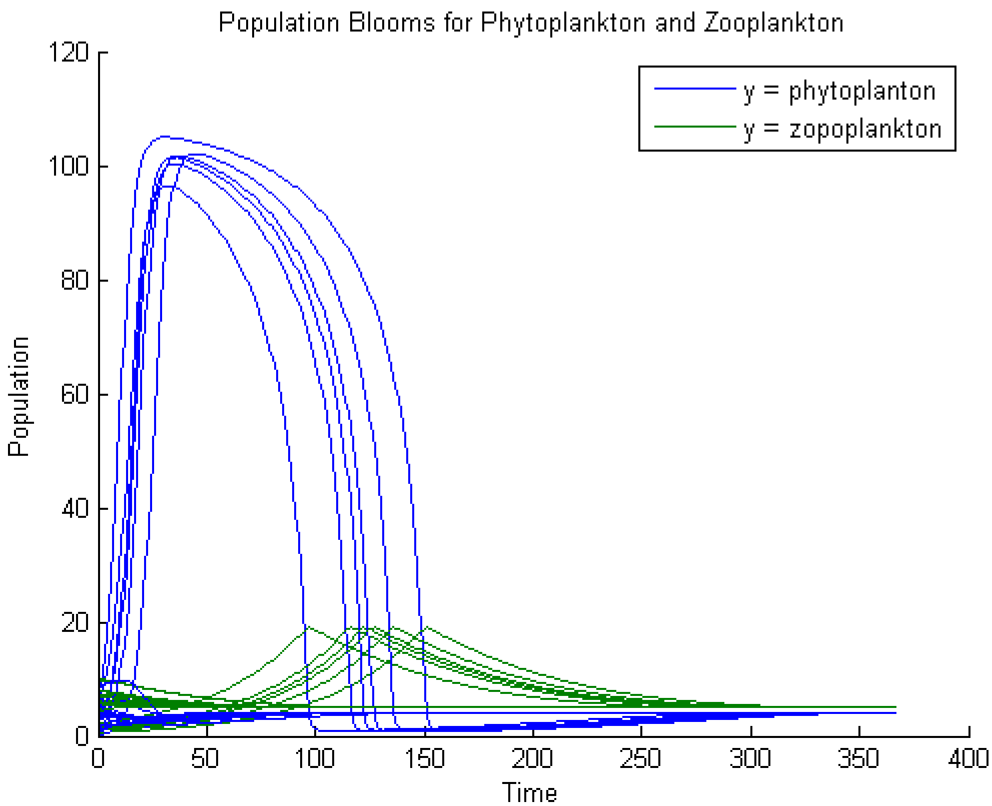
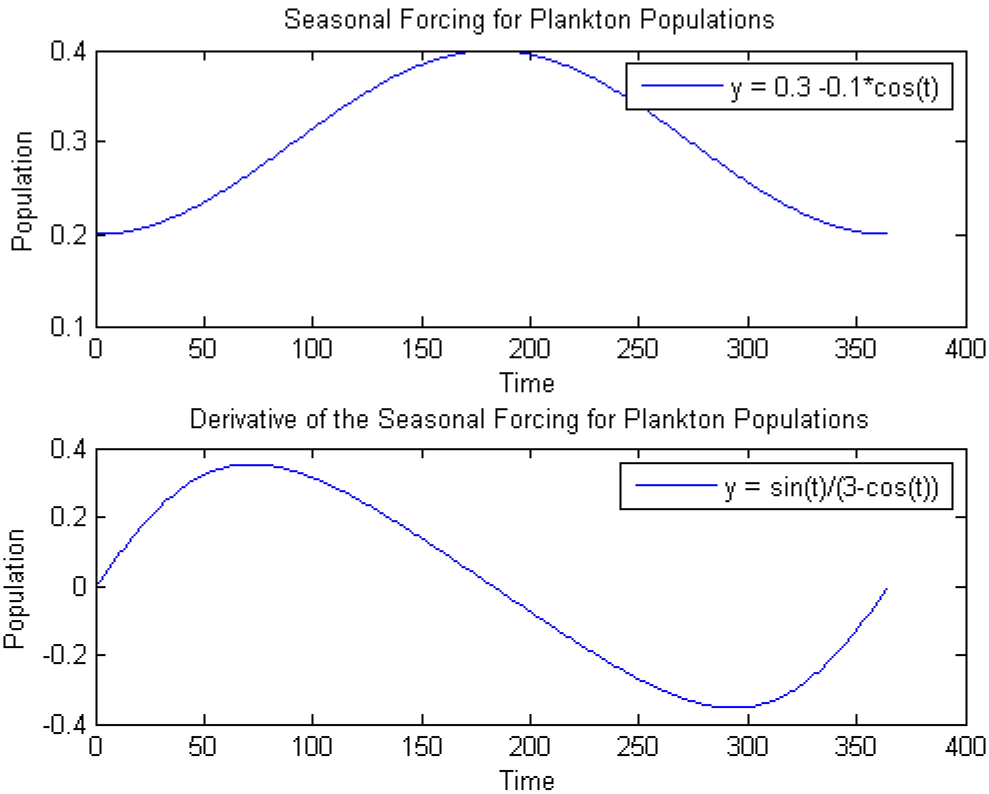
Matlab Assignment 3
Simon Stead

Question 1
Question 1 is very straightforward. ‘planktonderivs.m’ contains equations 2 and 3 from the paper, with all parameters as constant. I then integrated 20 instances of the system through time with ODE45 in ‘planktonode.m’.



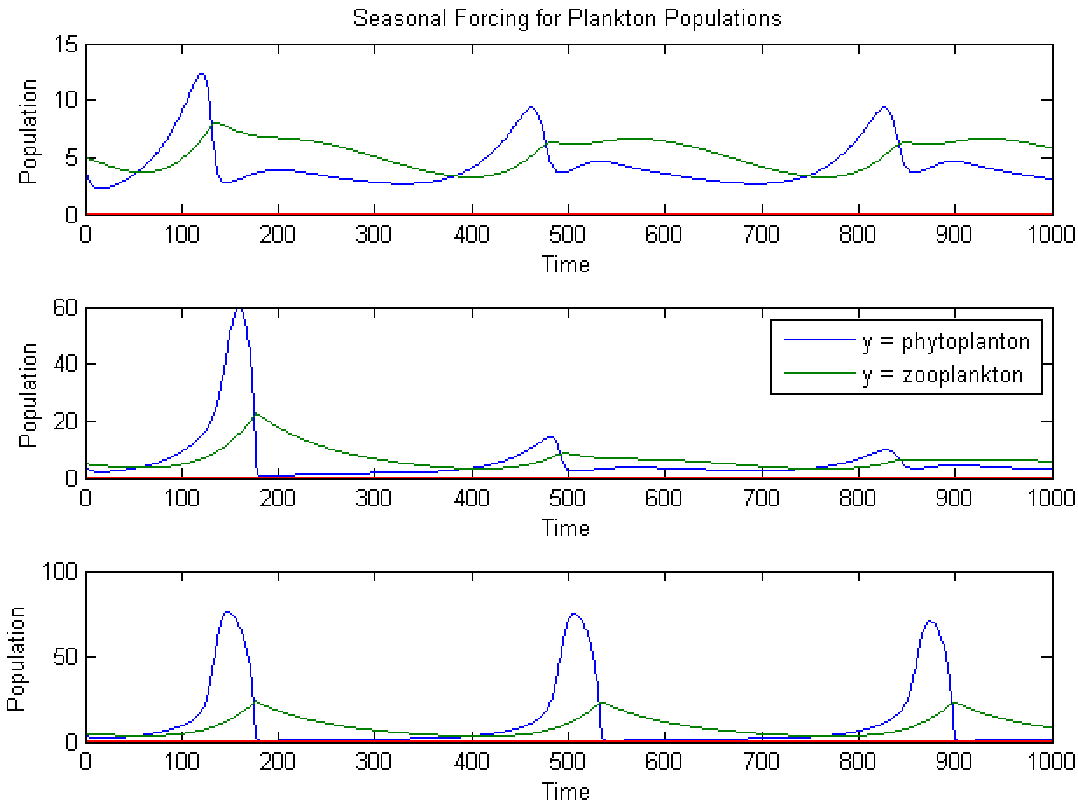
Question 2
I wrote two functions, ‘seasons.m’, ‘seasons2.m’. The first was to calculate the seasonal forcing function, and the second for the reciprocal of the derivative 1/(dr/dt). ‘question2.m’ then takes these functions, and plots them over 365 time steps to simulate the course of a year.



Question 3

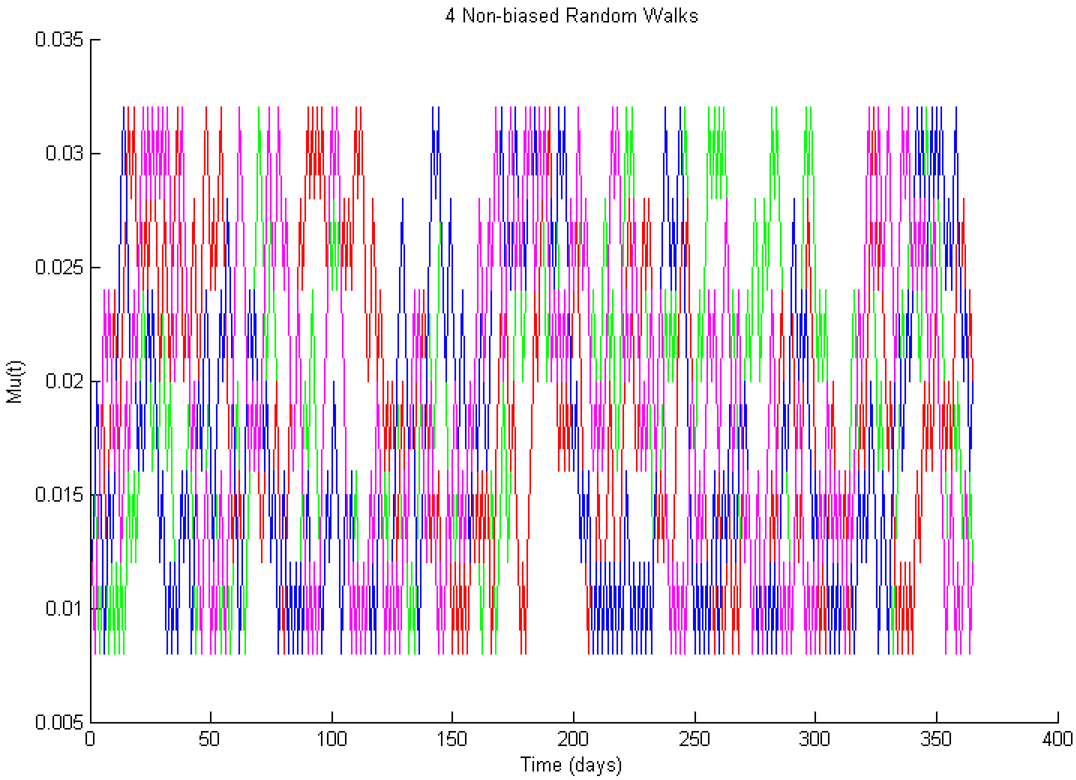
'periodicforcing.m' is near identical to 'planktonderivs.m', except now our r term is being affected by the equation from 'seasons.m'. It was easier to write the equation into 'periodicforcing.m' rather than calling 'seasons.m' it's necessary to vary the amplitude for this question. It takes in 3 variables, P, Z, and then A, the amplitude.

The questions asks for the minimum amplitude to cause a bloom, which I found to be 0.167. 'question3.m' takes the steadystate from question 1 and integrates 'periodicforcing.m' through time 3 times, one A = 0.167, and then for A +/- 0.001, to illustrate the effects of a perturbation of our amplitude.



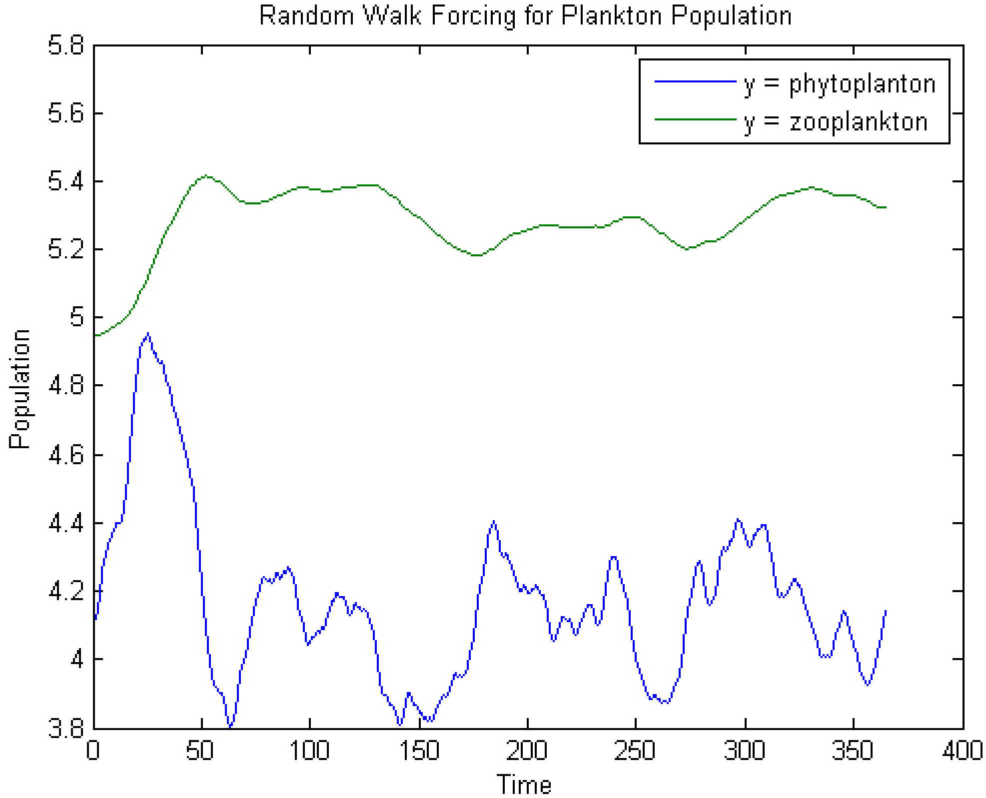
Question 4

‘randomwalk.m’ takes in a 6-vector of [probability, number of steps, initial position, step size, lower reflecting boundary, upper reflecting boundary], which is all the information we need to uniquely define our random walk. Pat A (‘question4.m’) requires four trajectories of this random walk.

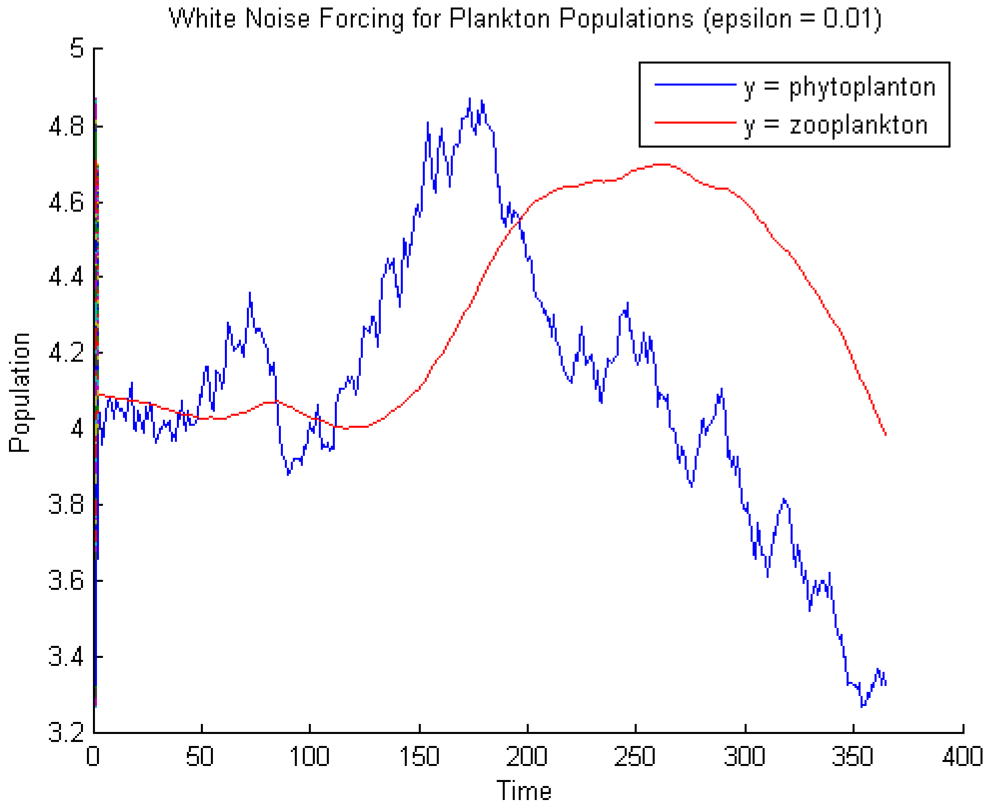


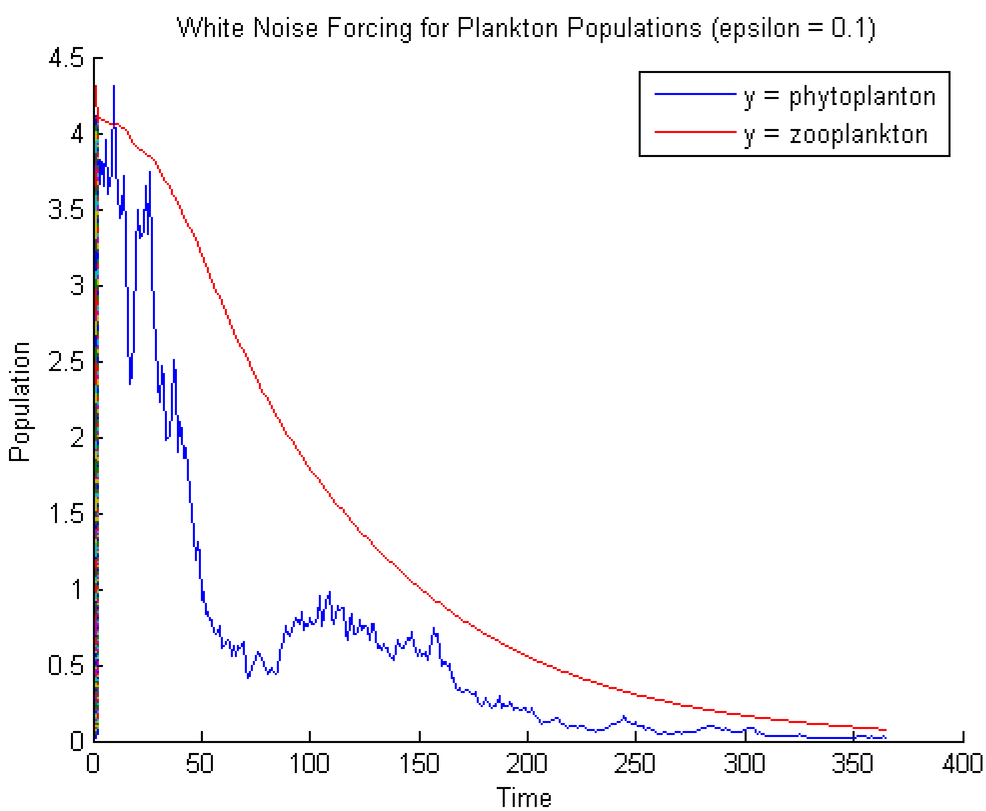
It then generates a uniform random number, and uses a for loop to step up and down the walk accordingly. In ‘randomwalkforcing.m’, we reset r to be 0.3, and instead vary mu. ‘question4.m’ sets up a vector of initial conditions and generates 4 unique random walks, displayed above.

‘question4b.m’ generates a random walk as above, and uses the euler method to estimate the time taken until a bloom appears. The graph suggests this perturbation is not large enough to cause a bloom.



Question 5
Again ‘whitenoiseforcing.m’ is similar to our previous set ups, except we modify equation (2) from the paper by adding an extra term, the white noise. Using the Euler–Maruyama method we generate a vector of values whose consecutive dis
tance is normally distributed.
‘question5.m’ users an euler method to discretize the time stamps and approximate the integratal. Below are the plots for epsilon = 0.1 and 0.01.


White Noise Forcing for Plankton Populations (epsilon = 0.01)
y = phytoplanton
y = zooplankton
Population
Time
5
4.8
4.6
4.4
4.2
4
3.8
3.6
3.4
3.2
0
50
100
150
200
250
300
350
400


White Noise Forcing for Plankton Populations (epsilon = 0.1)
y = phytoplanton
y = zooplankton
Population
Time
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
50
100
150
200
250
300
350
400