Introduction

Investigated the population growth of a parasite species using continuous and discrete analysis methods. System was modelled based on the given equations and solved using ODE45. A discrete analysis was carried out on a 200 by 200 grid to simulate the interactions between single agents within the system.

Methods

The simulation is carried out by iterating over each parasite agent at each time step of the simulation. An m by 3 matrix is used to store the [x, y, age] of each parasite. The parasite matrix is iterated over to move each individual parasite agent.

The food agents are stored in a m by 2 matrix representing the [x, y] coordinates.

In order to check collisions between agents a 200 by 200 mask is created. Three values (1, 0, 0.5) are used to represent an empty space, a parasite agent and a food agent respectively.

There are two functions used to set up the mask, parasite matrix and food matrix for a simulation. The first function randomly places both food and parasite agents on the grid. The second function localises the placements of food in one of four centre quadrants and randomises the parasite placements. Refer to q3/food\_parasite\_random\_placement.m and q3/localised\_food\_random\_placement.m for source code.

There are two simulation functions for each case of food reproduction. Simulation case 1 reproduces food in neighbouring cells and simulation case 2 reproduces food at a set number and randomly on the grid.

The simulation for case 1 is carried out in the following steps:

Iterate over each parasite using the parasite matrix. Generate a random direction and check for a collision. If the cell is empty move the parasite there. If there is a parasite there then there is no movement. If collision with a food agent occurs, update the position of the parasite which has consumed the food. Add a parasite to the mask where the existing food is. Append a new [x, y, age] row to the parasite matrix with appropriate values representing newly created parasite. Set the corresponding consumed food’s row of the food matrix to [-1 -1] (means it has been consumed).

Deleting elements is inefficient in MATLAB due to static nature of arrays. Deleted values are instead set to [-1, -1]. At the end of a simulation step, the negative values of the matrix are identified with a mask. The non-negative values are copied from the original matrix to a new matrix. This is to prevent iterating over more values as food and parasites will be continually deleted from their corresponding matrices.

Iterate over each food element, iff a uniform rand number is less than specified threshold, place a food element in the corresponding neighbouring cell. Depending on the rand number, the reproduced food can be N, S, E or W from the initial food location. Add a new row to food matrix representing the newly produced food agent. Update the mask with the new food agent.If a new uniform rand number is less than specified threshold then the food agent is deleted and the mask is updated.

Reallocate matrices by copying non negative rows to new matrix

For simulation case 2, the food reproduction is replaced with the following steps:

Create N food agents where N is the specified food creation number. Place the food in random x and y coordinates provided that the coordinates are empty. Update the mask for each food element. Append each new food row to the food matrix

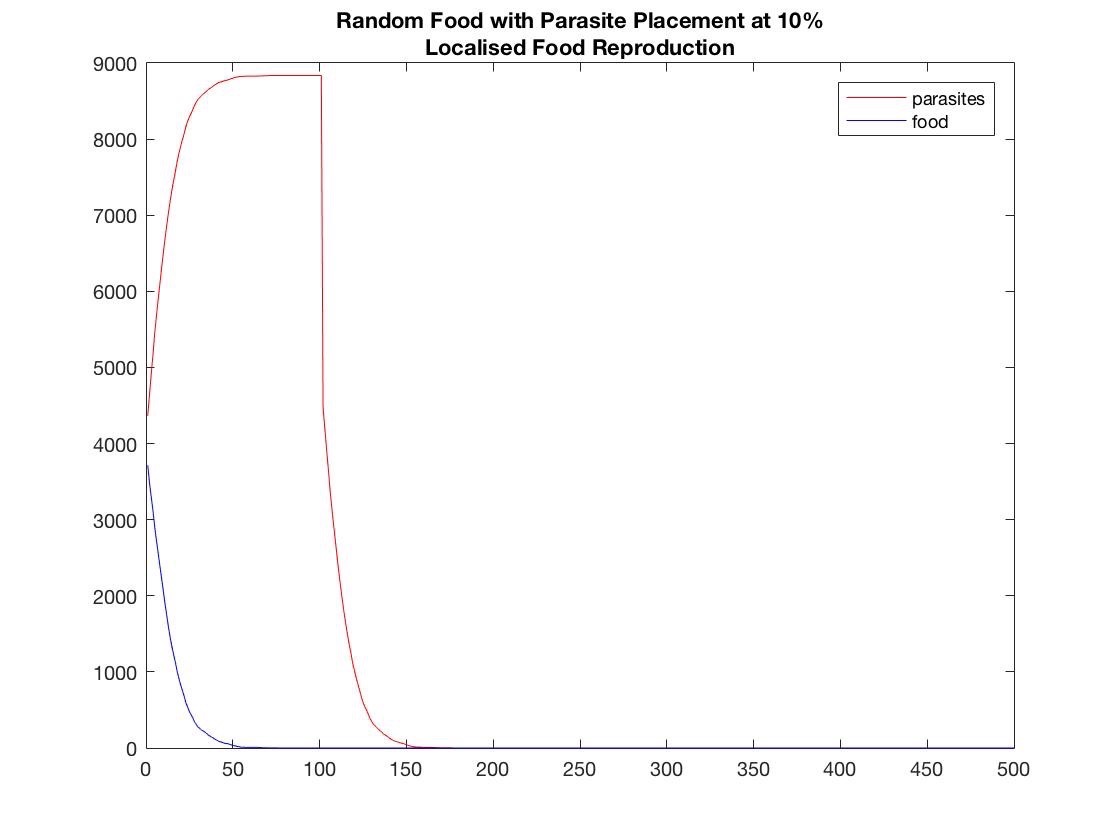
The population counts of the parasites and food agents are recorded after each simulation step for both functions.

Refer to q3/simulation\_case\_1 for the simulation using local food reproduction and q3/simulation\_case\_2 for the simulation using random food reproduction.

**Results**

**Scenario 1: Food extinction**

Food extinction was achievable for all densities and food placements and for the specific case of local food reproduction. In general, a low food creation threshold (birth rate) was selected. For the case of random food reproduction, food extinction was impossible because a set number of food agents are created at each step. An example plot showing food going extinct followed by parasites.



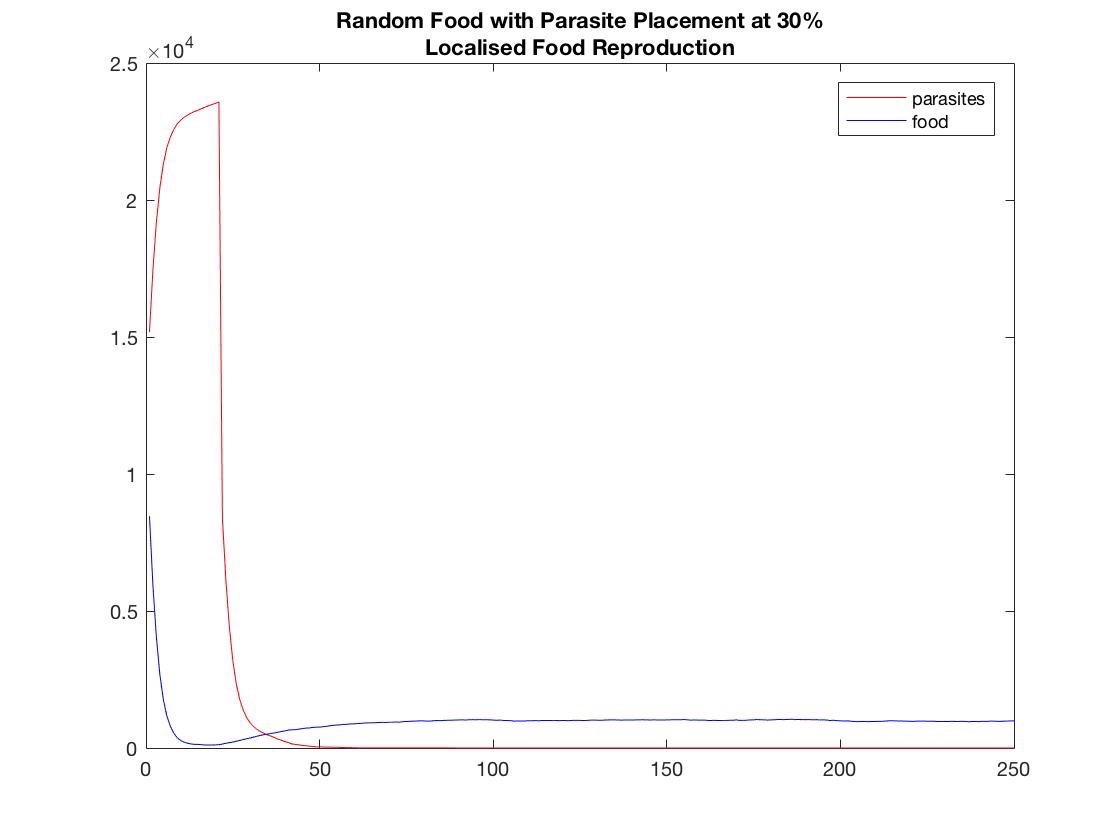
**Scenario 2: Parasite extinction**

Parasite extinction while maintaining food levels was achievable for all cases. For random food reproduction, the following parameters were used for all densities:

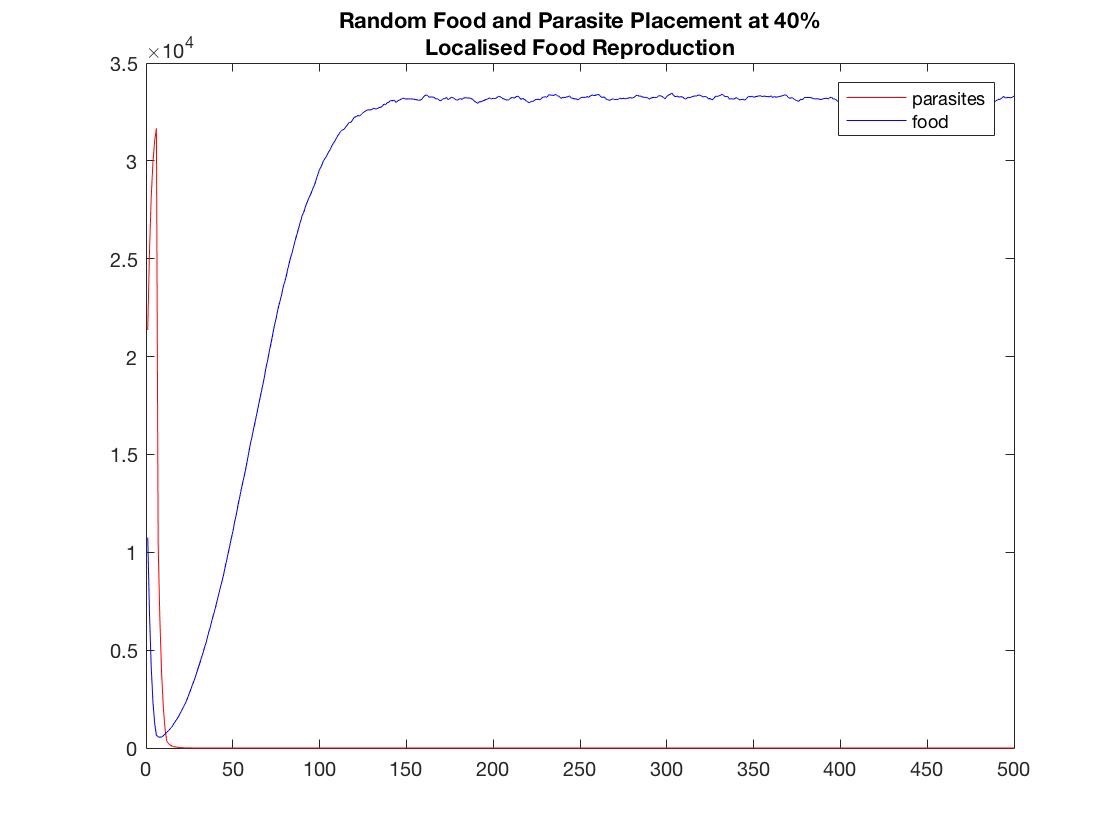
Parasite Age = 20;

Food Death Threshold = 0.05;

Number of food agents created at each step = 50;

By limiting the parasite age, the food is able to thrive, however creating too many food agents will cause the parasite population to bounce back. Therefore a low number of food agents are created at each time step. The case for 30% density with random food and localised food placements are graphed below.

Localised food reproduction with local food initial placement at densities of 30% and 40% required more effort to achieve parasite extinction. The parasites would surround the food and block them from reproducing. Therefore the age of the parasites was reduced to allow the food population to grow. Refer to video ‘case41.avi’ for parasites going extinct at 40% density. The graph for parasites going extinct at 40% is displayed below. Refer to ‘case41.avi’ for parasite extinction at 40% density with random food placement and local food production.

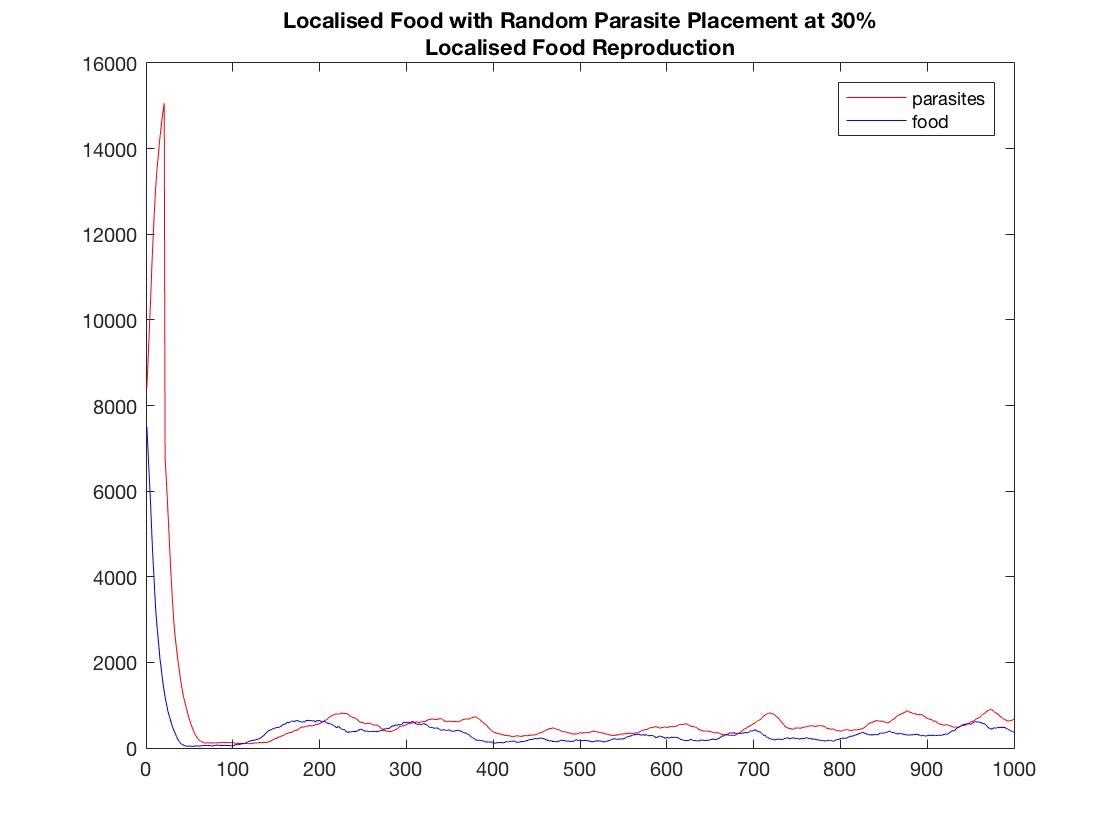
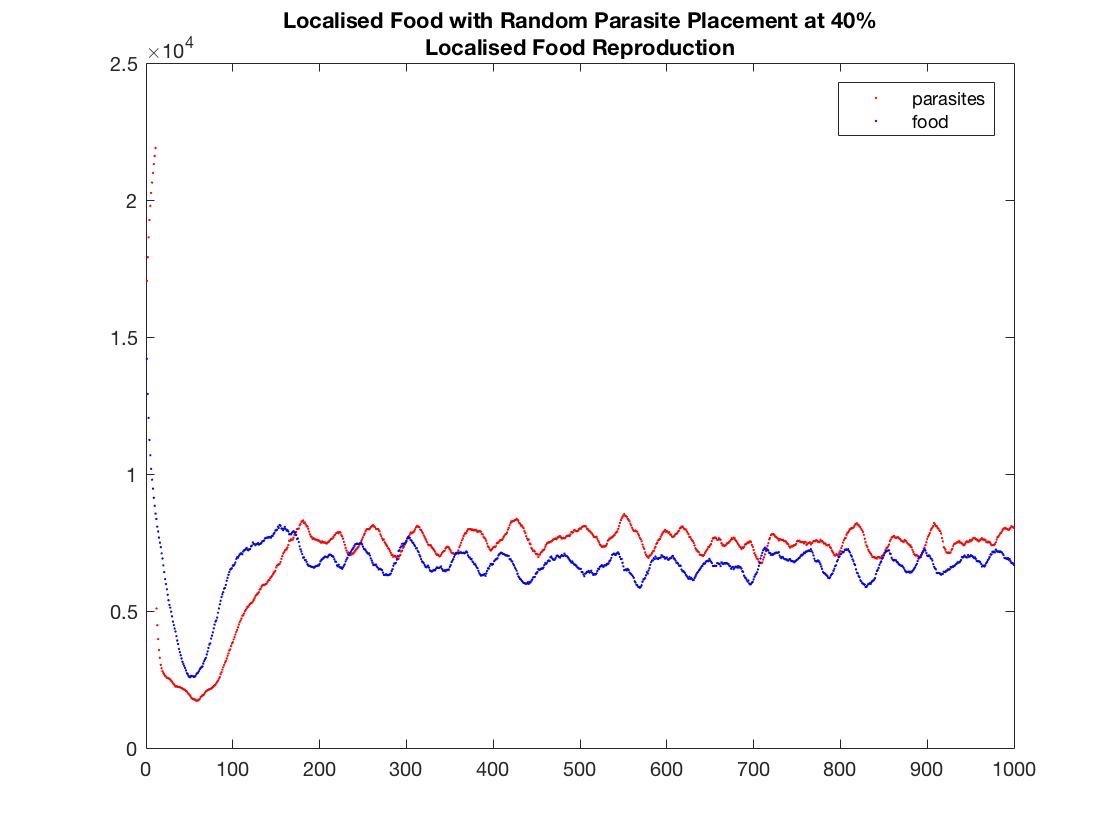


Refer to ‘case3.avi’ for parasite extinction at random food placement with local food reproduction at 10% density.

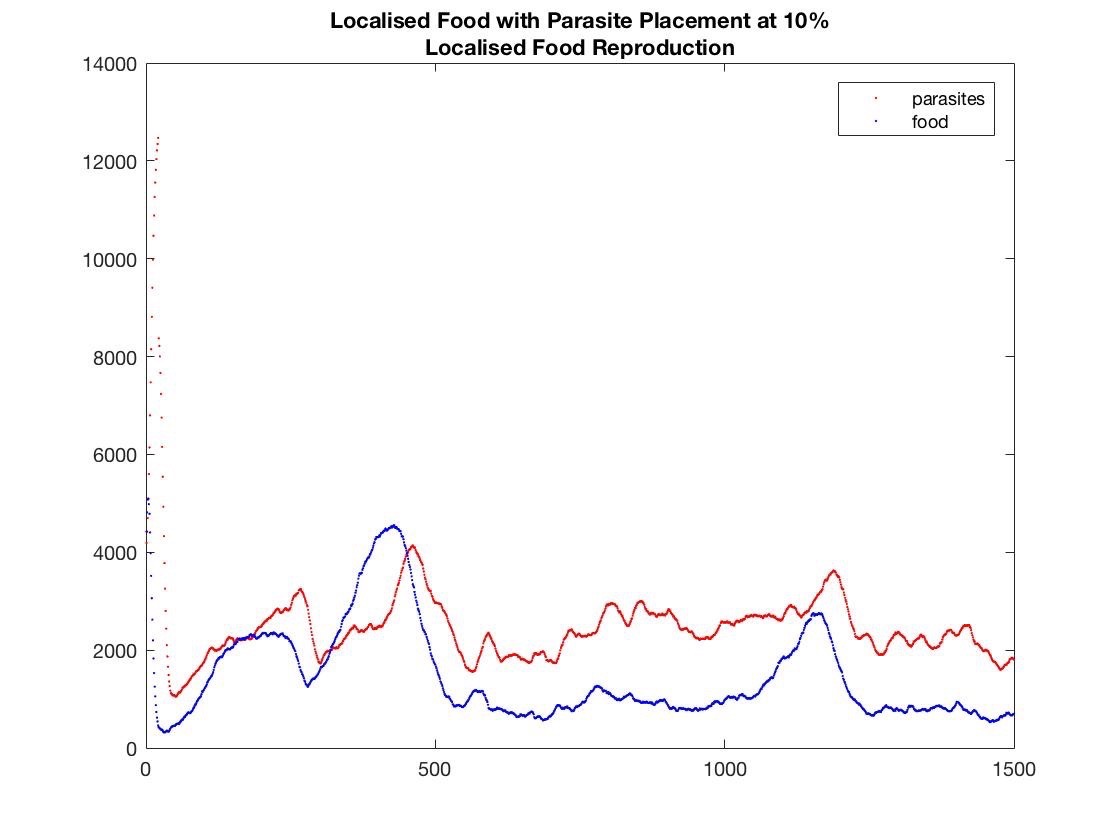
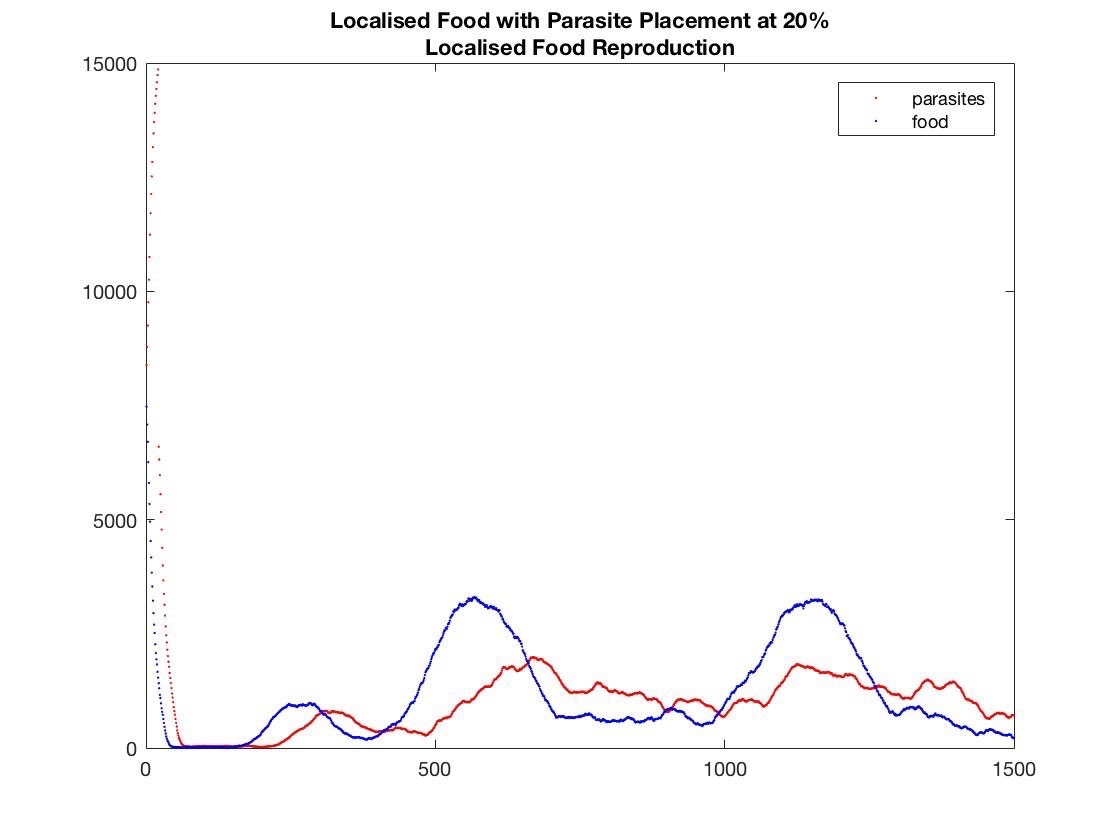
**Scenario 3: Equilibrium**

**Localised food placement with local food reproduction**

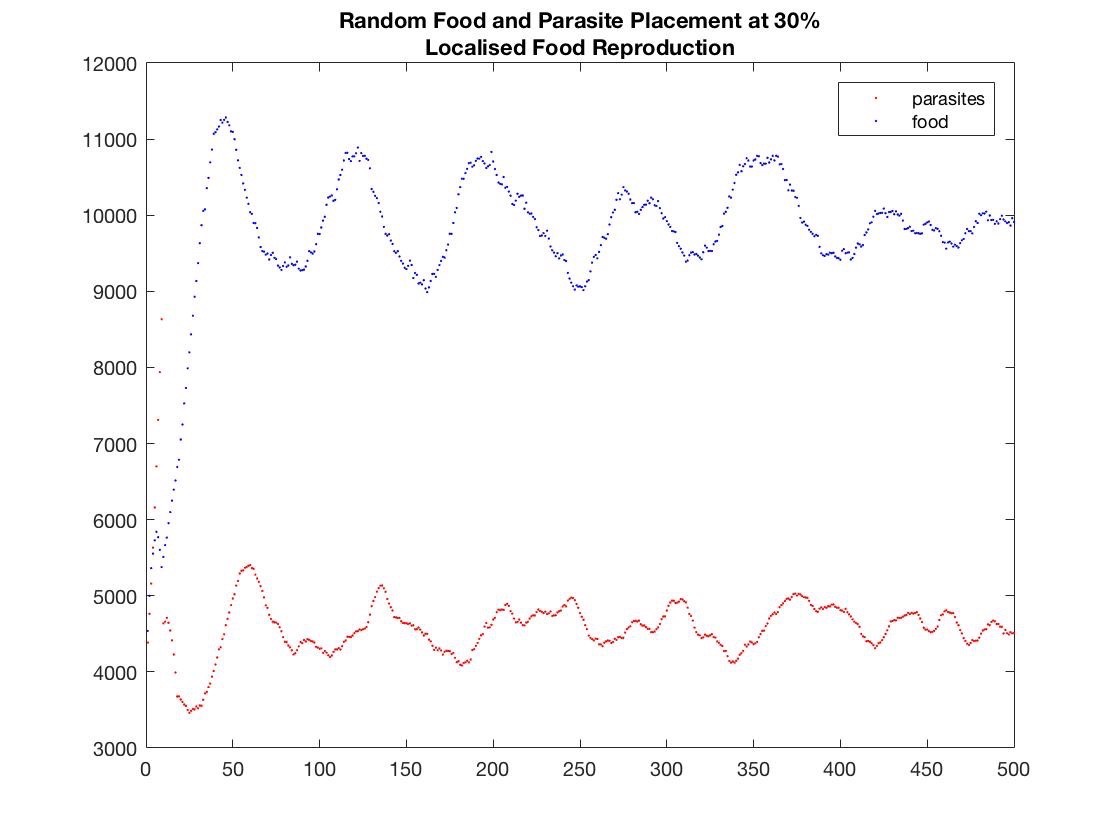
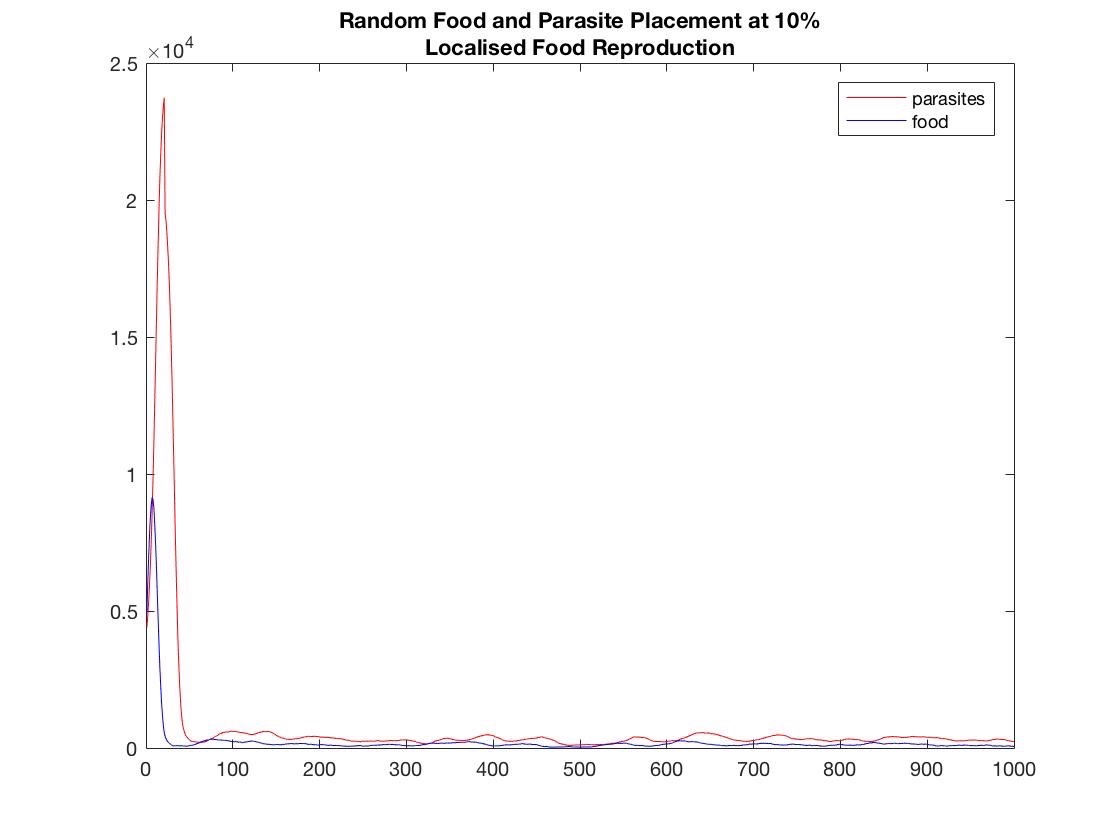
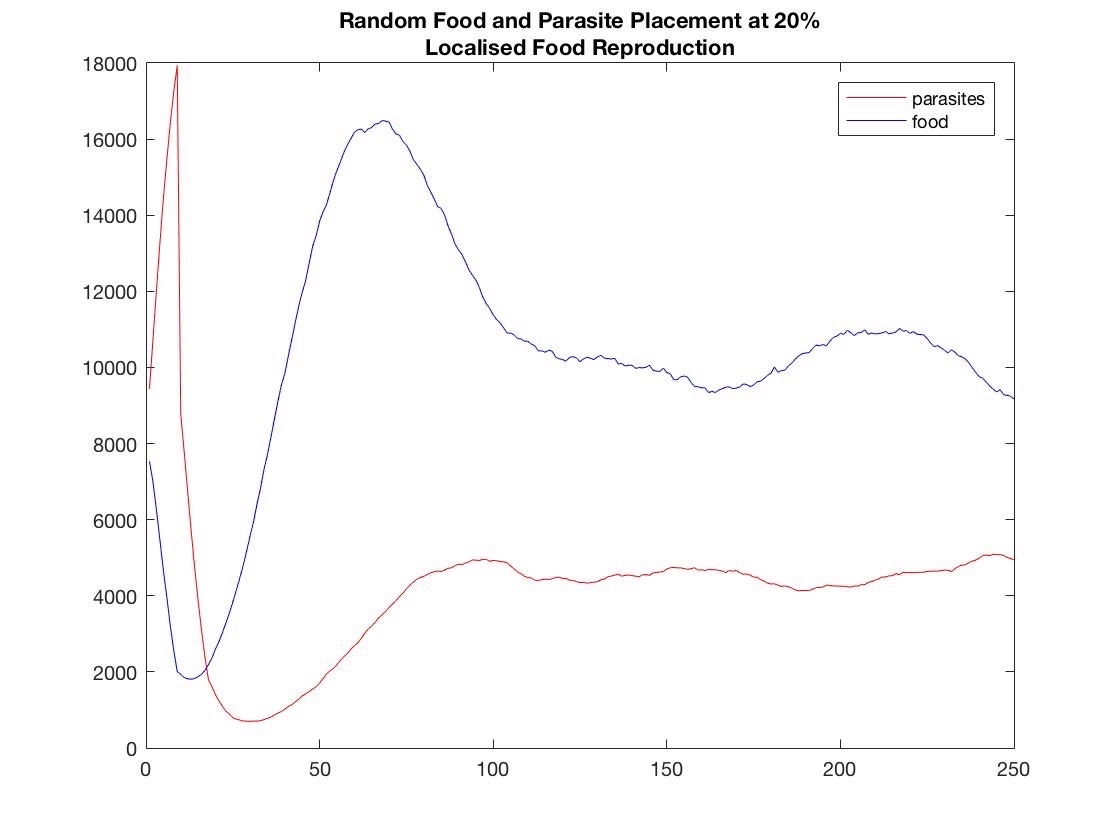
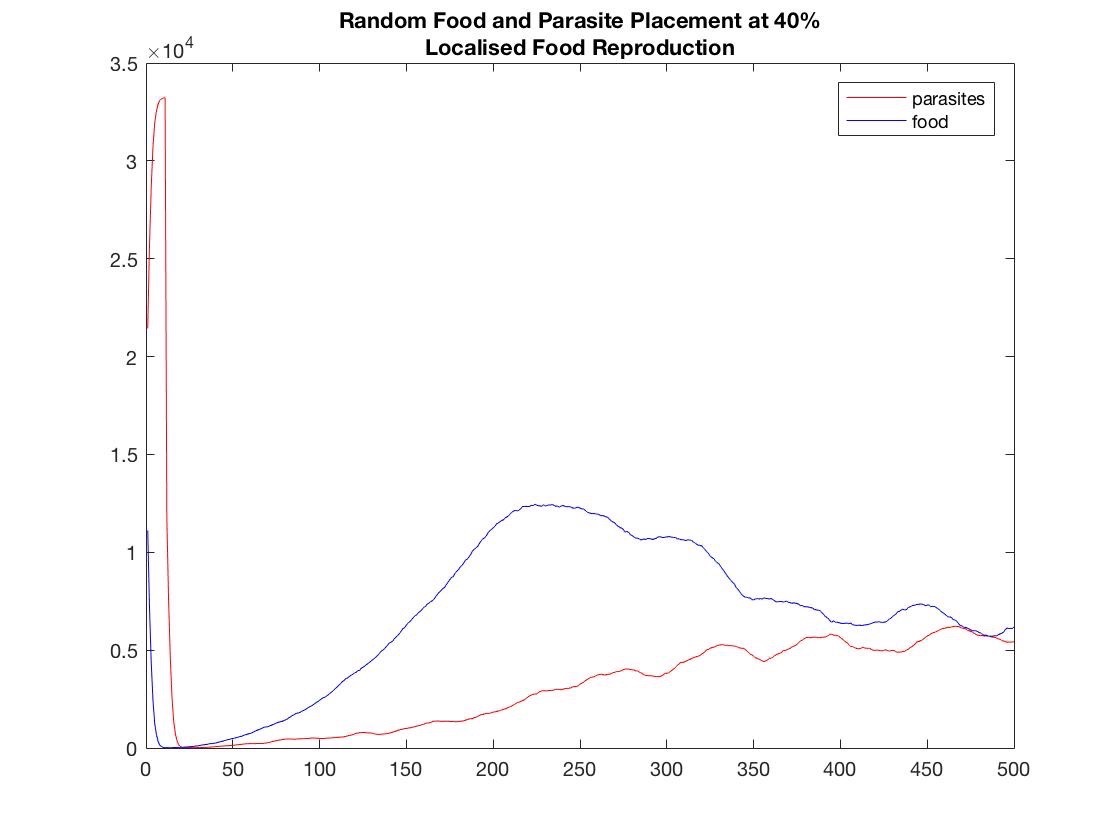
For 40% density, the parasites surround the food particles causing them to have limited reproduction space. Therefore a low parasite age of 10 steps is selected. The parasites have a large birth rate of 0.70 in order to survive and thrive. Refer to video ‘case45.avi’ for a simulation of this scenario.

For 30% density, the scenario had similar parameters to the 40% density case described above. To reach equilibrium the parasites had an age of 20 steps, food birth rate of 0.50 and a death rate of 0.10. Because this scenario is less dense than the 40% case, the parasites did not fully surround the localised food clusters which allowed the food to reproduce. The parasite age is twice as long as the 40% case because the parasites did not threaten to consume and restrict the food from growing. Refer to ‘case29.avi’ for the 30% density case.

For the 20% and 10% cases, there is significantly more space for the food to reproduce as they are not tightly constricted by the parasites. Therefore higher age limits are applied to the parasites as well as lower birth rates for the food. Refer to ‘case12.avi’ for the 20% density case and ‘case6.avi’ for the 10% density case.



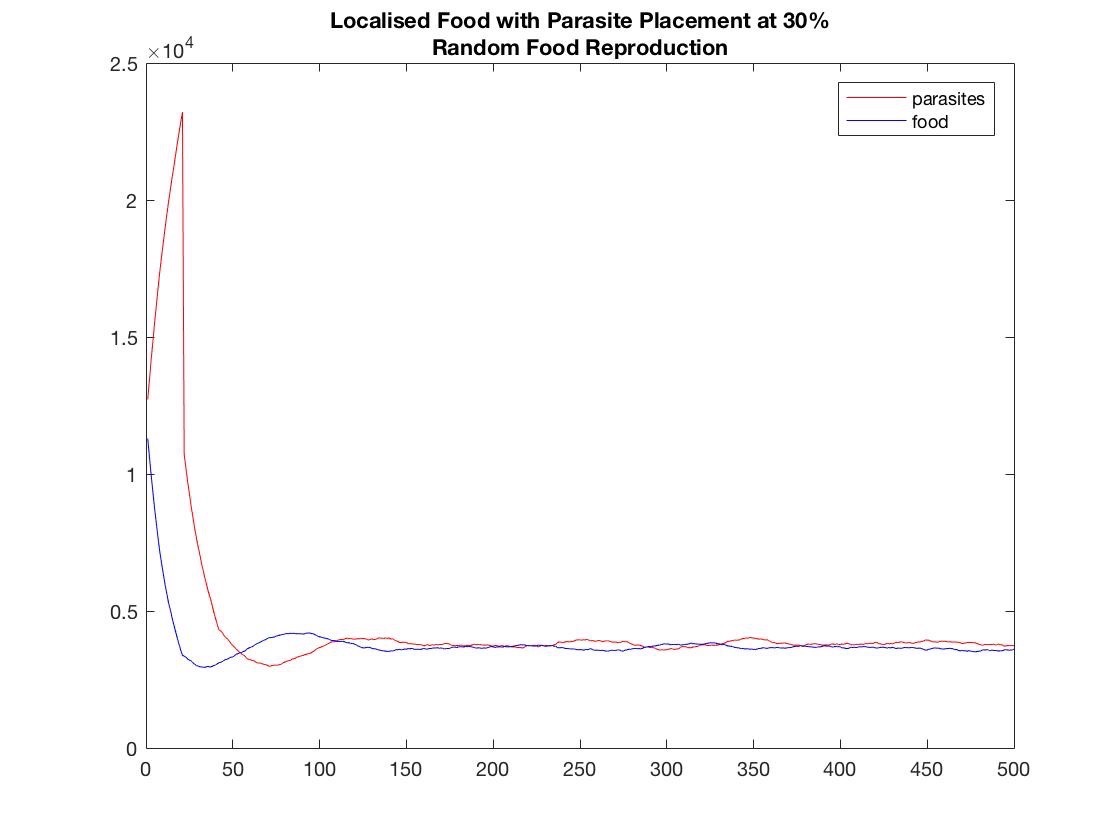
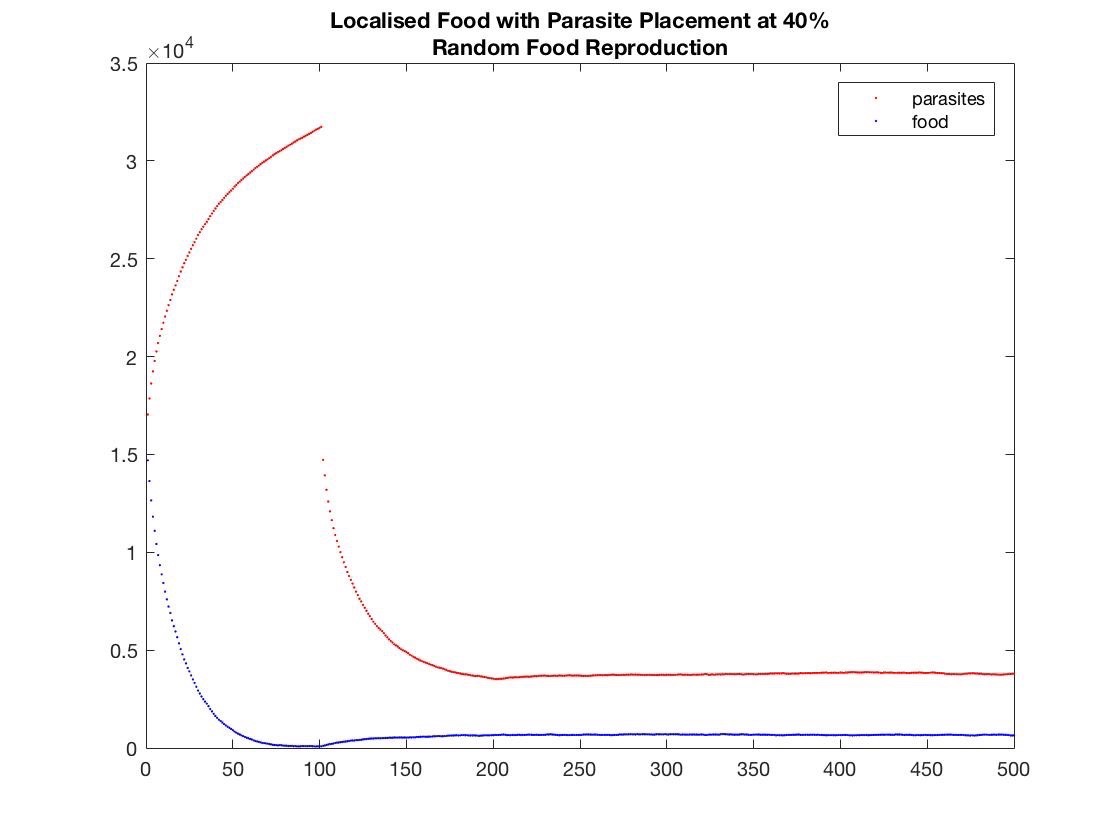
**Random food placement with localised food reproduction**

At 40% density, the food agents are not tightly packed in localised areas around the grid as was the case in the localised food placement scenario. Therefore the parasite maximum age was set to 20 steps, the reproduction rate of food is set to 0.50 and a food death rate of 0.05. The lower food birth rate and high parasite age limit was due to the food population not being threatened with extinction due to being surrounded by parasites.

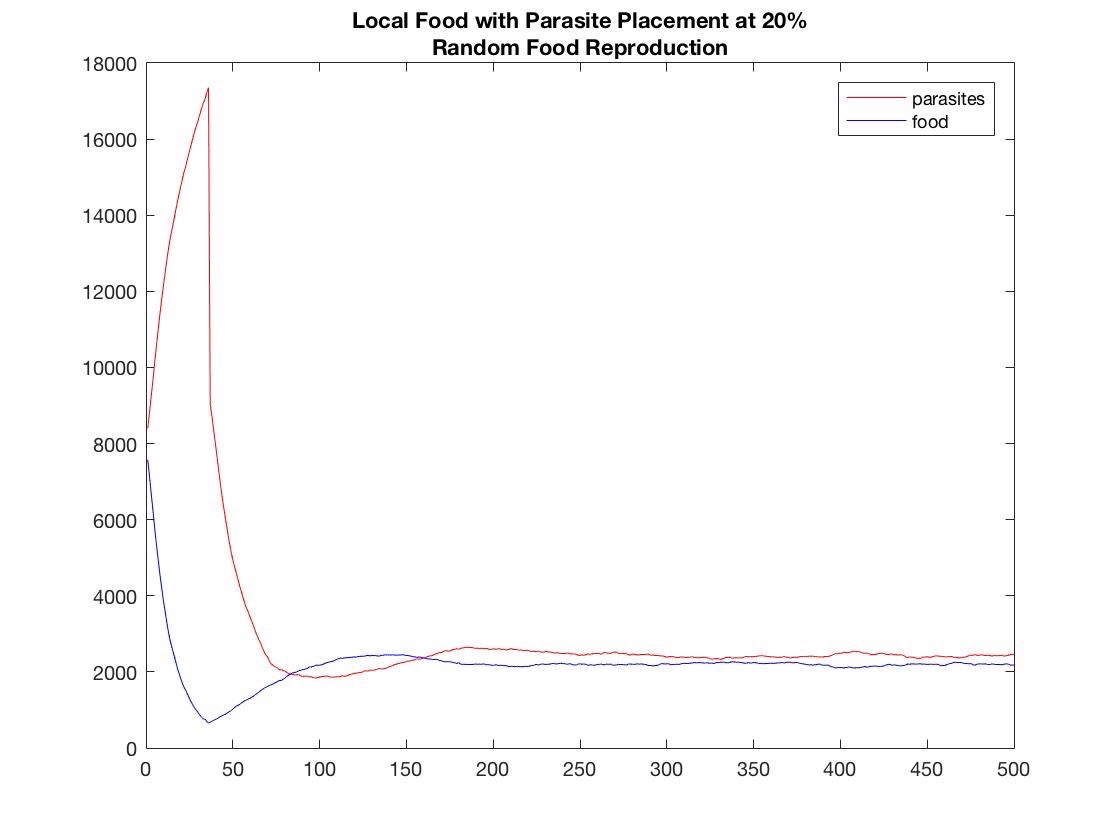
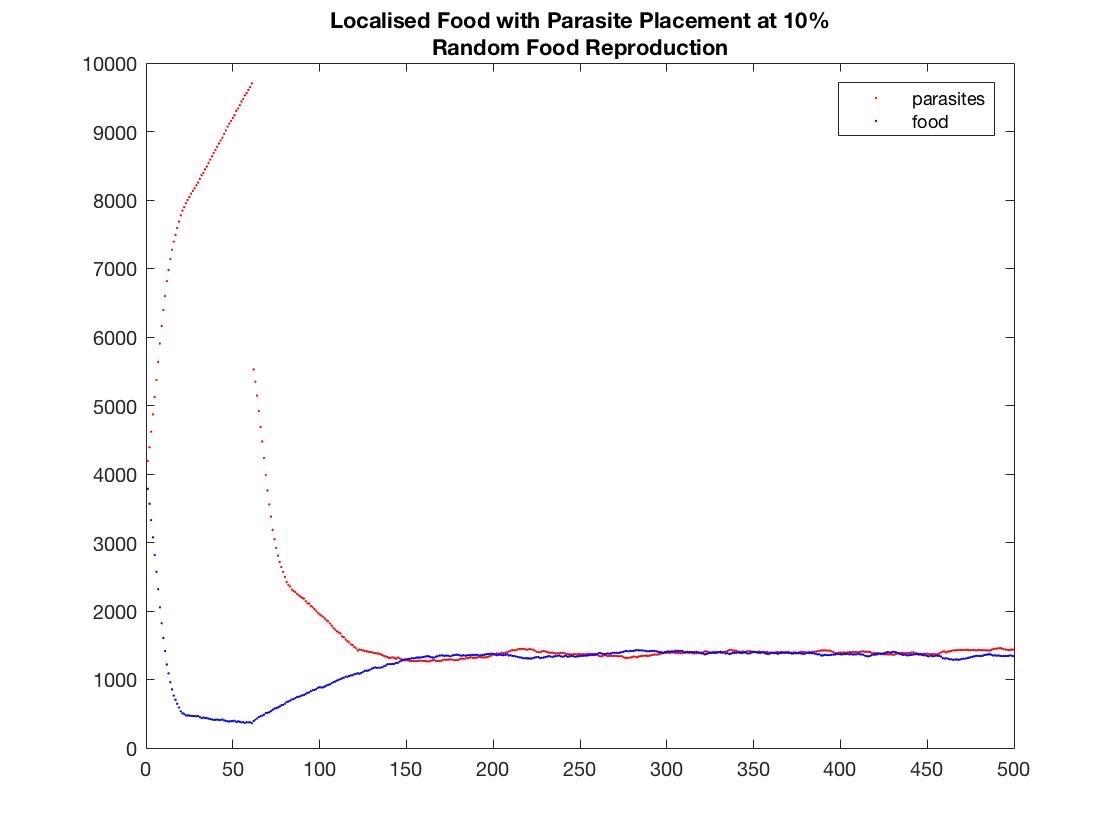
For 30% density the parameters for equilibrium are parasite age equaling 8, food reproduction equaling 0.50 and food death rate equaling 0.15. For this case, it was observed that the parasites quickly got out of control, therefore their age was limited to 8. For 20% density, the same parameters for 30% allowed equilibrium to be reached.

For 10% density, parasite age is set to 20, food creation set to 0.50 and food death is set to 0.02. Due to the lower populations of food, the parasites need to travel further in order to reproduce. Therefore the parasite age is set to 20 in order to reach equilibrium. Refer to ‘case2.avi’ and ‘case9.avi’ for 10% and 20% density cases.

**Localised food placement with random food reproduction**

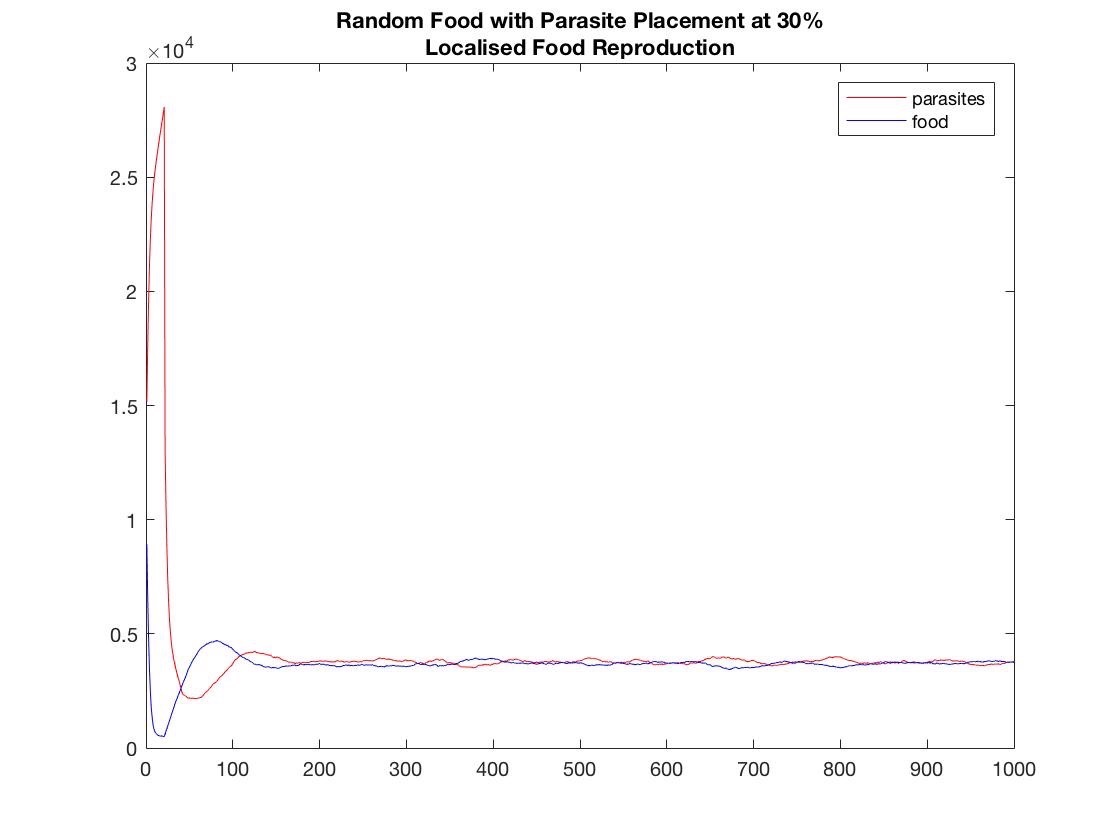
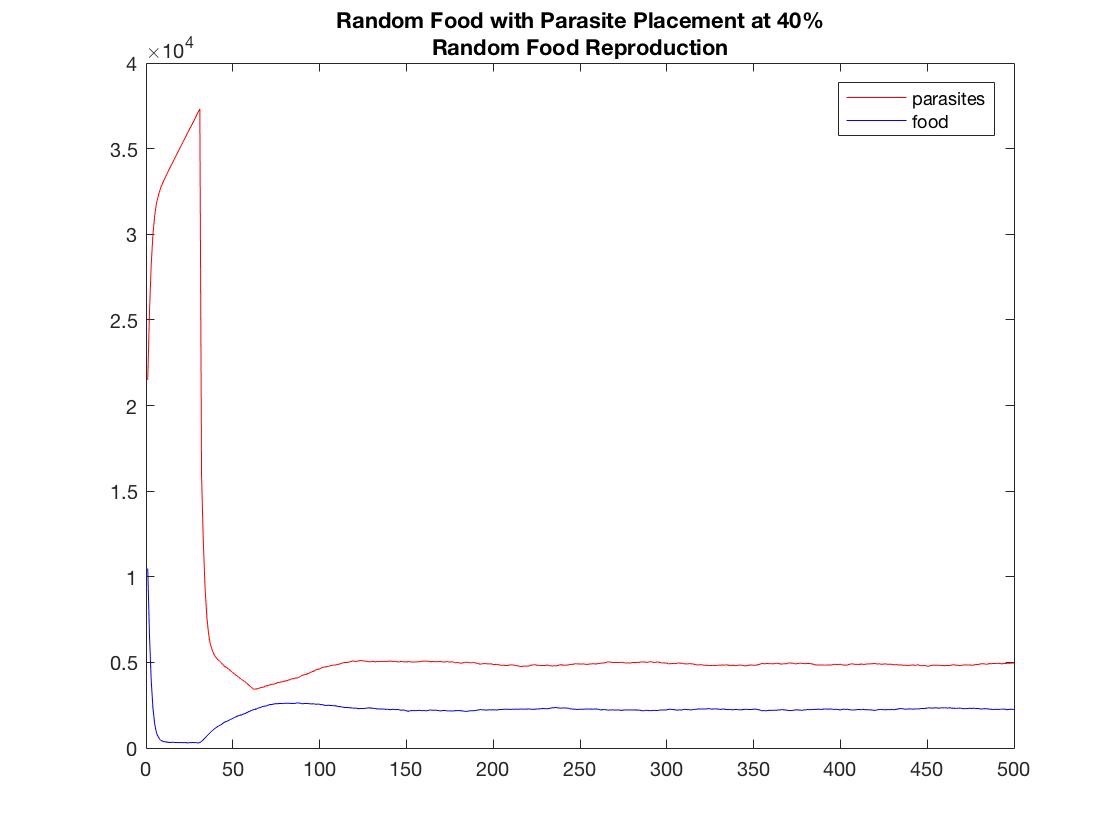
At 40% density, although the food agents were tightly bounded by parasites. They were able to easily replicate into random locations. Parasite age is set to 100 steps, food creation number is set to 50 and food death rate is set to 0.02. If the death rate is increased there would not be enough food to sustain the parasitic population. Refer to video ‘case38.avi’ for this scenario.

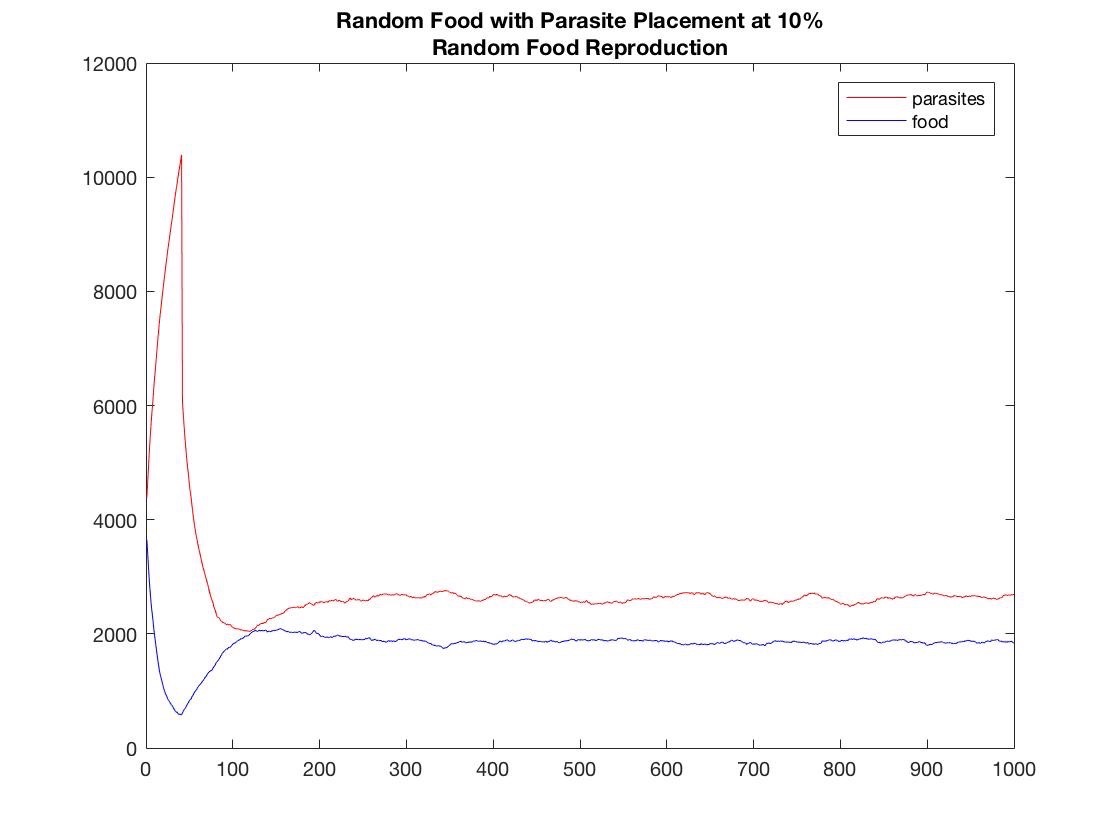
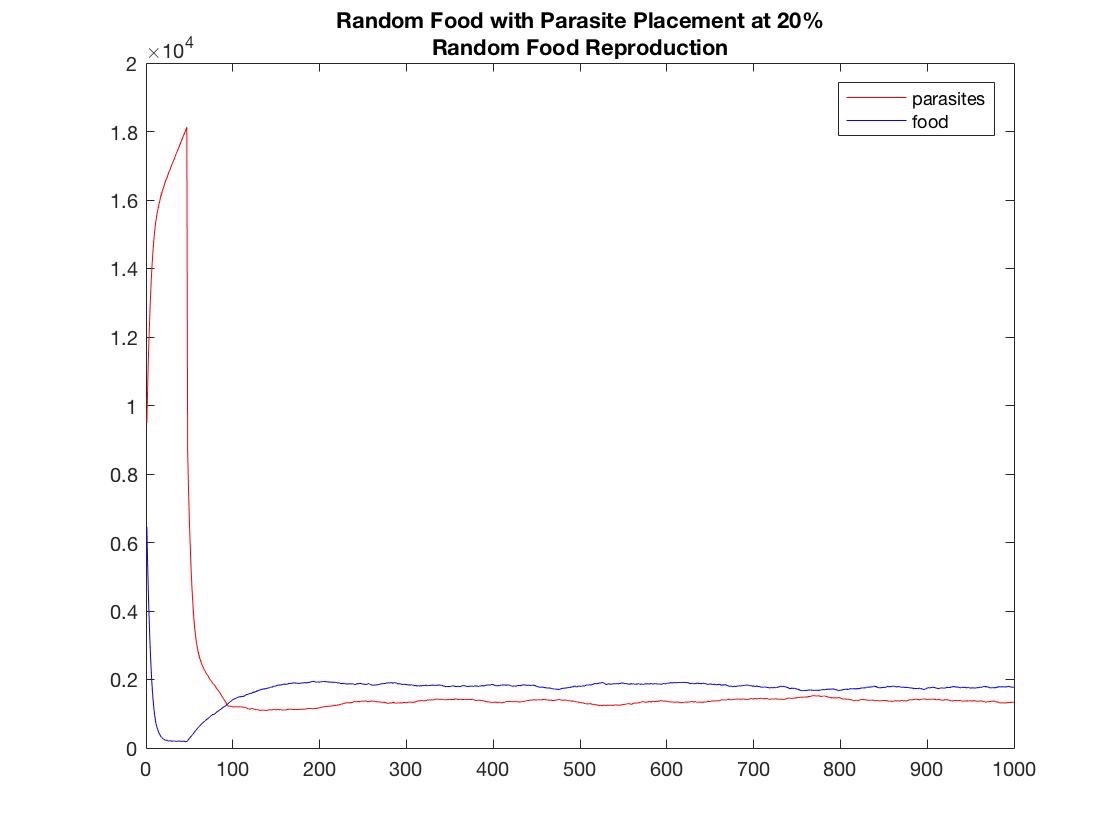
At 30% density, equilibrium is reached by setting parasite age to 100, food creation number to 100 and food death threshold to 0.02. Equilibrium is easily reached by reducing both populations to suitable levels. The 20% and 10% cases are similar to the 30% case.



For random food production, equilibrium is easily achieved as food cannot go extinct due to the constant number of food being created at each iteration. The constant number of food created determines the equilibrium solution of the simulation. Larger food reproduction will support larger parasitic populations. If the food is sparely populated due to a low reproduction number, the parasite age must be increased in order to reach the food and sustain their population.

**Random food placement with random food reproduction**

For the 40% density case, the parasites will quickly consume the food clusters which will create an imbalanced amount of parasites. The parasites population will decline shortly after however they will not go extinct due to the constant number of food being produced at random locations in the grid. It is evident that the equilibrium population levels can be altered by changing the amount of food reproduced at each iteration. The parasite age is set to 100, Refer to video ‘case38.avi’.



**Conclusion**

Continuous vs discrete models

speed of continuous model

Experienced long simulation times upwards of 15 minutes for the discrete model at 30% and 40% densities

Discrete model allows for minor changes to the system variables

e.g Reduce the food reproduction by 1% while keeping all other system variables constant

For random food creation

Easier to reach equilibrium for random food creation

Local placement of food does not affect the result much for random food creation

Densities of populations does not impede on reaching equilibrium

Due to constant amount of food agents created at each iteration, therefore food can never go extinct

Local food creation

Difficult to reach equilibrium at higher densities

In the local placement of food case, the food may be totally surround by parasites especially at higher densities. Therefore the parasites must quickly die off to provide space to allow the food population to grow.