Summary of Linear Algebra in MATLAB

NUMERIC Matrix and Vector Data Entry (Log onto *Turing*. Enter **matlab**.)

more on ENTER	Enable paging. SPACE to display next page.
A = [2 1 3; -5 2 -1; 6 -1 4] ENTER	Semicolon terminates matrix row.
B = [4 3 ENTER	ENTER also terminates a matrix row.
1 2] ENTER	
C = [2 -1 4 6; 0 1 5 ... ENTER	Ellipsis to continue same matrix row.
7; 2 -9 0 3] ENTER	
w = [9; 1; 6] ENTER	Column vector.
v = [5 -2 4], u = [2; -7; 9] ENTER	Comma to separate multiple commands.
disp(C) ENTER	Suppresses the "C=" in the display.
C(: , 2:4), C(2:3, 2) ENTER	Access parts of a matrix.
E = A; ENTER	Semicolon after command suppresses printing.
E(3, 1) = 7 ENTER	Modify a matrix element.
C(3, :) = [-4 7 2 0], C(:, 1) = w ENTER	Modify a matrix row or column.
save pracdata ENTER	Save practice data for loading at later session.
ans	Variable for the most recent answer.

UpArrow to recall previous commands for editing and re-entry.

Special Constants, Matrices, and Functions

inf infinity; **NaN** Not a Number; **i** imaginary unit √(-1); **sqrt(2)** √2

eye(3), **eye(3,5)**, **eye(5,3)**	Identity matrices.
zeros(3), **zeros(3,5)**	Zero matrices.
rand(3), **rand(3,5)**	Random matrices.
diag(A), **diag(v)**	Diagonal of matrix, diagonal matrix.
min(v), **max(C)**, **[Y, I] = max(C)**	Min & max of vector, matrix columns.
sum(v), **prod(w)**, **sum(C)**	Sum & product of vector, matrix columns.
fliplr(B), **flipud(B)**, **rot90(B)**	Transform position of matrix elements.

System commands

help, **help eig**	Help on commands.
who, **whos**	List current variables.
clear B, C, v	Clear some current variables.
cd *yourdirectory*	Change to your working directory.
save *yourfile*	Store current variables in your file.
load *yourfile*	Retrieve variables from your file.
delete *yourfile*	Erase your file.
dir	List files in current directory.
diary *yourdiary*	Save session a your diary file.
diary off	Suspend recording of your session.
type *yourdiary*	View your diary file.
quit or **exit** **logout**	First leave MATLAB. Then leave *Turing*.

NUMERIC Matrix and Vector Operations

A + E, **A - E**, **2*w**, **3*A + 2*E**, **5*v - 3*A(1, :)**	Linear combinations.
A*C, **v*A**, **A*w**, **C*A**	Matrix products.
inv(B), **B^3**, **B^(-1)**, **B^(-2)**, **inv(E)**	Matrix inverse and powers.
inv(A), rats(ans)	Convert display of answer to simple rational form.
inv(A)*w, **A\ w**, **v*inv(A)**, **v/A**, **X = A\ C**, **A*X-C**	Solution to A*X = C.
det(B), **trace(A)**, **rank(B)**, **rank(C)**, **rank(E)**, **C′**	Transpose.
u′*w, **v*v′**, **sqrt(w′*w)**, **norm(w)**, **norm(C)**	Dot product & norm.
length(v), **length(w)**, **size(A)**, **size(v)**, **size(w)**	Dimensions of matrix.

T = C(2, :) ; C(2, :) = C(3, :) ; C(3, :) = T	Interchange rows 2 and 3 of C.
C(1, :) = 2*C(1, :)	Double the first row of C.
C(3, :) = 2*C(1, :) + C(3, :)	Add twice first row to 3rd row.

NUMERIC Linear Algebra Algorithms

R = rref(E), rats(R)	Reduced row-echelon form, display rational.
format rat	Display answers as simple rational.
rref(E)	Notice the display in common fractions.
format long	Switch to display of 15 digits.
rref(E)	
rrefmovie(C)	Watch steps in Gaussian elimination.
format short	Switch to display of 5 digits.

[L, U] = lu(A)	A = L*U. L permuted lower triangular, U upper triangular.
[L, U, P] = lu(A)	P*A = L*U. L lower, U upper, P permutation matrix.
Q = null(C)	Columns of Q form orthonormal basis for nullspace of C.
Q = orth(C)	Columns of Q are orthonormal basis for column space of C.
[Q, R] = qr(A)	A = Q*R. Q has orthonormal columns, R upper triangular.
[V, D] = eig(B)	B*V = V*D. Columns of V are eigenvectors, D diagonal.
d = eig(B)	eigenvalues of B.
p = poly(A)	Coefficients of characteristic polynomial: det(λI-A).
q = poly(w)	Row of coefficients of polynomial with roots column w.
r = roots(q)	Column of roots of polynomial with coefficients row q.
s = svd(C)	Singular values of C.
[U, S, V] = svd(C)	C = U*S*V′. U,V orthogonal, S diagonal singular values.
c = cond(C)	Condition number = ratio of largest singular value to smallest.

SYMBOLIC Matrix and Vector Data Entry (Quit MATLAB and start it fresh.)

more on ENTER	Enable paging whenever you start MATLAB.
load pracdata ENTER	Retrieve the numeric matrices saved earlier.
H = sym('[1, 2; k, 3]') ENTER	Note use of commas in symbolic matrix.
z = sym('[a; b]') ENTER	Quotes enclose things that are symbolic.
K = sym(A) ENTER	Copy numeric matrix to symbolic matrix.
sym(K, 3, 1) ENTER	Access matrix element.
L = sym(K, 3, 1, 't') ENTER	Modify matrix element.
M = sym('[(x+5)/(x+3), x^2-4; sin(x)^2 + cos(x)^2, (x+y)*(x^2+y)]') ENTER	
pretty(M) ENTER	Alternate display for symbolic matrices.
save symdata ENTER	Save the combined practice and symbolic data.

SYMBOLIC Matrix and Vector Operations

symadd(A, L), **symsub(A, L)**
symmul(H, L), **symmul(H, z)**, **sympow(H, 2)**
symdiv(A, L), **subs(ans, 6, 't')** Compute A/L and substitute.
transpose(H), **determ(H)**, **inverse(H)**
symop(H, '*', z, '-', z) Computes H*z - z. Possible symbols: **+ - * / ^ ( )**

SYMBOLIC Linear Algebra Algorithms

x = linsolve(H, z)	Solve linear system: H*x = z.
X = linsolve(H, B)	Solve linear systems: H*X = B.
[x, Z] = linsolve(E, u)	General solution: x + tZ
nullspace(E)	Compare nullspace of E with Z in line above.
colspace(E)	Columns form basis for column space of E.
colspace(C′)	Columns form basis for row space of C.
charpoly(H)	Symbolic characteristic polynomial for H.
d = eigensys(H)	Symbolic eigenvalues for H.
[V, d] = eigensys(H)	Columns of V are eigenvectors for H. "E(1)" and "E(2)" denote the two eigenvalues.

SYMBOLIC Simplification

disp(M), **Simplify(M)**	Observe carefully what happens.
expand(M), **factor(ans)**	
collect(M, 'y')	May omit symbolic variable if only one.