EEB 324: Theoretical Ecology (TA: Marissa Baskett)
Useful Matlab commands

Basics	
help functionName	Get quick help on function functionName; you can also use the Help menu
+ - * / ^	Simple addition, subtraction, multiplication, division, and power
.* ./ .^	Element-by-element multiplication, division, and power for vectors and matrices (e.g., [1,2].*[3,4] = [3,8])
x = 4	Assign the value 4 to x
x = 4**;**	Assign 4 to x without reporting back
sin(x), **cos**(x), **tan**(x)	sine, cosine, and tangent
exp(x), **log**(x), **log10**(x)	exponential, natural log, base-10 log
abs(x), **sqrt**(x)	Absolute value, square root

Vectors	
v = 1**:**10	A vector from 1 to 10 in increments of 1
v = 1**:**0.5**:**10	A vector from 1 to 10 in increments of 0.5
v = [1**,** 3**,** 8]	A vector with values 1, 3, 8
v**(**3**)**	Third element of vector v
v(**end**)	Last entry in vector v
length(v)	Length of vector v
sum(v)	Sum of all entries in vector v

Matrices	
M = [1 2**;** 3 4]	Create the matrix $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
M(1,**:**)	The first row of M
size(M)	Size of the matrix M
zeros(2,3), **ones**(2,3)	Create a 2×3 matrix of zeros or ones
eye(3)	3 by 3 identity matrix (ones on diagonal, zeros everywhere else)
sum(M)	Sum of each column in matrix M; *sum(M,**2**)* gives sum across rows
det(M)	Determinant of matrix M
trace(M)	Trace of matrix M
M**'**	Transpose of matrix M
evals = **eig**(M);	Eigenvalues of M
[evecs, evals] = **eig**(M);	Eigenvalues and eigenvectors of M
M*****v	Multiplication of matrix M and vector v

Random numbers	
rand(m,n)	m by n matrix of Uniform(0,1) random entries
randn(m,n)	m by n matrix of Normal(0,1) random entries

Plotting	
figure	Start a new figure
hold on	Put multiple plot commands on the same plot
plot(x,y)	Plot x vs. y
xlabel('x'), **ylabel**('y'), **title**('Figure 1')	Label plot
subplot(rows, cols, num)	Work in subplot #*num* in a figure with *rows* by *cols* subplots
bar(v)	Plot a bar graph of vector v
print -dpdf filename.pdf	Save the current figure to a pdf file (best for opening and printing out a graph by itself)
print -djepg filename.jpeg	Save the current figure to a jepg file (best for inserting a graph into a MSWord file)

Loops	
> < <= >= == ~=	Comparisons: less than, greater than, less than or equal to, greater than or equal to, equal to, and not equal to
& | ~	Logical operators: and, or, not
if(*test*) *action*; end	If *test* (e.g., x<0) is true, do *action* (e.g., x = -x)
if(*test*) *action 1*; **else** *action 2*; end	If *test* is true, do *action 1*, and if not, do *action 2*
if(*test 1*) *action 1*; **elseif**(*test 2*) *action 2*; end	If *test 1* is true, do *action 1*, or if *test 2* is true, do *action 2*
for(i=1:10) *action*; end	For i equals 1, 2, 3, ..., 10, do *action*
while(*condition*) *action*; end	While *condition* (for example, x>=0 & x<=10) is true, do *action*

Scripts and functions	
function [output] = myfun(input)	create a function (saved in the file myfun.m) that takes input and gives output
% This function is for ...	Comments in function and script files: any text after the % is ignored by Matlab
global x	Make x a global variable, so that a value assigned to x outside a function will be recognized in that function
clear x	Clear the value for the variable x; writing *clear* clears values for all variables
fprintf('Some words here')	Write *Some words here* in the command window

Oridinary differential equations	
[vt, vx] = **ode45**(@myodefun, [t0 tf], x0); *where* function dxdt = myodefun(t, x) dxdt = ...	Numerically integrate myodefun, starting at x0, from time t0 to tf; this gives vector of values vx at the times in vector vt

Input/Output	
For input and output within Matlab	
save('fileName.mat','x','y')	Save x and y in a file called fileName.mat (Matlab format)
load('fileName.mat')	Upload the variables in fileName.mat (Matlab format)
For input and output between Matlab and spreadsheet programs (e.g., Excel)	
A = **dlmread**('fileName', '\t')	Read the contents of *fileName*, where columns are separated by tabs (the delimiter can be anything, like ',' for commas, etc.), with the option of specifying the range as a third argument in the form of [firstRow firstCol lastRow lastCol], where counting starts with zero
dlmwrite('fileName', A, '\t')	Save the matrix A in a file called *fileName*, with columns separated by tabs
For output to the command window	
fprintf('some text')	Write *some text* in the command window
fprintf('some text**n** more text')	Write *some text*, then *more text* on a new line
fprintf('Time = **%f**', t)	Write *Time = 4* (or whatever number t is)
For input from and output to text files	
fid = **fopen**('fileName.txt', **'r'**)	Open existent fileName.txt to read its contents and save the file's identifier as *fid*
A = **fscanf**(fid, '%f %f', [2 inf])	Take a file, identified by *fid*, with contents in the form of: 0.1 0.2 0.2 0.4 0.3 0.6 etc. and store the contents in a 2-row matrix A
fid = **fopen**('fileName.txt', **'a'**)	Open new or existent fileName.txt to append to its contents and save the file's identifier as *fid*
fid = **fopen**('fileName.txt', **'w'**)	Open new or existent fileName.txt to write to it, erasing any current information, and save the file's identifier as *fid*
fprintf(**fid**, 'some text')	Write *some text* in the file identified by *fid*; see above for more options
fclose(fid)	Close the file identified by *fid*
Use the Help Menu for more options and examples	