Matlab Quick Reference

Run Matlab

`matlab` (evoke Matlab frontend) `quit` or `exit` (exit Matlab)

Command

`what` (listing M-files)	`dir` (list all files)	`ls` (i.e. dir)	`type t` (display t)
`delete t` (delete t)	`cd p` (change to p)	`cddir p` (i.e. cd p)	`cd` (show folder)
`...;` (run ... no display)	`pwd` (i.e. cd)	`which t` (display to t)	`help t` (help for t)
`lookfor t` (help for t)	`clear` (clear variables)	`who` (list variables)	`demo` (run demos)

Arithmetic Operations and Scalar Multiplication

`+`, `-`, `*`, `/` or `\`, `^` (power), e.g. `a*x^2 + 2*x - 3/5` $(ax^2 + 2x - 3/5$

Array Operations

`+`, `-`, `.*`, `./` or `.\`, `.^` (power), e.g. `x.*y.^2-2*z+1` $(xy^2 - 2z + 1 = (x_i y_i^2 - 2z_i + 1)_{i=1}^N)$

Number Display Formats

`long` (5.83333333333334)	`short e` $(3.5833e+01)$	`long e` $(5.83...334e+01)$	`hex` (4041eaaaaaaaab)
`bank` (35.83)	`+` $(+5.3833)$	`rat` $(215/6)$	`short` (35.833)

Elementary Functions

`abs(`x`)` $(\|x\|)$	`acos(`x`)` $(\cos^{-1} x)$	`acosh(`x`)` $(\cosh^{-1} x)$	`angle(`x`)` (argument)
`asin(`x`)` $(\sin^{-1} x)$	`asinh(`x`)` $(\sinh^{-1} x)$	`atan(`x`)` $(\tan^{-1} x)$	`atan2(`x, y`)` $(\tan^{-1} \frac{x}{y})$
`atanh(`x`)` $(\tanh^{-1} x)$	`ceil(`x`)` (round $\to \infty$)	`conj(`x`)` $(\bar{x})$	`cos(`x`)` $(\cos x)$
`cosh(`x`)` $(\cos^{-1} x)$	`exp(`x`)` (e^x)	`fix(`x`)` (round $\to 0$)	`floor(`x`)` (round $\to -\infty$)
`imag(`x`)` (imaginary part)	`log(`x`)` $(\log x)$	`log10(`x`)` $(\log_{10} x)$	`real(`x`)` (real part)
`rem(`x, y`)` (rem. ofx/y)	`round(`x`)` (round $\to$ int)	`sign(`x`)` (signum function)	`sin(`x`)` $(\sin x)$
`sinh(`x`)` $(\sin^{-1} x)$	`sqrt(`x`)` $(\sqrt{x})$	`tan(`x`)` $(\tan x)$	`tanh(`x`)` $(\tan^{-1} x)$

Special Functions

`bessel(`x`)` (Bessel function) `beta(`x`)` (Beta function) `erf(`x`)` (error function) `gamma(`x`)` (Gamma function)

Mathematical Constants

`pi` $(\pi \approx 3.14159)$	`eps` $(\varepsilon \approx 2.2204e-16)$	`i and j` (i and j $= \sqrt{-1}$)	`inf` $(\infty, i.e. 1/0)$
`NaN` $(i.e. 0/0)$	`realmin` $(\approx 2.2251e-308)$	`realmax` $(\approx 1.7977e+308)$	`ans` (Name for result)

Special Matrices

`zeros(m,n)(zeros(n))` $(m \times n (n \times n)$ zeros matrix)	`ones(m,n)(ones(n))` $(m \times n (n \times n)$ ones matrix)
`eye(m,n)(eye(n))` $(m \times n (n \times n)$ eye matrix)	`rand(m,n)(rand(n))` $(m \times n (n \times n)$ random matrix)
`linspace(x0,x1,n)` (n linear-spaced vector on [x0,x1])	`logspsce(x0,x1,n)` (n log-spaced vector on [x0,x1])
`meshgrid(x,y)` (size(x)$\times$ size(y) matrix)	`hilb(n)` ($n \times n$ Hilbert matrix)
`invhilb(n)` ($n \times n$ inverse Hilbert matrix)	`magic(n)` ($n \times n$ magic square matrix)
`pascal(n)` ($n \times n$ Pascal matrix)	`rosser` (Classic symmetric 8×8 eigenvalue test matrix)
`wilkinson(n)` ($n \times n$ Wilkinson's eigenvalue test matrix)	

Matrix Analysis

`expm(`x`)` (e^x)	`logm(`x`)` $(\log x)$	`sqrtm(`x`)` $(\sqrt{x})$	x is a matrix
`length(x)` (length)	`size(x)` (size)	`cond(x)` (condition number)	`det(x)` (determinant)
`norm(x)` (norm)	`null(x)` (null space)	`orth(x)` (orthogonalization)	`rank(x)` (rank)
`inv(x)` (inverse)	`pinv(x)` (pseudoinverse)	`hess(x)` (Hessenberg form)	`balance(x)` (diagonal scaling)
`rcond(x)` (LINPACK reciprocal condition estimator)		`rref(x)` (reduced row echelon form)	
`trace(x)` (sum of diagonal elements)		`chol(x)` (Cholesky factorization)	
`lu(x)` (Factors from Gaussian elimation)		`qr(x)` (orthogonal-triangular decomposition)	
`eig(x)` (eigenvalues and eigenvectors)		`poly(x)` (characteristic polynomial)	
`schur(x)` (Schur decomposition)		`svd(x)` (singular value decomposition)	

Data (Vector) Analysis

`cumprod(x)` (cumulative product of vector x)	`cumsum(x)` (cumulative sum of vector x)
`max(x)` (largest component of vector x)	`mean(x)` (average or mean value of vector x)
`median(x)` (median value of vector x)	`min(x)` (smallest component of vector x)
`prod(x)` (product of elements in x)	`sort(x)` (sort in ascending order of vector x)
`std(x)` (standard deviation of vector x)	`sum(x)` (sum of elements of vector x)
`trapz(x)` (numerical integration using trapezoidal rule)	

Fourier transform functions

`abs(x)` (magnitude) `angle(x)` (phase angle)
`fft(x)` (discrete Fourier transform) `fft2(x)` (2-d discrete Fourier transform)
`ifft(x)` (inverse discrete Fourier transform) `ifft2(x)` (inverse 2-d discrete Fourier transform)

Polynomial and interpolation functions

`roots(p)` (compute roots of the polynomial p) `poly(r)` (find polynomial associated with roots r)
`conv(a,b)` (multiply the two polynomials a,b) `deconv(b,c)` (divide the polynomial b into c)
`polyder(p)` (compute the derivate of polynomial p) `polyval(p,x)` (evaluate the polynomial p at all values in x)
`residue(n,d)` (compute the derivate of polynomial p) `polyval(p,x)` (compute the ratio of n to d)
`polyder(n,d)` (compute the derivate of ratio of n to d) `polyfit(x,y,n)` (fit y=f(x) by up to n-th order polynomial)
`interp1(x,y,z)` (1-d linear interpolation of y=f(x) at z)
`interp1(x,y,z,'spline')` (1-d spline interpolation of y=f(x) at z)
`interp2(x,y,z,r,x0)` (2-d linear interpolation of z=f(x,y) with resolution r center x0)
`interp2(x,y,z,r,x0,'cubic'` (2-d cubic interpolation of z=f(x,y) with resolution r center x0)

2-d Graphics

`loglog(x,y)` (log-log scale plot y=f(x)) `plot(x,y)` (plot y=f(x))
`semilogx(x,y)` (semilog on x-axis scale plot y=f(x)) `semilogy(x,y)` (semilog on y-axis scale plot y=f(x))
`bar(x,y)` (bar plot y=f(x)) `comet(x,y)` (animated comet plot y=f(x))
`compass(x,y)` (compass plot y=f(x)) `errorbar(x,y)` (error bar plot y=f(x))
`feather(x,y)` (feather plot y=f(x)) `hist(x,y)` (Histogram plot y=y(x))
`feather(x,y)` (feather plot y=y(x)) `polar(t,r)` (polar coordinate plot r=r(t))
`stairs(x,y)` (stairstep plot of y=y(x)) `rose(x,n)` (angle histogram plot of x using n equally bins)
`stairs(x,y)` (stairstep plot of y=y(x)) `stem(x,y)` (Stem plot for discrete sequence data y=y(x))
`fill(x,y,c)` (draw filled 2-d polygonsdefined by x,y with color specified byc)

3-d Graphics

`fill3(x,y,z,c)` (filled 3-d polygons) `plot3(x,y,z)` (plot with (x,y,z) as coordinate)
`comet3(x,y,z)` (3-d animated comet plot) `contour(x,y,z,n)` (contour plot of z=z(x,y))
`contour3(x,y,z,n)` (3-d contour plot of z=z(x,y)) `contourc(x,y,z,n)` (contour plot computation)
`quiver(x,y,dx,dy)` (quiver plot of (dx,dy)) `pcolor(x,y,z)` (pseudocolor(checkboard) plot of z=z(x,y))
`mesh(x,y,z)` (3-d mesh surface plot of z=z(x,y)) `meshc(x,y,z)` (combination mesh/contour plot)
`surf(x,y,z)` (3-d shadedsurface plot of z=z(x,y)) `surfc(x,y,z)` (combination surf/contour plot)
`surfl(x,y,z)` (3-d shaded surface with lighting) `meshz(x,y,z)` (3-d mesh with zero plane plot of z=z(x,y))
`waterfall(x,y,z)` (waterfall plot of z=z(x,y))

Graph annotation

`grid on(off)` (grid (no)) `gtext('x')` (add text x) `legend 'x'` (add legend) `title 'x'` (add title)
`xlabel('x')` (add xlabel) `ylabel('x')` (add ylabel) `zlabel('x')` (add zlabel) `clabel` (contour label)
`axis on (off)` (axis (no)) `hold on(off)` (hold (no)) `colormap gray` (→ gray) `axis(c)` (change axisscale)
`rotate3d` (rotate) `figure(n)` (create figure) `shading flat` (shading flat)

Programming

```
for x=array          while expression     if expression        if expression        if expression1
  commands             commands             commands             commands1            commands1
end                  end                  end                  else                 elseif expression2
                                                                  commands2              commands2
                                                                end                  else
                                                                                       commands3
                                                                                     end
```

Relational Operators

`==` (equal to) `=` (unequal) `>=` (greater or equal) `>` (greater)
`<` (less than) `<=` (less or equal)

Logical Operators

`&` (and) `|` (or) `~` (not)

Input/Output

`a1` (excute a m-file a1.m) `save a.dat` (save matrix variable a to a disk file a.dat)
`load a.dat` (retrieve matrix variablea from file a.dat on disk)