables and matrices in files to be used in later sessions. You can save these values in a file by typing

>> save filename

This creates a file filename.mat

which contains the values of the variables from your session. If you do not want to save all variables there are two options. One is to clear the variables off with the command
>> clear a b c
which will remove the variables a,b,c. The other option is to use the command
>> save x y z
which will save the variables x,y,z in the file filename.mat. The variables can be reloaded in a future session by typing
>> load filename

When you are ready to print out the results of a session, you can store the results in a file and print the file from the operating system using the "print" command appropriate for your operating system. The file is created using the command

Programming in MATLAB

MATLAB is also a programming language. By creating a file with the extension .m you can easily write and run programs. Open an m-file (File, New, m-file)\

Type the following code in the m-file the value of x. Here is a short program illustrating the use of assignment.

```
function r=mod(a,d)

% r=mod(a,d). If a and d are integers, then
% r is the integer remainder of a after
% division by d. If a and b are integer matrices,
% then r is the matrix of remainders after division
% by corresponding entries. Compare with REM.

r=a-d.*floor(a./d);
```

Now save it as mod.m . In the command line in Matlab assign some integer values for a and d. Run

>> mod(a,d)

This should run just like any built-in MATLAB function. Type
help mod
This should produce the five lines of comments which follow the % signs. The % signs generally indicate that what follows on that line is a comment which will be ignored when the program is being executed.

The variable r is being assigned the value of a-d.*floor(a./d); The operations on the right hand side of the assignment have the meaning which you have just been practicing (the / is division) with the "." meaning the entry-wise operation as opposed to a matrix operation. Finally, the ";" prevents printing the answer to the screen before the end of execution.

If statement

if <condition>, <program> end

The condition is a MATLAB function usually, but not necessarily with values 0 or 1 and the entire construction allows the execution of the program just in case the value of condition is not 0. If that value is 0, the control moves on to the next program construction. You should keep in mind that MATLAB regards a==b and a<=b as functions with values 0 or 1.

Frequently, this construction is elaborated with

if <condition1>, <program1> else <program2> end
In this case if condition is 0, then program2 is executed.

Another variation is

```
if <condition1>, <program1>
elseif <condition2>, <program2>
end
```

Now if condition1 is not 0, then program1 is executed, if condition1 is 0 and if condition2 is not 0, then program2 is executed, and otherwise control is passed on to the next construction. Here is a short program to illustrate branching.

```
function b=even(n)

% b=even(n). If n is an even integer, then b=1
% otherwise, b=0.

if mod(n,2)==0
  b=1;
else b=0;
end;
```

For Loops

A for loop is a construction of the form

```
for i=1:n, <program>, end
```

The <program> is repeated until I is equal to n. After each iteration I is incremented. Write a function for matrix addition.

```
function c=add(a,b)

% c=add(a,b). This is the function which adds
% the matrices a and b. It duplicates the MATLAB
% function a+b.

[m,n]=size(a);
[k,l]=size(b);
if m~=k | n~=l,
  r='ERROR using add: matrices are not the same
size';
  return,
end
c=zeros(m,n);
for i=1:m,
  for j=1:n,
    c(i,j)=a(i,j)+b(i,j);
  end
end
```

While Loops

A while loop is a construction of the form

```
while <condition>, <program>, end
```

where condition is a MATLAB function. The program will execute successively as long as condition is not 0. DANGER!!! Some times while loops never end – check the end condition

```
function l=twolog(n)

% l=twolog(n). l is the floor of the base 2
% logarithm of n.

l=0;
m=2;
while m<=n
  l=l+1;
  m=2*m;
end
```

Scripts

A script is an m-file without the function declaration at the top. A script behaves differently. When you type who you are given a list of the variables which are in force during the current session. Suppose that x is one of those variables. When you write a program using a function file and you use the variable x inside the program, the program will not use the value of x from your session (unless x was one of the input values in the function), rather x will have the value appropriate to the program. Furthermore, unless you declare a new value for x, the program will not change the value of x from the session. This is very convenient since it means that you do not have to worry too much about the session variables while your program is running. All this has happened because of the function declaration. If you do not make that function declaration, then the variables in your session can be altered. Sometimes this is quite useful, but I usually recommend that you use function files.

Some MATLAB built-in functions

This is a list of functions available in Matlab as of 1984, which should be taken as a quick reminder of the most basic tools available. See the Matlab help screens and excerpts from those screens reprinted in section Some MATLAB function descriptions. In any case, your version of Matlab may vary slightly.

intro	<	chol	end	function	lu	quit	sprintf
help	>	clc	eps	global	macro	qz	sqrt
demo	=	clear	error	grid	magic	rand	startup
[	&	clg	eval	hess	max	rcond	string
]	|	clock	exist	hold	memory	real	subplot
(	~	conj	exit	home	mesh	relop	sum
)	abs	contour	exp	ident	meta	rem	svd
.	all	cos	expm	if	min	return	tan
,	ans	cumprod	eye	imag	nan	round	text
;	any	cumsum	feval	inf	nargin	save	title
%	acos	delete	fft	input	norm	schur	type
!	asin	det	filter	inv	ones	script	what
:	atan	diag	find	isnan	pack	semilogx	while
'	atan2	diary	finite	keyboard	pause	semilogy	who
+	axis	dir	fix	load	pi	setstr	xlabel
-	balance	disp	floor	log	plot	shg	ylabel
*	break	echo	flops	loglog	polar	sign	zeros
\	casesen	eig	for	logop	prod	sin	
/	ceil	else	format	ltifr	prtsc	size	
^	chdir	elseif	fprintf	ltitr	qr	sort	

acosh	demo	hankel	membrane	print	table1
angle	demolist	hds	menu	quad	table2
asinh	dft	hilb	meshdemo	quaddemo	tanh
atanh	diff	hist	meshdom	quadstep	tek
bar	eigmovie	histogram	mkpp	rank	tek4100
bench	ergo	hp2647	movies	rat	terminal
bessel	etime	humps	nademo	ratmovie	toeplitz
bessela	expm1	idft	nelder	readme	trace
besselh	expm2	ieee	neldstep	residue	translate
besseln	expm3	ifft	nnls	retro	tril
blanks	feval	ifft2	null	roots	triu
cdf2rdf	fft2	info	num2str	rot90	unmkpp
census	fftshift	inquire	ode23	rratref	vdpol
citoh	fitdemo	int2str	ode45	rratrefmovie	versa

cla	fitfun	invhilb	odedemo	rref	vt100
compan	flipx	isempty	orth	rsf2csf	vt240
computer	flipy	kron	pinv	sc2dc	why
cond	funm	length	plotdemo	sg100	wow
conv	gallery	log10	poly	sg200	xterm
conv2	gamma	logm	polyfit	sinh	zerodemo
corr	getenv	logspace	polyline	spline	zeroin
cosh	ginput	matdemo	polymark	sqrtm	
ctheorem	gpp	matlab	polyval	square	
dc2sc	graphon	mean	polyvalm	std	
deconv	hadamard	median	ppval	sun	

Elementary matrices and matrix manipulation.

Elementary matrices.

zeros	Zeros matrix.
ones	Ones matrix.
eye	Identity matrix.
rand	Uniformly distributed random numbers.
randn	Normally distributed random numbers.
linspace	Linearly spaced vector.
logspace	Logarithmically spaced vector.
meshgrid	X and Y arrays for 3
:	Regularly spaced vector.

Special variables and constants.

ans	Most recent answer.
eps	Floating point relative accuracy.
pi	3.1415926535897....
i, j	Imaginary unit.
inf	Infinity.
NaN	Not

Elementary math functions.

Trigonometric.

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

Exponential.

exp	Exponential.
log	Natural logarithm.
log10	Common logarithm.
sqrt	Square root.

Complex.

abs	Absolute value.
angle	Phase angle.
conj	Complex conjugate.
imag	Complex imaginary part.

real	Complex real part.

Numeric.

fix	Round towards zero.
floor	Round towards minus infinity.
ceil	Round towards plus infinity.
round	Round towards nearest integer.
rem	Remainder after division.
sign	Signum function.

Matrix functions

Matrix analysis.

cond	Matrix condition number.
norm	Matrix or vector norm.
rcond	LINPACK reciprocal condition estimator.
rank	Number of linearly independent rows or columns.
det	Determinant.
trace	Sum of diagonal elements.
null	Null space.
orth	Orthogonalization.
rref	Reduced row echelon form.
inv	Matrix inverse.

Eigenvalues and singular values.

eig	Eigenvalues and eigenvectors.
poly	Characteristic polynomial.

Managing variables and the workspace.

who	List current variables.
whos	List current variables, long form.
load	Retrieve variables from disk.
save	Save workspace variables to disk.
clear	Clear variables and functions from memory.
size	Size of matrix.
length	Length of vector.

Functions

ode23	Solve differential equations, low order method.
ode23p	Solve and plot solutions.
ode45	Solve differential equations, high order method.
quad	Numerically evaluate integral, low order method.
quad8	Numerically evaluate integral, high order method.
fmin	Minimize function of one variable.
fmins	Minimize function of several variables.
fzero	Find zero of function of one variable.
fplot	Plot function.

Operators and special characters.

Char	Name
+	Plus
-	Minus
*	Matrix multiplication
.*	Array multiplication
^	Matrix power
.^	Array power
\	Backslash or left division
/	Slash or right division
./	Array division
kron	Kronecker tensor product
:	Colon
( )	Parentheses
[ ]	Brackets
.	Decimal point
..	Parent directory
...	Continuation
,	Comma
;	Semicolon
%	Comment
!	Exclamation point
'	Transpose and quote
=	Assignment
==	Equality
<,>	Relational operators
&	Logical AND
|	Logical OR

~	Logical NOT
xor	Logical EXCLUSIVE OR

MATLAB programming

Control flow.

if	Conditionally execute statements.
else	Used with IF.
elseif	Used with IF.
end	Terminate the scope of FOR, WHILE and IF statements.
for	Repeat statements a specific number of times.
while	Repeat statements an indefinite number of times.
break	Terminate execution of loop.
return	Return to invoking function.
error	Display message and abort function.

Two dimensional graphics.

plot	Linear plot.
loglog	Log-log scale
semilogx	semi-log scale plot
semilogy	semi-log scale plot
polar	Polar coordinate plot.
bar	Bar graph.
stairs	Stairstep plot.
errorbar	Error bar plot.
hist	Histogram plot.
rose	Angle histogram plot.
compass	Compass plot.
feather	Feather plot.
fplot	Plot function.

Graph annotation.

title	Graph title.
xlabel	X-axis label
ylabel	Y-axis label
text	Text annotation.
gtext	Mouse placement of text.
grid	Grid lines.

3D graphs

plot3

surf

mesh