Matlab Quick Reference

Run matlab

`matlab`	(evoke matlab)	`quit` *or* `exit`	(exit matlab)

Arithmetic

`+, -, *, /, \, ^`	(e.g. `a*x^2 + 2*x - 3/5` $(ax^2 + 2x - 3/5)$)
`.*, ./, .\, .^, .'`	(elementwise multiply, left and right divide, exponentiation, transpose for matrices)

Mathematical Functions

`sqrt(`x`)`	$(\sqrt{x})$	`exp(`x`)`	(e^x)	`log(`x`)`	$(\ln x)$	`log10(x)`	$(\log_{10} x)$
`sin(`x`)`	$(\sin x)$	`cos(`x`)`	$(\cos x)$	`tan(`x`)`	$(\tan x)$	`asin(`x`)`	$(\sin^{-1} x)$
`acos(`x`)`	$(\cos^{-1} x)$	`atan(`x`)`	$(\tan^{-1} x)$	`sinh(`x`)`	$(\sinh x)$	`cosh(`x`)`	$(\cosh x)$
`tanh(`x`)`	$(\tanh x)$	`asinh(`x`)`	$(\sinh^{-1} x)$	`factorial(`n`)`	$(n!)$	`abs(`x`)`	$(\lvert x \rvert)$
`rem(`n,m`)`	(remainder)	`max(`v`)`	(max)	`min(`v`)`	(min)	`sign(`x`)`	$(x/\lvert x \rvert)$
`real(`z`)`	(real part)	`imag(`z`)`	(imag part)	`conj(`z`)`	$(\bar{z})$	`angle(`z`)`	(argument)
`floor(`x`)`	$(\lfloor x \rfloor)$	`ceil(`x`)`	$(\lceil x \rceil)$	`fix(`x`)`	(round to 0)	`besselj(`n,z`)`	$(J_n(z))$
`erf(`x`)`	$(\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt)$	`gamma(`z`)`	(Euler $\Gamma(z)$)	`ellipj(`u,m`)`	(elliptic)	`rand`	(number)

Mathematical Constants

`pi`	$(\pi \approx 3.14159)$	`i`	$(i = \sqrt{-1})$	`inf`	(∞)	`NaN`	(Not-a-Number)

Data Objects

`123`	(integer)	`1.0`	(real)	`2 + 8i`	(complex)	*'text'*	(string)
`[`$a, b, \ldots$ `]`			(array)	`[1,2;3,4;5,6]`			(3×2 matrix)
$x.f$			(f field of structure x)	`C{2,2}`			(2×2 cell)
`A(:,`j`)`			(vector of j-th column of A)	`A(`m`:`n`,`k`:`l`)`			(submatrix row m to n, column k to l)
`A(`i`:`s`:`j`)`							(array with elements A_i, A_{i+s}, A_{i+2s}, until A_{i+ks} such that $i + ks \le j$)
a`:`s`:`b							(the array $(a, a+s, a+2s, \ldots, a+ks)$, where k is largest such that $a + ks \le b$)

Defining Variables and Functions

x `=` *value*	(assign a value to variable x)	`clear` x	(remove x from memory)
`function [out]= f(x);`	(define a function f)	`global` x	(global variable in function)

Matrix and Linear Algebra

`eye(`n`)`	(identity matrix)	`ones(`n`)`	(all 1)	`zeros(`n`,`m`)`	(0 matrix)
`inv(`m`)`	(inverse of matrix)	m`'`	(complex conjugate transpose)	`det(`m`)`	(determinant)
`pinv(`m`)`	(pseudoinverse)	`cond(`m`)`	(condition number)	`trace(`m`)`	(sum of diagonal)
m`^`n	(matrix power)	`expm(`m`)`	(e^m)	$m\backslash b$	(solve $mx = b$)
`null(m)`	(null space)	`sqrtm(`m`)`	$(m^{1/2})$	b/m	(solve $xm = b$)
`eig(`m`)`	(eigensystem)	`svd(`m`)`	(singular value)	`lu(`m`)`	(LU factor)
`sparse(`m`)`	(create sparse matrix)	`full(`s`)`	(sparse to full)	`eigs(`s`)`	(eigenvalues)

Numerical Math

`roots(`p`)`	(roots of polynomial)	`polyfit(`x`,`y`,`n`)`	(fit data)	`interp1`	(interpolation)
`quad('f',0,1)`	(integrate, 0 to 1)	`ode45('f',[`t_0`, `t_1`], `x_0`)`	(ODE)	`fft(`v`)`	(fast Fourier transform)

Graphics

`plot(`x`,`y`)` (2D plot x vs y)
`plot3(`x`, `y`, `z`)` (three-dimensional plot of curve)
`surf(`x`, `y`, `z`)` (three-dimensional surface)
`loglog(`x`,`y`)` (2D plot $\log x$ vs $\log y$)
`spy(`s`)` (plot a sparse matrix)
`gplot(`B`, `v`)` (plot a (graph-theoretical) graph)

Plot options:

`axis(`s`)`	(aspect ratio)	`xlabel(`s`)`	(add labels)	`title(`s`)`	(titel of plot)

Programming

`function [xout, yout] = funcname[xin, yin];`	(function header)
`% ...`	(start a comment line)
`return`	(force an early return in function)
`feval(`*s*`)`	(evaluate function specified by string s)
`for` i `= 1:100`	(for loop with i runs from 1 to 100)
statements	
`end`	(for, while, switch, if, all end with 'end')
`while i < 10,` *body*`, end`	(repeat *body* while condition is true)
`if i>5`	(if branch)
$statement_1$	
`elseif i<0`	(optional)
$statement_2$	
`else`	(optional)
$statement_3$	
`end`	(end conditional)
`switch flag`	(multiple choice, unlike C, no 'fall through')
`case v1`	(case of flag equal v1)
$\texttt{statement}_1$	
`case v2`	(case of flag equal v2)
$\texttt{statement}_2$	
`...`	
`otherwise`	(default case)
statements	
`end`	(end cases)
`break`	(terminate the innermost loop)
`error(`*s*`)`	(display an error message)
`n = input(`*s*`)`	(prompt user with string s, read input to n)
`fopen`, `fclose`, `fprintf`, `fscanf`, `fgets`, *etc*	(C-like I/O functions)

Relational and Logical Operators

`==`	(equal)	`~=`	(unequal)	`<`	(less than)	`<=`	(less or equal)
p `|` q	(logical or)	p `&` q	(logical and)	`>`	(greater than)	`>=`	(greater or equal)
`~`p	(logical not)	`xor(`p,q`)`	(exclusive or)	`all(`v`)`	(all are true)	`any(`v`)`	(1 or more true)

Input Editing and Help

arrow key	(edit entered command)	`help` f	(help on function f)	`whos`	(workspace arrays)
`Ctrl-C`	(interrupt execution)	`!`	(shell escape)		

References

MATLAB, the language of technical computing, the MathWorks Inc

Getting Started with MATLAB: Version 6: A quick intro for scientists and Engineers, R Pratap, Oxford, 2002