MXB261 – Modelling and Simulation Science

Assignment 2 – A Simulation Project

Due: 24th October, Semester 2, 2018

Group 6 – Statement of Contributions:

Name	Student Number	Individual Task	Group Contribution
Free Sobolewski	N9947108	Task 1	25%
Riva Mendoza	N9941215	Task 2	25%
Saskia Mathers	N9945938	Task 3	25%
Jacinta Roberts	N9954619	Task 4	25%

Table of Contents

1.0 Introduction

The field of population dynamics provides valuable insight into parameters that impact the evolution, characteristics and distribution of populations. Most famously, is the Lotka-Volterra model, which captures the relationship between antagonistic species such as predator-prey systems through pairs of dependent first-order nonlinear differential equations. This extends quite fluidly to host-parasite systems, which were analysed in this report.

The model considered the interactions between a parasite and host species, and was based upon the following system of differential equations:

$$\frac{dX_1}{dt} = X_1(k_1X_2 - k_2)$$
$$\frac{dX_2}{dt} = k_3 - k_4X_2 - k_5X_1$$

In this system, X_1 represented the population of parasites, whilst X_2 represented the host population. Parameters k_1 through to k_5 were representative of the rates of birth rate of X_1, death rate of X_1, growth of X_2's food source, decay of X_2's food source, and the rate of host consumption (X_2) by the parasites (X_1) respectively. By examining these equations, the overarching behaviours were deduced, which showed that with plentiful food (i.e. hosts) the parasites would increase, and with too many parasites and too little food, the hosts would decrease.

Holistically, the motivation for the tasks was to study the effects parameter values (i.e. birth rates, death rates, immigration, life expectancy) had on the temporal and spatial system dynamics of a parasite model. Tasks One through Three utilised parameter sweeps of one and two variables, and Latin Hypercube Sampling of three and four variables to efficiently analyse the behaviour of the deterministic system. Task Four extended this by employing a spatial agent-based model to explore species interactions on an individual level which considered movement as stochastic.

This report provided relevant background information and motivations for the four tasks investigated. Then, the methods for reproducibility and concise results for each task were presented in sections two to five. Following this, cohesive links were ascertained by effectively comparing individual tasks in the discussion, and the conclusion summarised the main findings of the report.

2.0 Task One

2.1 Methods

In this task, the system of differential equations was analysed over the time span [0,20] with fixed parameters $k_1 = 1$ and $k_2 = 2$, and the initial conditions $X_1(0) = 1$ and $X_2(0) = 1$. Thus, the system simplified to $\frac{dX_1}{dt} = X_1(X_2 - 2)$ and $\frac{dX_2}{dt} = k_3 - k_4X_2 - k_5X_1$.

Given these equations, the system equilibrium was identified by setting $\frac{dX_1}{dt} = \frac{dX_2}{dt} = 0$.

$\frac{dX_1}{dt} = 0 \ : \ X_1 = 0 \quad : \ \frac{dX_2}{dt} = k_3 - k_4X_2 \quad : \ \frac{dX_2}{dt} = 0$ when $X_2 = \frac{k_3}{k_4}$

$\frac{dX_1}{dt} = 0 \ : \ X_2 = 2 \quad : \ \frac{dX_2}{dt} = k_3 - 2k_4 - k_5X_1 \ : \ \frac{dX_2}{dt} = 0$ when $X_1 = \frac{k_3 - 2k_4}{k_5}$

$\frac{dX_2}{dt} = 0 \ : \ X_2 = \frac{k_3 - k_5X_1}{k_4} \ : \ \frac{dX_1}{dt} = X_1\left(\frac{k_3 - k_5X_1}{k_4} - 2\right) \ : \ \frac{dX_1}{dt} = 0$ when $X_1 = 0$ or $X_1 = \frac{k_3 - 2k_4}{k_5}$

$\frac{dX_2}{dt} = 0 \ : \ X_1 = \frac{k_3 - k_4X_2}{k_5} \ : \ \frac{dX_1}{dt} = \frac{k_3 - k_4X_2}{3}(X_2 - 2) \ : \ \frac{dX_1}{dt} = 0$ when $X_2 = 2$ or $X_2 = \frac{k_3}{k_4}$

Therefore, the equilibrium points were found at $(\overline{X_1}, \overline{X_2}) = \left(0, \frac{k_3}{k_4}\right)$ and $(\overline{X_1}, \overline{X_2}) = \left(\frac{k_3 - 2k_4}{k_5}, 2\right)$.

This task investigated the effects of:

(a) Varying the rate of growth of the host populations food source (i.e. k_3) on [0,50], while holding all other parameters constant such that $k_1 = 1$, $k_2 = 2$, $k_4 = 4$, and $k_5 = 3$.

(b) Varying the rates of growth and decay of the host populations food source (k_3 and k_4) on [0,50], whilst holding all other parameters constant such that $k_1 = 1$, $k_2 = 2$, and $k_5 = 3$.

(c) Varying the decay of the host populations food source and consumption of the host population by the parasite population (i.e. k_4 and k_5) on [0,50], whilst holding all other parameters constant such that $k_1 = 1$, $k_2 = 2$, and $k_3 = 10$.

Parameter values were considered successful if they maintained non-negative populations at all times on [0,20] and yielded solutions that tended toward either of the steady state populations within specified tolerances, i.e. $X_1 \to 0 + \text{Tol}$ or $X_2 \to 2 \pm \text{Tol}$ for $\text{Tol} = 10^{-1}$ and $\text{Tol} = 10^{-2}$.

The system defined in Section 1.2 was solved using the built-in MATLAB function 'ode45', in combination with the custom function 'ParasiteModelFn.m'. This function defined the model given the specified time points t, corresponding population vector $X = [X_1, X_2]^T$ and parameters k_1 through to k_5. In addition to this, 'ParasiteModelFn.m' was used to define the Parasite Model as a function handle in terms of X and t. Passing this into 'ode45', along with a vector defining the time span and an initial conditions vector, the system may be solved as follows:

```
[t,X] = ode45(@(t,X)ParasiteModelFn(t,X,k1,k2,k3,k4,k5), tspan, X0);
```

Three additional functions were written to perform parameter sweeps of one and two variables.

2.1.1 One Parameter Sweep

'OneParameterSweep.m' performed a parameter sweep for potential values of k_3. It required input values for tolerance, bounds on the range of k_3, an increment to step from the lower bound to the upper bound of the range of k_3, the initial conditions of X_1 and X_2 and fixed parameter values for k_1, k_2, k_4, and k_5.

In addition to this, a vector containing the potential values of k_3 was generated using the 'linspace' function which created an empty two column array of equal size in which successful parameter values were recorded. After the initialisation was complete, a for-loop was introduced which tested each k_3 value. This is done by calculating the system solution for each k_3 value in the outer for-loop, using 'ode45' and 'ParasiteModelFn' as explained above, and employing an if-statement which records the k_3 values if they met the necessary criteria (i.e. 'successful' k_3 values).

2.1.2 Two Parameter Sweep

'TwoParameterSweepk3k4.m' performed a parameter sweep for possible combinations of k_3 and k_4. The inputs included: a tolerance (i.e. Tol such that $X_1 \to 0 + \text{Tol}$ or $X_2 \to 2 \pm \text{Tol}$), bounds on the potential range of k_3 and k_4, an increment used to step from the lower bound to the upper bound of the range of k_3 and k_4, the initial conditions of X_1 and X_2, and fixed parameter values for k_1, k_2, and k_5. The initialisation process was very similar to 'OneParameterSweep.m' except this time, a three column array the size of the square of the size of the potential values vector was defined to store the successful parameter pairs.

Following this, two nested for-loops were used to loop k_3 and k_4 values to test all possible combinations. Then, the system solution for each k_3 and k_4 combination was determined using 'ode45' and 'ParasiteModelFn' and if-statements were utilised to test for the necessary criteria. The output of this function was a three-column array which contained each successful pair, with k_3 in the first column, k_4 in the second column, and the coded 'success' criteria values in the third column. Finally, 'TwoParameterSweepk4k5.m' was identical to this function but instead, k_4 and k_5 parameters were tested and k_3 was a constant input in place of k_5.

2.2 Results

2.2.1 One Parameter Sweep

To complete this task and determine the influence the rate of growth of the host populations food source (i.e. k_3) had over the system, 'OneParameterSweep.m' was called twice; the first time using a tolerance of 10^{-1}, and the second using a tolerance of 10^{-2}. The potential values of k_3 were considered to be all those on [0,50] at increments of 0.5 (i.e. $0, 0.5, \dots, 49.5, 50$), and all other parameters were fixed such that $k_1 = 1$, $k_2 = 2$, $k_4 = 4$, and $k_5 = 3$. An increment of 0.5 for k_3 was selected as it obtained a substantial number of samples but was not too computationally expensive. After generating the two 'successful' parameter vectors, for $\text{Tol} = 10^{-1}$ and $\text{Tol} = 10^{-2}$, the values were plotted to provide a visual representation of which values of k_3 met each of the 'success' criteria.

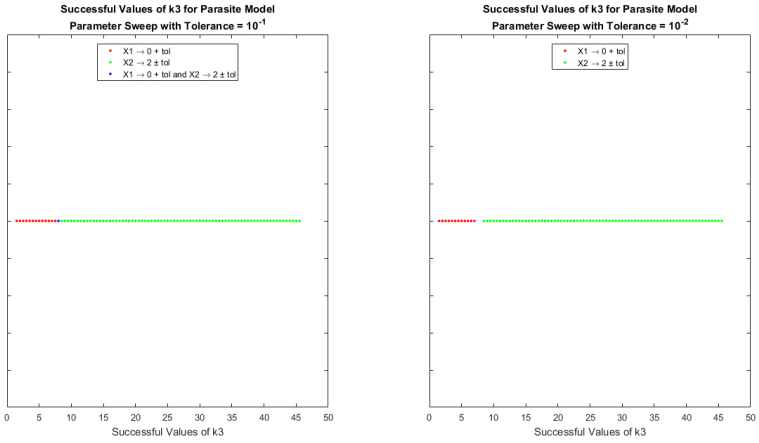


Figure 1: Successful values of k_3, colour coded to indicate 'success' criteria

For $\text{Tol} = 10^{-1}$ (left plot) the 'successful' k_3 values were found to be $1.5 \leq k_3 \leq 45.5$. Similarly, for $\text{Tol} = 10^{-2}$ (right plot) the 'successful' k_3 values were found to be $1.5 \leq k_3 \leq 7$ and $8.5 \leq k_3 \leq 45.5$ (Figure 1).

2.2.2 Two Parameter Sweep (k_3, k_4)

'TwoParameterSweepk3k4.m' was called twice; using a similar process. The parameters, k_3 and k_4, were investigated on [0,50] at increments of 0.5. After generating the two 'successful' parameter arrays for $\text{Tol} = 10^{-1}$ and 10^{-2}, the pairs were plotted on a scatter plot (Figure 2).

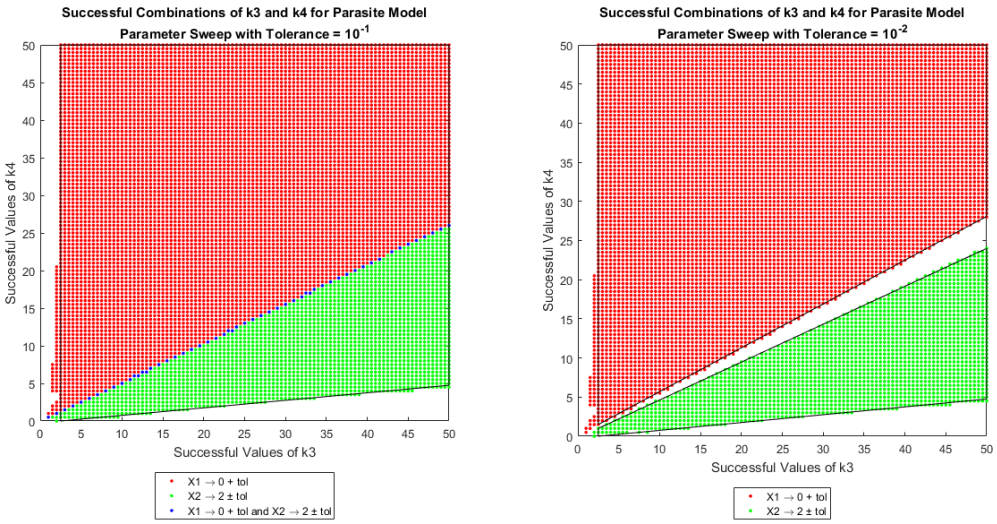


Figure 2: Successful pairs of (k_3, k_4), with additional lines marking defined inequalities

Figure 2 highlighted the successful (k_3, k_4) pairs for Tol $= 10^{-1}$ (left) were bounded by $\frac{1}{10}(k_3) - \frac{1}{4} \leq k_4 \leq 50$ on $\frac{5}{2} \leq k_3 \leq 50$.
For Tol $= 10^{-2}$ (right) the 'successful' pairs were from $\frac{1}{10}(k_3) - \frac{1}{4} \leq k_4 \leq \frac{46}{95}(k_3) - \frac{4}{19}$ and $\frac{14}{25}(k_3) + \frac{1}{10} \leq k_4 \leq 50$ on $\frac{5}{2} \leq k_3 \leq 50$.

2.2.3 Two Parameter Sweep (k_4, k_5)

To determine the influence the rates decay of the host populations food source (k_4) and consumption of the host population by the parasites (k_5), 'TwoParameterSweepk4k5.m' was called twice to generate the following results. All aspects of the previous parameter sweep were maintained, however, k_4 was varied in place of k_3, and k_5 was varied in place of k_4 (Figure 3).

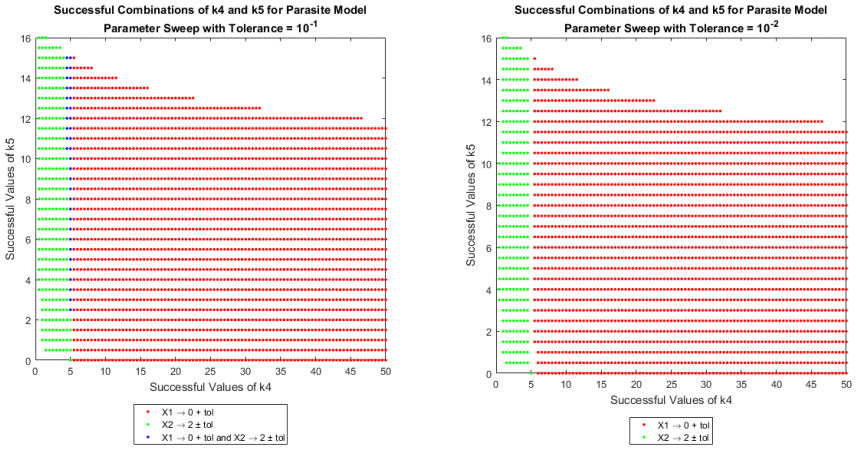


Figure 3: Successful pairs of (k_4, k_5), colour-coded to show 'successful' pairs

For Tol $= 10^{-1}$ (left) the 'successful' (k_4, k_5) pairs were bounded by $\frac{1}{18}(k_4)^2 - \frac{13}{12}(k_4) + \frac{145}{36} \leq k_5 \leq \frac{9}{5000}(k_4)^2 - \frac{9}{50}(k_4) + 16$ on $0 \leq k_4 \leq 5$, and $0 \leq k_5 \leq \frac{9}{5000}(k_4)^2 - \frac{9}{50}(k_4) + 16$ on $5 \leq k_4 \leq 50$. This region was similar for Tol $= 10^{-2}$ (right), but with a defined rift around $k_4 = 5$.

3.0 Task Two

3.1 Methods

This task was a natural extension of the first, as the same effects were investigated while the parameters were fixed such that $k_1 = 1$ and $k_2 = 2$. Successful parameter values were those that maintained non-negative populations on [0,20], (i.e. $X_1, X_2 \geq 0$) and those which yielded solutions that tended toward either of the steady state populations within specified tolerances ($X_1 \to 0 + \text{Tol}$ or $X_2 \to 2 \pm \text{Tol}$ for $\text{Tol} = 10^{-1}$). To clarify, $\text{Tol} = 10^{-2}$ was not considered in this instance to avoid over complication, though the methods used could be easily extended in future investigations to examine such scenarios.

If the methodology in Task One was employed here, three dimensions would require the trial of 101^3 combinations of values, which was too computationally expensive. Thus, the Latin Hypercube Sampling method was instead implemented to build a population of 'successful' 3-tuples, i.e. (k_3, k_4, k_5) that could be used to determine a bounded region of parameter values that yielded the desired long-term system behaviours.

To implement this, 'LHS3D.m' was written to perform parameter sweeps of three variables using Latin Hypercube Sampling. This function required: input values for the tolerance (Tol such that $X_1 \to 0 + \text{Tol}$ or $X_2 \to 2 \pm \text{Tol}$); bounds on the potential range of k_3, k_4, and k_5; an increment used to step from the lower bound to the upper bound of the range of k_3, k_4, and k_5; the initial conditions of X_1 and X_2; and fixed parameter values for k_1 and k_2.

After passing this information into the function, the vectors $tspan$ and X_0 were defined for later use in 'ode45'. Additionally, an array containing potential 3-tuples was generated in accordance with the Latin Hypercube Sampling Process. A $(n \times n)$ Latin Square in 2D space existed when there was only one sample in each row and column. Extending this to 3D, the function LHS3D defined k_3 as a vector containing all its potential values using MATLAB's built-in function 'linspace' based on the bounds and increment specified in the input. The corresponding k_4 and k_5 values were then generated by randomly sampling k_3 without replacement. Finally, the four-column array was defined with columns one to three containing k_3, k_4, and k_5, and column four containing coded values to indicate whether the 3-tuple was 'successful' in accordance with the specified criteria. These coded values are set to 0 by default (i.e. 'unsuccessful').

After these variables were initialised, a for-loop was introduced which iterated through each of the potential 3-tuples. The if-statements overwrote the coded values with ones for all combinations that were classified as 'successful'.

The output of this function was the four-column array where each tested 3-tuple of parameter values were recorded, storing k_3 in the first column, k_4 in the second column, k_5 in the third column, and the coded success/failure values in the fourth column. To construct a considerable population of 'successful' tuples, the results from several runs were compiled and recorded.

3.2 Results

To complete this task, 'LHS3D.m' was called 100 times using a for-loop, solving the system on a time span of [0,20], with the initial conditions $X_1(0) = 1$ and $X_2(0) = 1$, and parameters $k_1 = 1$ and $k_2 = 2$ fixed. The potential values of k_3, k_4, and k_5 were considered to be all those on [0,50] at increments of 0.5 (i.e. $0, 0.5, \dots, 49.5, 50$). 0.5 was the chosen step value as with Task One, as it was found to be a suitable median between sampling the continuous domain at a substantial number of equally spaced points and not becoming too computationally expensive.

Within the loop that executed 'LHS3D.m' 100 times, all the results were recorded in a single array, and all unnecessary/repeated rows were removed. After generating this array, the 3-tuples were plotted in 3D scatter plots, colour coded to indicate which combinations were 'successful'.

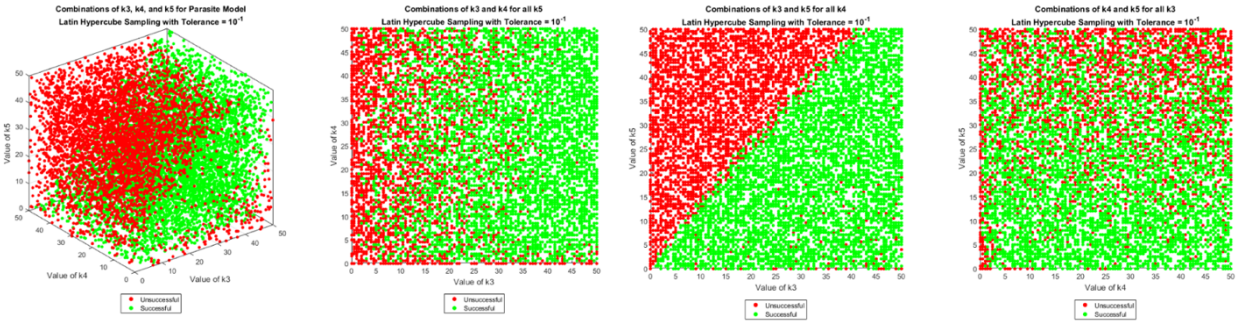


Figure 4: Plot of the results of 100 LHS trials on (k_3, k_4, k_5).

In these plots, clear relationships between the variables were exhibited. Most distinctly among these could be seen in the lower left plot of k_3 and k_5, where 'successful' points and 'unsuccessful' points were clearly divided by a positive linear line at approximately $k_5 = \alpha k_3$ for some α between 1 and 1.5.

By approximating $\alpha = 1.1$ and removing all points wherein $k_5 \geq \left(\frac{11}{10}\right) k_3$, the vast majority of 'unsuccessful' combinations were eliminated. This provides considerable clarity with regards to further inequalities that may be imposed to eliminate the remaining 'unsuccessful' points.

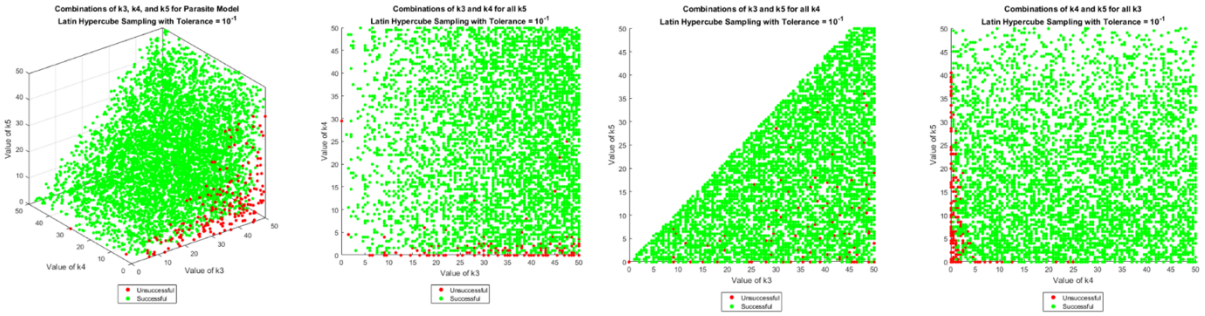


Figure 5: Plot of the results of 100 LHS trials on (k_3, k_4, k_5), where $k_5 \leq \left(\frac{11}{10}\right) k_3$.

In these plots, there was a vague correlation between the k_3 and k_4 parameters in the upper right, and the k_4 and k_5 parameters in the lower right. Directing attention first to the second plot of k_3 and k_4, 'successful' points and 'unsuccessful' points appeared to be divided by a positive linear line at approximately $k_4 = q + pk_3$ for some q between 0 and 10, and p between 0 and 0.5. Next, the fourth plot of k_4 and k_5 appeared to divide 'successful' points and 'unsuccessful' points around the negative linear line at approximately $k_5 = m + nk_4$ for some m between 0 and 10, and n between -1 and 0.

Given these three approximate inequalities, a loop was written in MATLAB that tested each combination of the coefficients on the intervals estimated above. Within this loop, the coefficient combinations were recorded along with the fraction of successes within the data set and the number of success observed post the application of the inequalities with the corresponding coefficients. From this information, the coefficient combination that yielded 100% successes (i.e. eliminated all unsuccessful points) but maintained the most observations (i.e. removed minimal successful points) were selected. The resultant inequalities were thus defined to be:

- $k_4 \geq \frac{1}{10}(k_3)$
- $k_5 \geq 3 - \frac{1}{10}(k_4)$
- $k_5 \leq \frac{10}{9}(k_3)$

Applying these inequalities to the data set, all 'unsuccessful' combinations were eliminated, and the number of 'successful' combinations eliminated was minimized, losing only 14.48% of the successes originally observed.

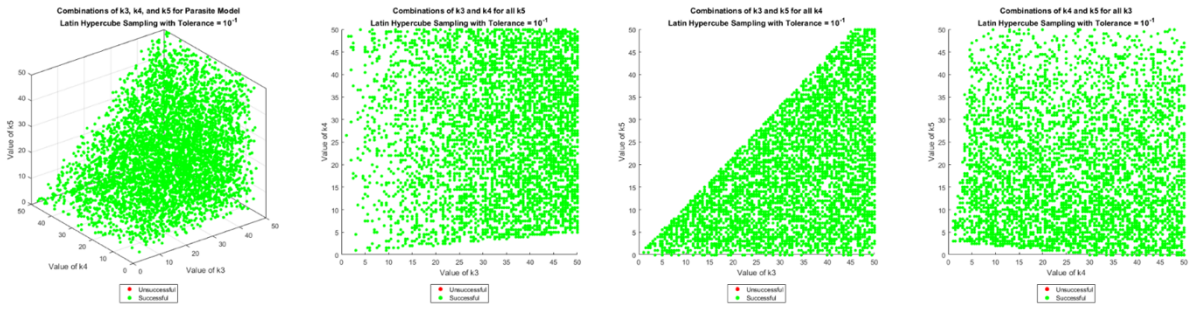


Figure 6: Plot of the results of 100 LHS trials on (k_3, k_4, k_5), with all inequalities applied.

Thus, it can be said that successful parameter 3-tuples lie in the region bounded by $k_4 \geq \frac{1}{10}(k_3)$ and $3 - \frac{1}{10}(k_4) \leq k_5 \leq \frac{10}{9}(k_3)$ on $0 \leq k_3 \leq 50$.

3.2.1 Extension

Given that the results of Latin Hypercube Sampling were dependent on the population of 'successful' 3-tuples generated accurately representing the true spatial distribution of these 'successful' combinations, a Three Parameter Sweep function similar to those written in Task One was written to test all $(101)^3$ combinations of (k_3, k_4, k_5) on [0,50] at increments of 0.5 to confirm the results. This was incredibly computationally expensive and took the best part of a day to execute, however the results were incredibly insightful.

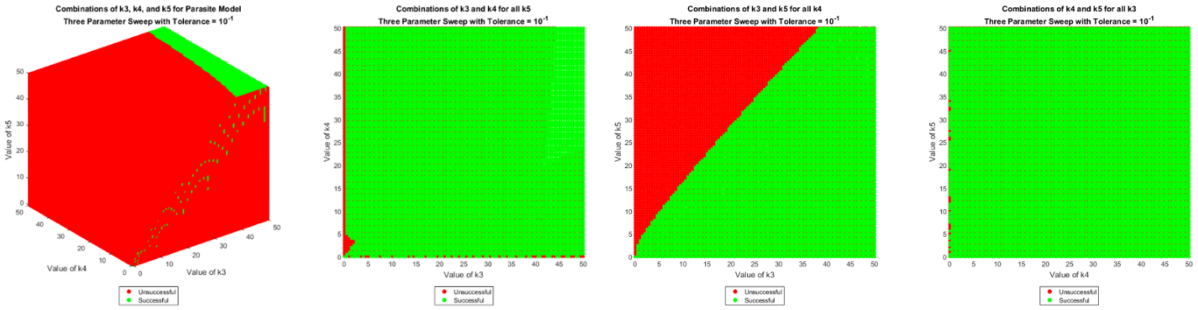


Figure 7: All combinations of (k_3, k_4, k_5), colour coded to indicate success/failure.

As with the LHS result, a clear relationship between the variables was observed. However, in this case it can also be seen that there was a layer of 'unsuccessful' points covering the 3D plot when k_3 and k_4 were equal to 0. In addition to this, when $k_5 \leq 1$, there was some erratic behaviour. While this could be explained using more complex inequalities, it was simpler to generalise and condition so that $k_5 \geq 1.5$, $k_3 \geq 0.5$, and $k_4 \geq 0.5$.

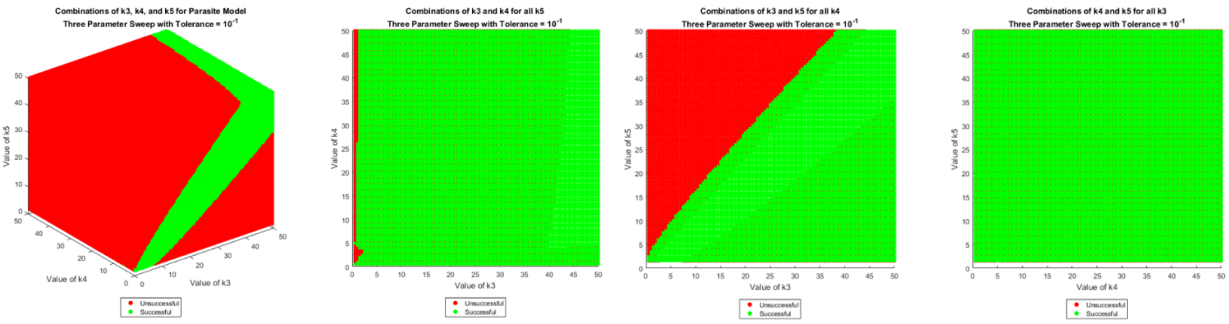


Figure 8: All combinations of (k_3, k_4, k_5), colour coded to indicate success/failure, where $k_5 \geq 1.5$, $k_3 \geq 0.5$, and $k_4 \geq 0.5$.

As seen above, once these conditions were applied to the parameters, the relationships become even clearer. Using similar logical processes as with the LHS it could be seen that the majority of 'unsuccessful' points lied in the regions where:

$k_5 \leq i + j(k_3)$ for $-\frac{1}{2} \leq i \leq 0$ and $1 \leq j \leq \frac{3}{2}$

$k_5 \geq m - n(k_4)$ for $35 \leq m \leq 45$ and $0 \leq n \leq 10$

Given these three approximate inequalities, a loop was written in MATLAB that tested each combination of the coefficients on the intervals estimated above. Within this loop, the coefficient combinations were recorded along with the fraction of successes within the data set and the number of success observed post the application of the inequalities with the corresponding coefficients. From this information, the coefficient combination that yielded 100% successes (i.e. eliminated all unsuccessful points) but maintained the most observations (i.e. removed minimal successful points) was selected. The resultant inequalities were thus defined to be:

$k_5 \leq \frac{10}{9}(k_3)$

$k_5 \geq 43 - 9(k_4)$

Applying these inequalities to the data set, all 'unsuccessful' combinations were eliminated, and the number of 'successful' combinations eliminated is minimized, losing only 14.55% of the successes originally observed.

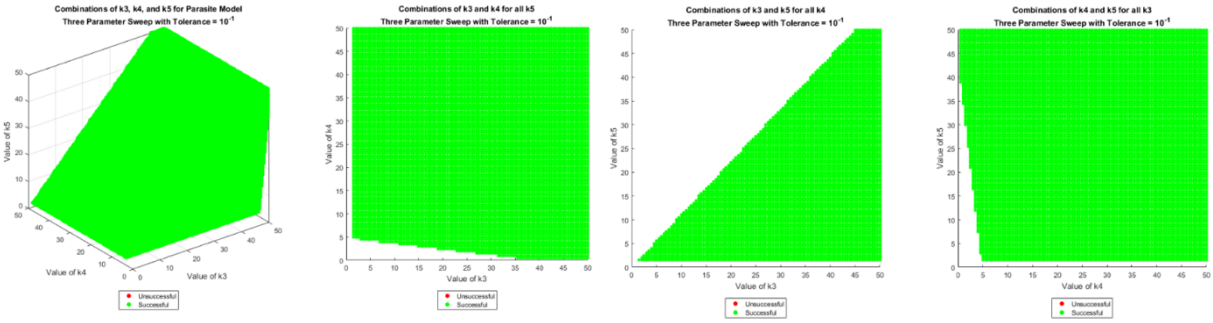


Figure 9: All combinations of (k_3, k_4, k_5), with inequalities applied.

Thus, it can be said that successful parameter 3-tuples lie in the region bounded by $43 - 9(k_4) \leq k_5 \leq \frac{10}{9}(k_3)$ for $0.5 \leq k_3 \leq 50$, $0.5 \leq k_4 \leq 50$, and $1.5 \leq k_5 \leq 50$.

Note: Applying these inequalities to the data obtained using LHS it can be seen that 100% of unsuccessful points are still removed, but 15.22% of successful points are lost in the process. This can be attributed to the fact that LHS is not a flawless representation of the space in its entirety. However, it provided a decent estimate.

4.0 Task Three

4.1 Methods

Task Three was a further extension to Task Two, which applied Latin Hypercube Sampling over a 4D parameter space. 'LHS3D.m' was modified and saved as 'LHS4D.m' to generate a population of successful 4-tuples, instead of 3-tuples. The input variables were identical, requiring the tolerance value such that $X_1 \rightarrow 0 +$ Tol or $X_2 \rightarrow 2 \pm$ Tol (tol), the lower bound on the potential ranges of k_3, k_4, and k_5 (lb), the upper bound on the potential ranges of k_3, k_4, and k_5 (ub), the increment value stepping from the lower bound to the upper bound of the ranges of k_3, k_4, and k_5 (inc), the lower and upper bounds on the time span of the simulation (tl and tu), the initial conditions of X1 and X2 (X1i and X2i), the birth rate of X1 (k1) and the death rate of X1 (k2). When called, this function outputted an array (LHk) which contained all combinations of k_2, k_3, k_4, and k_5 in columns one to four using Latin Hypercube Sampling. Column five stored 1s and 0s to indicate whether the combination was 'successful'.

k_2 was generated using MATLAB's linspace and 4-tuples of k_3, k_4, and k_5 were then generated randomly using MATLAB's datasample function to randomly sample k_2 without replacement. Two vectors were also initialised: tspan was defined to store the time span given and X0 stored the initial conditions given.

A for-loop tested each 4-tuple combination and utilised MATLAB's ode45 and the 'ParasiteModelFn' to determine the system solution for that combination. An if-statement determined whether that solution was successful and updated the corresponding row in the fifth column to 1 if so.

This function was called 100 times in the script 'LHS4D_Task03.m' to obtain 101×100 4-tuples of successful and unsuccessful LHS combinations. These were stored in the array called LHkIterated. Duplicate rows were removed using MATLAB's unique function and the combinations were sorted by the first column, then second, then third, etc.

4.2 Results

This 4-dimensional data was modelled in two ways, using the idea of the plotmatrix function and using a 3D scatter plot similar to Task Two. Figure 10 was plotted using scatter in nested for loops, which reveal comparisons between each parameter in a 2D space. The 3D scatter plot was modelled from Task Two's scatter plot, plotting k_2, k_3 and k_4 in the x, y, z axes, in addition to visualising k_5, in a spectrum of colour from dark blue (0) to yellow (50).

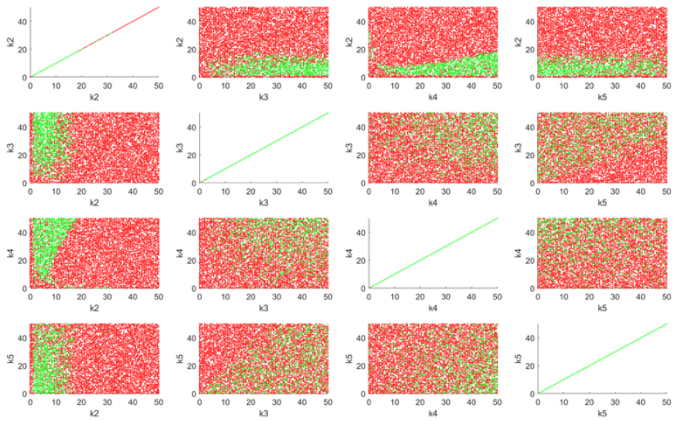


Figure 10: Comparison of k_2, k_3, k_4 and k_5 variables in combined scatter plots. Red depicts **unsuccessful** models, and green for **successful**.

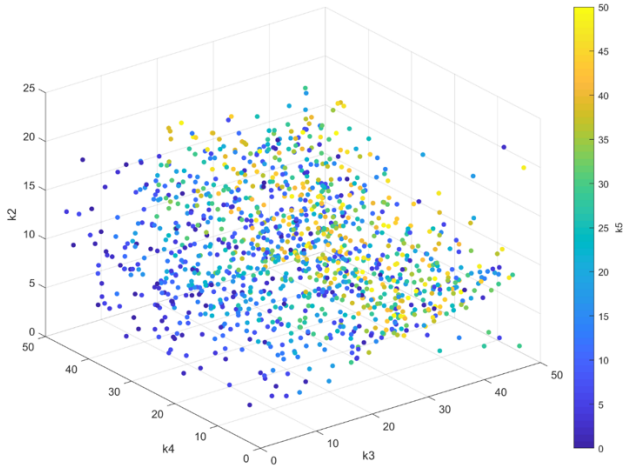


Figure 11: Successful models with k_2, k_3 and k_4 plotted in a 3D scatter plot, with colour corresponding to k_5.

Figure 10 and 11 revealed a distinct correlation between the parameters. Furthermore, it was shown that more successes occurred when the rate of food growth (k_3) and the rate of food consumption (k_5) were high. Thus, higher rates of food consumption occurred when there were greater amounts of food. Further, the rate of food decay (k_4) and the death rate of parasites (k_2) were closely related in that the lower the rate of food decay, the lower the rate in which parasites died. Figure 10 provided clear evidence of this, visualising the correlation between these two parameters which indicated that the higher the rate of food decay, the greater the parasite death rate. The domain of this death rate distinctly ranged from $\sim$1 to $\sim$17, with only one outlier greater than 20. This was seen clearly on both figures, particularly Figure 11. However, Figure 10 revealed that unsuccessful models occurred when the death rate was too low, suggesting that too many parasites would cause the model to fail - likely due to the availability of food. Thus, the highest success rates tended towards the highest food growth, lowest food decay, and lowest parasite death rates.

5.0 Task Four

5.1 Methods

The spatial-agent based approach of the parasite model explored in task four showcased the species' interactions at the individual level by simulating a random walk in MATLAB. To improve the code readability and aid in the troubleshooting process, four different files were created. The outermost layer 'MXB261_Task_4.m', called the 'SpatialAgentWalk' function with various combinations of parameters to ascertain the results required for analysis. It was decided that a 200x200 matrix would be used as the grid, with the food agents represented by 1's (green), parasites denoted by 2's (red) and empty cells were 0 (white).

'SpatialAgentWalkSetUp.m' populated the grid with food and parasite agents using the initial population specified by population density (PD). Parasites were placed randomly, and it was decided that the food should be placed all over the grid initially but when food growth occurred in 'Process' it would be placed according to the food-placement strategy investigated (pos). (The task did not explicitly specify whether the food had to be randomised or placed according to the strategy in the first frame, and to ensure that the code could place the desired population of food, the former approach was assumed.) To ensure that the food was positioned correctly, a while loop ensured that the space was not occupied to continually check the generated position before placing the food and increment the counter. A similar process was required for the parasites except their positions were always randomised. This ensured that the grid always began with the correct number of agents and none were overwritten. The outputs were the 200x200 matrix with initialised starting positions (InitialGrid), a Px2 vector (P_life) which stored how many iterations each parasite has undertaken and a Px3 vector (P_pos) which stored the position (parasite number, row, column) of each parasite – where 'P' represented the number of parasites.

The 'SpatialAgentWalkProcess.m' function performed one iteration of the simulation and output the new grid. There were four required steps with each iteration. In the first step, each adult parasite moved to an adjacent cell using a randomly generated direction (integer ranging from 1 to 4) and 1 of 3 scenarios occurred: 1. The new cell was empty, and it successfully moved, 2. The cell was occupied by another parasite and the move did not occur or 3. The new cell was occupied by food, which was consumed and a birth took place with the baby placed in the original cell.

The second step removed parasites which had surpassed their life-span. This was handled by incrementing every adult parasite, and then implementing a for loop to iterate through the 'P_life' matrix in conjunction with an if-statement to determine if the particular parasite was greater than or equal to the inputted life-span value, 'f_1'. If this condition was satisfied, the parasite was removed from the simulation. Then, the births from parasite movement were then appended to the end of the 'P_life' matrix.

Thirdly, the food agents were killed by sampling a uniform random distribution $u \sim U(0,1)$ (different for each agent) and checking if the value was less than the input variable 'f_2' (likelihood for food to die). This was implemented with a for loop to generate a random sample for each food agent and then remove the agent if it satisfied the conditional statement.

Finally, new food agents were added to the grid according to the specified amount (f_3) and position (pos). This was conducted in a similar fashion to when the grid was initially populated by initialising a counter (food_birth) and while loop. However, the new food position (row, column) was now randomly generated according to 'pos' (x_width, x_offset, y_width, y_offset). If this coordinate was empty (0), then the food was immediately placed, and the counter was incremented. Alternatively, if the cell was occupied, a nearby cell was selected using a series of if-statements to check below, above, left and right of the current cell. If there were no possible places left for food growth, the 'break_counter' was incremented and if it exceeded an arbitrarily large value (10,000) then the food-placement while loop was broken (avoided stall).

'SpatialAgentWalk.m' simulated the parasite model by utilising the 'SetUp' and 'Process' functions and generated a .avi movie file to illustrate the evolution. Firstly, to capture the video, a VideoWriter object was initialised and a for loop was used to step the simulation according to 'no_frames'. To capture the successive frames, MATLAB's 'spy' function was used to display the current simulation state and 'agent_counter' recorded the food and parasite population at each frame. To save this collection of frames, MATLAB's 'writeVideo' function was employed at the end of the loop. Additionally, another figure was generated to depict the final frame alongside the parasite and food agent population over time.

5.2 Results

Each parameter: population density (PD), food-placement strategy (pos), parasite lifespan (f_1), food death-rate (f_2) and food births (f_3) were swept through a range to identify key contributors that affected the system dynamics (Appendix 3). This investigation also highlighted the greater impact of f_1, f_2 and f_3 compared to the population density and food-placement strategy on the model behaviour. From this, a control was determined which was close to the median value, which made the effects of altering each parameter easily measured. Each sweep iteration was recorded, and the control values were selected as: population density = 0.2 (20%), $f_1 = 10$, $f_2 = 0.02$, $f_3 = 300$. With these parameters, the model was simulated using 150 frames, the randomised food-placement strategy ([200,0,200,0]) and initial population density was varied (Figure 12). The '.avi' and '.fig' files attached showcased the results from the report, however, if the simulation were to be run again, slightly different results would be obtained due to stochasticity. This could be simulated by running lines 14-17 from 'MXB261_Task_4.m' or calling the following line and varying the 3rd parameter from 0.1 and the filename.

```
SpatialAgentWalk('pop10%.avi',150,0.1,[200,0,200,0],10,0.02,300);
```

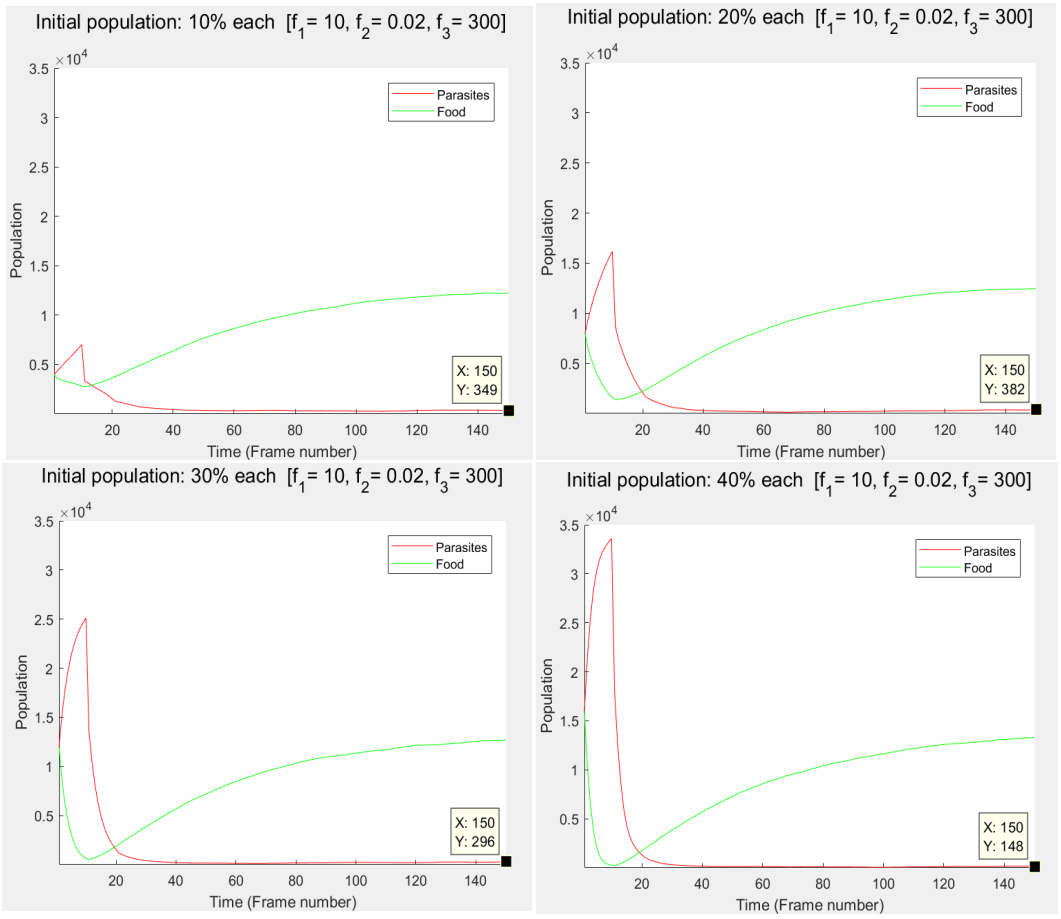


Figure 12: Initial population density sweep – 10%, 20% (control), 30%, 40%

Figure 12 highlighted that initial population density (PD) had minimal effect on the system's steady-state. The coordinates in Figure 12 highlighted that parasite population at frame 150 in all four simulations were 349, 382, 296 and 148, respectively (no linear trend). Compared to the beginning of the curve, this represented a minute difference (of 284 agents) and the food agents shared very similar equilibrium populations. However, in terms of the initial behaviour of the system, the population density had significant impact as both populations exhibited a peak, and this peak was much sharper when the initial population was greater.

Figure 13 (refer to Appendix 4 for an enlarged version) investigated the model behaviour when the initial densities and food placement-strategies patterns (including size) were varied whilst the control parameters (population density = 0.2 (20%), $f_1 = 10$, $f_2 = 0.02$, $f_3 = 300$) were used. This showed that the most significant difference when comparing the two areas (22,500 units for left two plots vs 2500 units for right two plots), was the reduction in food agent's equilibrium population (green line). To highlight this, in frame 150, the maximum number of food agents was 4312 which occurred in the 2nd graph (large area), and minimum occurred in 3rd graph (small area) with 173 food agents (96% decrease). This occurred as there was now only 11% of the original grid (from 22,500 to 2500) available for food growth. The parasite population appeared somewhat similar (as shown by the highlighted coordinates), however the smaller area simulations tended to have a slightly greater number of parasite agents as they were able to breed easier (less distance to travel for food). Furthermore, the population graph of the larger area appeared more periodic (oscillatory) in shape which could be explained by the fact that there was more space for the agents to spread and die off (rather than colliding with food and breeding).

It was clear that the positioning and shape of food-placement had lesser impact upon the model behaviour than its overall area. This was corroborated by the similarity between the population graphs of the two left plots and the two right plots. These pairs had the same area but were shifted to another location (or rectangular shape) which did not impact the overall system behaviour.

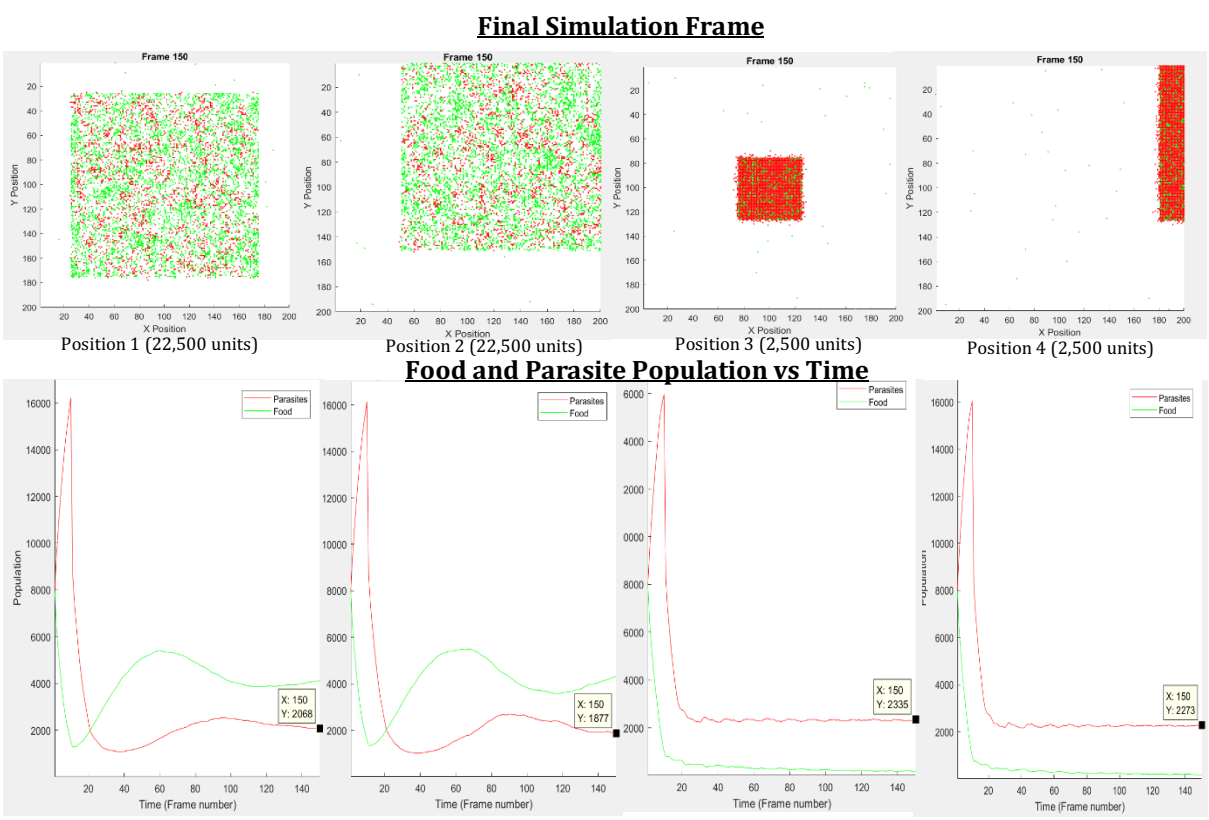


Figure 13: Food-placement strategy for centralised and top right position (22,500 units vs 2,500 units)

6.0 Discussion

6.1 Assumptions

To generate the model, several simplifying assumptions were made about the real-world system. The first assumption was the supposition that there were only two species being modelled – a parasite species and a host species. However, more realistically, a parasitic population could find a new host species or alternative source of sustenance. The second assumption was that there was no environmental complexity (i.e. both populations move unsystematically through a homogeneous environment), and that the ambient spatial and temporal conditions did not change in favour of either population. This was unlikely to occur, as biological organisms tend to move with the intent of finding food, mates, shelter, or to avoid predators and/or hazardous conditions. Such spatial and temporal heterogeneity could influence the interactions between the populations. The final assumption was that the parasites had limitless appetites and would consume infinite quantities of the host population, given the opportunity. This was unrealistic, and may even be temporally dependent, with some organisms consuming at different rate dependent on their environment (i.e. temperature).

6.2 Comparisons

Throughout the results in Task One, several visualisations were included to display successful parameter combinations. Furthermore, points were colour-coded to aid in distinguishing between the three types of successful solutions:

- Red points indicated the given parameters directed solutions toward the equilibrium point at $\left(0, \frac{k_3}{k_4}\right)$, i.e. $X_1 \to 0 + \text{Tol}$.
- Green points indicated the given parameters directed solutions toward the equilibrium point at $\left(\frac{k_3 - 2k_4}{k_5}, 2\right)$, i.e. $X_2 \to 2 \pm Tol$.
- Blue points indicated the given parameters directed solutions toward the single equilibrium point at $(0,2)$, i.e. $X_1 \to 0 + \text{Tol}$ and $X_2 \to 2 \pm Tol$.

Rifts frequently occurred which highlighted a difference between the $\text{Tol} = 10^{-1}$ and $\text{Tol} = 10^{-2}$ results. In the one parameter sweep, it was a simple gap in the line around $k_3 = 8$, in the second instance it was a linear line with a positive gradient of approximately $k_4 = \frac{1}{2}(k_3)$ and in the final instance, it was a vertical line at approximately $k_4 = 5$. In all cases, the rift appeared in place of the blue points on the corresponding 10^{-1} plots. These results could be directly attributed to the qualities of the stationary points of the deterministic system, as defined in Section 2.1. It was stated that the system had two separate steady state solutions, one at $\left(0, \frac{k_3}{k_4}\right)$ and the other at $\left(\frac{k_3 - 2k_4}{k_5}, 2\right)$, unless $k_3 = 2k_4$. In the event $k_3 = 2k_4$ the two states merged at $(0,2)$.

In terms of the real-world context of this model, the following observations were made:

- If $k_3 < 2k_4$, i.e. if the rate of growth of the host population's food source was less than double the rate of its decay, populations tended toward $\left(0, \frac{k_3}{k_4}\right)$ – parasites to 0 and hosts to $\frac{k_3}{k_4}$. Consequently, if k_3 (growth rate of host food) tended to zero or k_4 (decay rate of host food) tended to infinity, both the host and parasite populations became extinct.
- If $k_3 = 2k_4$, i.e. if the rate of growth of the host populations food source was exactly double the rate of its decay, the populations will always tend toward $(0,2)$, regardless of the other parameters.
- If $k_3 > 2k_4$, i.e. if the rate of growth of the host populations food source was more than double the rate of its decay, the populations tended toward $\left(\frac{k_3 - 2k_4}{k_5}, 2\right)$. However, if k_5 (i.e. the rate of consumption of the host population by the parasite population) tended to infinity, or $k_3 - 2k_4$ was close to zero, the populations tended toward $(0,2)$ regardless.

Thus, in all cases, the parasite population was more likely to go extinct, whilst the host population only went extinct if $k_3 \to 0$, $k_4 \to \infty$, or $k_3 \ll k_4$ (food growth goes to 0, food decay goes to infinity or food growth is significantly less than food decay).

Similarly, in Task Two, the values of k_3, k_4, and k_5 influenced the 'success' of the system solutions independently, pairwise, and all together. Independently, for $k_3 < 0.5$, $k_4 < 0.5$, and $k_5 < 1.5$ there were a considerable number of failures, regardless of the other parameter values. For k_3 and k_4 this was more consistent, showing 99.99% and 98.52% failure rates respectively. Less so, only 13.18% of solutions failed when $k_5 < 1.5$. However, given the less clustered nature of these failures, it was simpler to generalise. Pairwise, the interaction between k_3 and k_5 was the strongest, with the clear division between 'successful' and 'unsuccessful' combinations at approximately $k_5 = \frac{10}{9}(k_3)$. In terms of three-way interactions, the 3D plot and the plot comparing k_3 and k_4, and k_4 and k_5, it was observed that low k_4 values resulted in more 'failures' as k_3 increased. However, these 'failures' were high for low k_5 values but decreased as k_3 parameter increased. These results correlated with the findings of Task One, as the results indicated probable success unless k_3 and/or k_4 were small.

Task Three revealed that the death rate of parasites could be varied and still yield successful models. It was found that successful models occurred when $0 < k_2 \leq 17$, with only one outlier recorded. This model had an advantage over the previous task, as only k_1 was fixed. Comparing to the 3D model of the Latin Hypercube Sampling, this 4D model showed how the different death rates affected the rest of the system and its successful combinations.

Task Four focused upon the impacts initial population and food-placement strategy had on the system dynamics with the f_1, f_2 and f_3 parameters controlled. The results beared striking similarity to the deterministic model explored previous where the parameters had a corresponding constant in the parasite model. For example: $f_1 \to k_2$ (as f_1 increased, parasite (X_1) decay decreased), $f_2 \to k_4$ (as f_2 increased, the food/host decay rate (X_2) increased) and $f_3 \to k_3$ (as f_3 increased, the number of food/host (X_2) births increased).

From observations made in Task One regarding real-world applications, it was clear that the host population would definitely go extinct if $f_3 = 0$ (no food births occurred), $f_2 = 1$ (100% chance for food to decay) or $f_3 \ll f_2$ (the number of food births was much smaller than the decay rate/chance of the food). This was heavily supported by Appendix 4 which showcased parameter sweeps for f_1, f_2 and f_3. The key findings of these sweeps were presented in Figure 28 below which depicted the end-population against the varied parameter. When $f_1 = 15$ (maximum), $f_2 = 0$ (minimum) and $f_3 = 400$ (maximum) the parasites tended to have a larger population - which meant that greater parasite life-spans, lower food decay and higher food growth corresponded to increases in parasite population. Furthermore, this also showed that an equilibrium (success) - when parasite population became extinct ($X_0 = 0$) – was achieved when: $f_1 \leq 5$, $f_2 \geq 0.05$ or $f_3 \leq 200$ (using the control parameters).

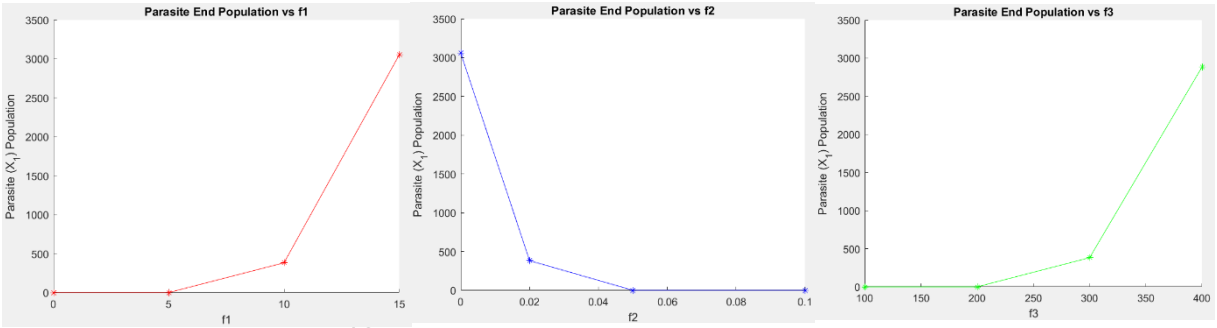


Figure 28: f_1, f_2 and f_3 Parameter Sweeps (from left to right)

The initial food and parasite population densities had minimal impact on the final steady-state of the solution. This was because the solution eventually stabilised to equilibrium which was more dependent upon the long-term parameter ratios (i.e. f_1, f_2 and f_3). Furthermore, the 'location' of the food-placement did not have as much impact as the 'area' of the grid available for food-placement (Figure 13). Finally, if an alternative location or rectangular shape was simulated, very little change occurred when compared to another simulation with same area – and these minor differences were mostly attributed to the stochastic behaviour of the model.

7.0 Conclusion

Tasks One through Three were founded on the system of differential equations and yielded deterministic solutions, and each simulation produced the same result with fixed parameters and interaction rules (i.e. no randomness). By utilising Latin Hypercube Sampling in Task Two and Three, parameter values were able to be altered and parameter sweeps conducted in 3D, and 4D space, revealed further insight into the model's behaviour. Task Two revealed the effects that k_3, k_4, and k_5 had on the success of system solutions and Task Three highlighted how k_2 also influenced the outcomes. Additional correlations between species' interactions were uncovered which clearly linked to birth/death rates of the population, food growth/decay rates and food consumption. Task Four explored the system dynamics of the model at the individual level and considering movement as a stochastic process. This agent-based approach modelled the same behaviour and the results showed distinct similarities to the deterministic approach. It also highlighted that the initial population density and spatial effects (positioning strategy) impacted the model far less than parameter ratios (f_1, f_2 and f_3). Overall, these two approaches provided a holistic analysis of the Parasite Model, and how the parameters behaved in both the average and individual population.

8.0 Appendices

8.1 Appendix 1 – Task Two Full-Sized Figures 8-9

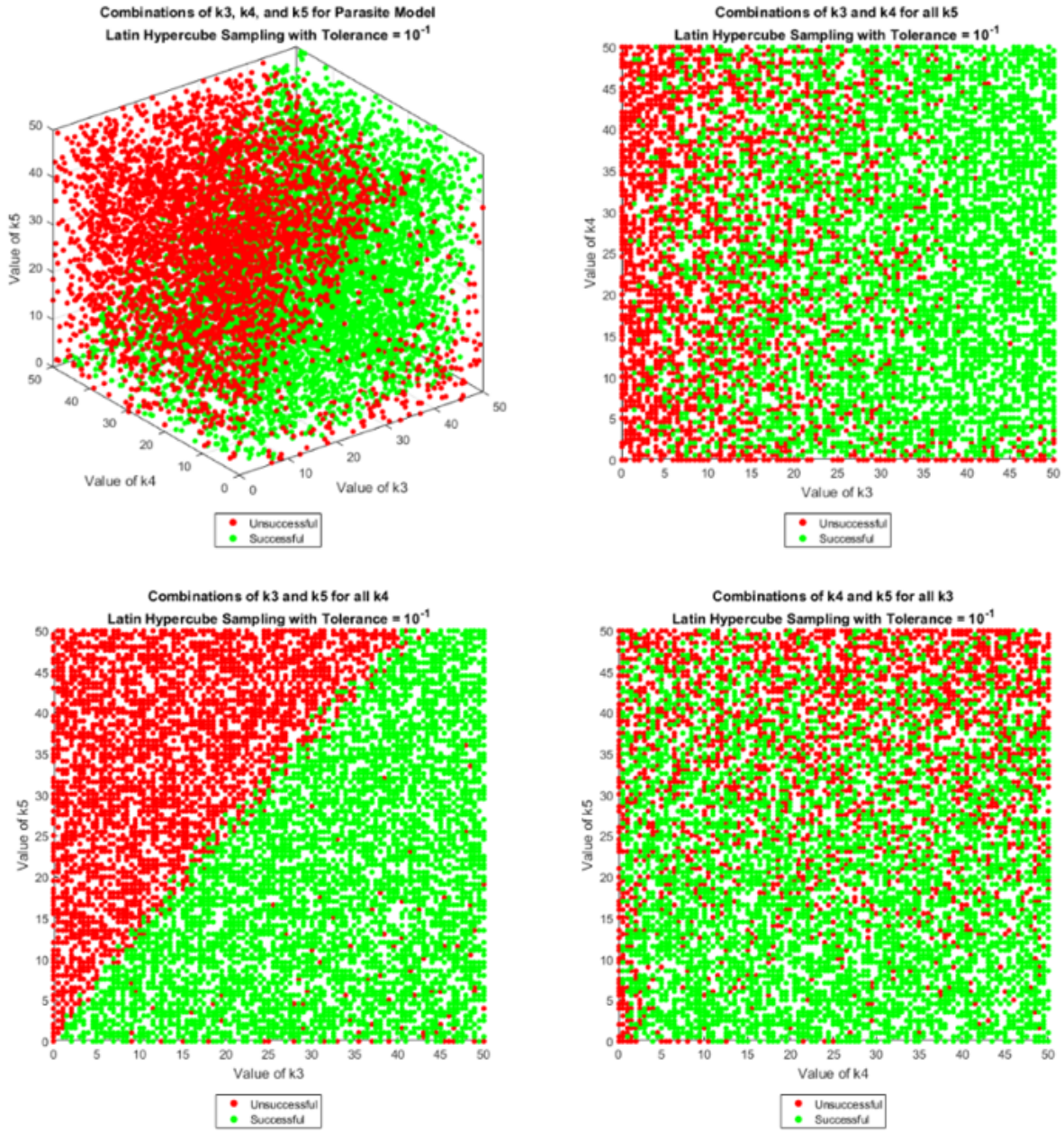


Figure 4: Plot of the results of 100 LHS trials on (k_3, k_4, k_5).

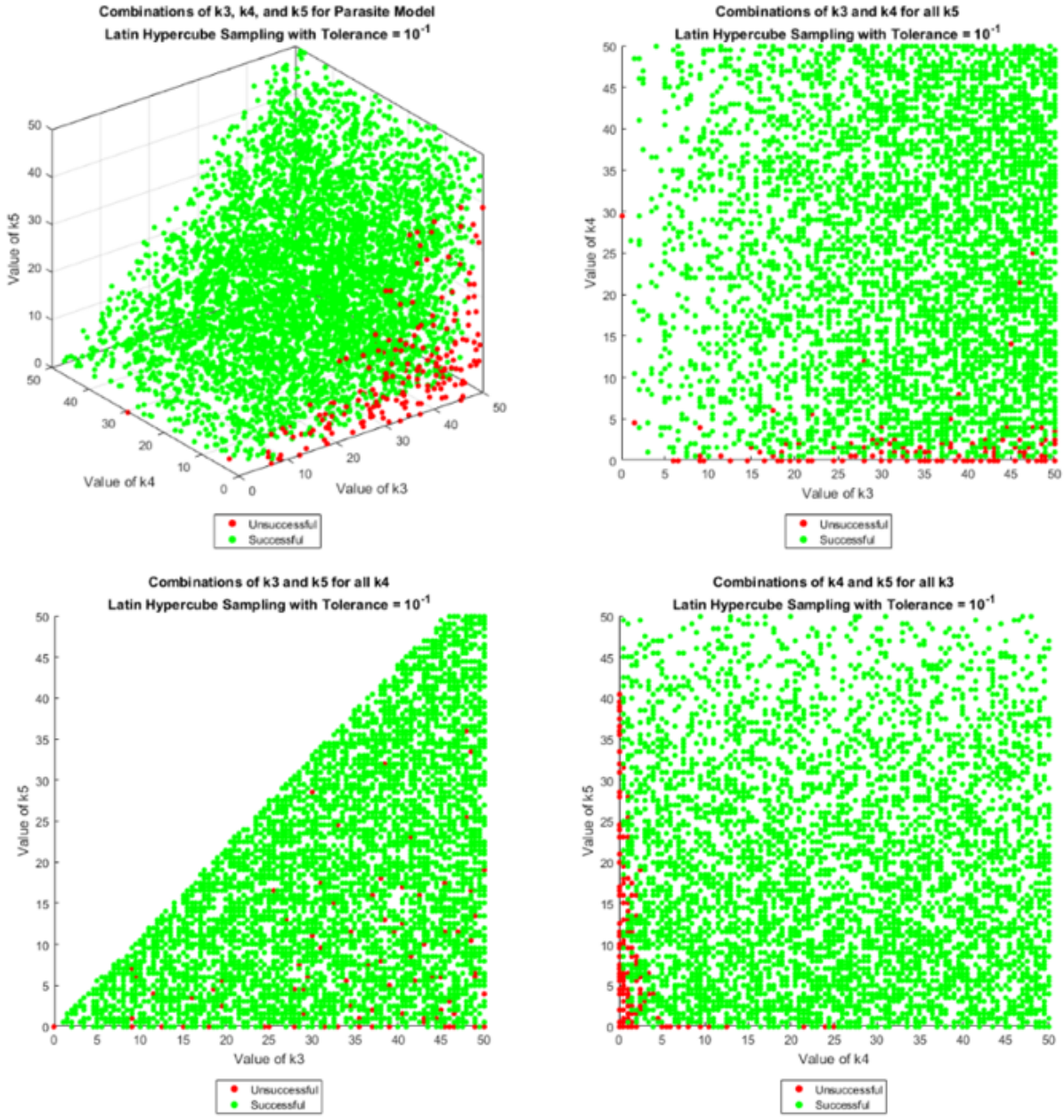


Figure 5: Plot of the results of 100 LHS trials on (k_3, k_4, k_5), where $k_5 \leq \left(\frac{11}{10}\right) k_3$.

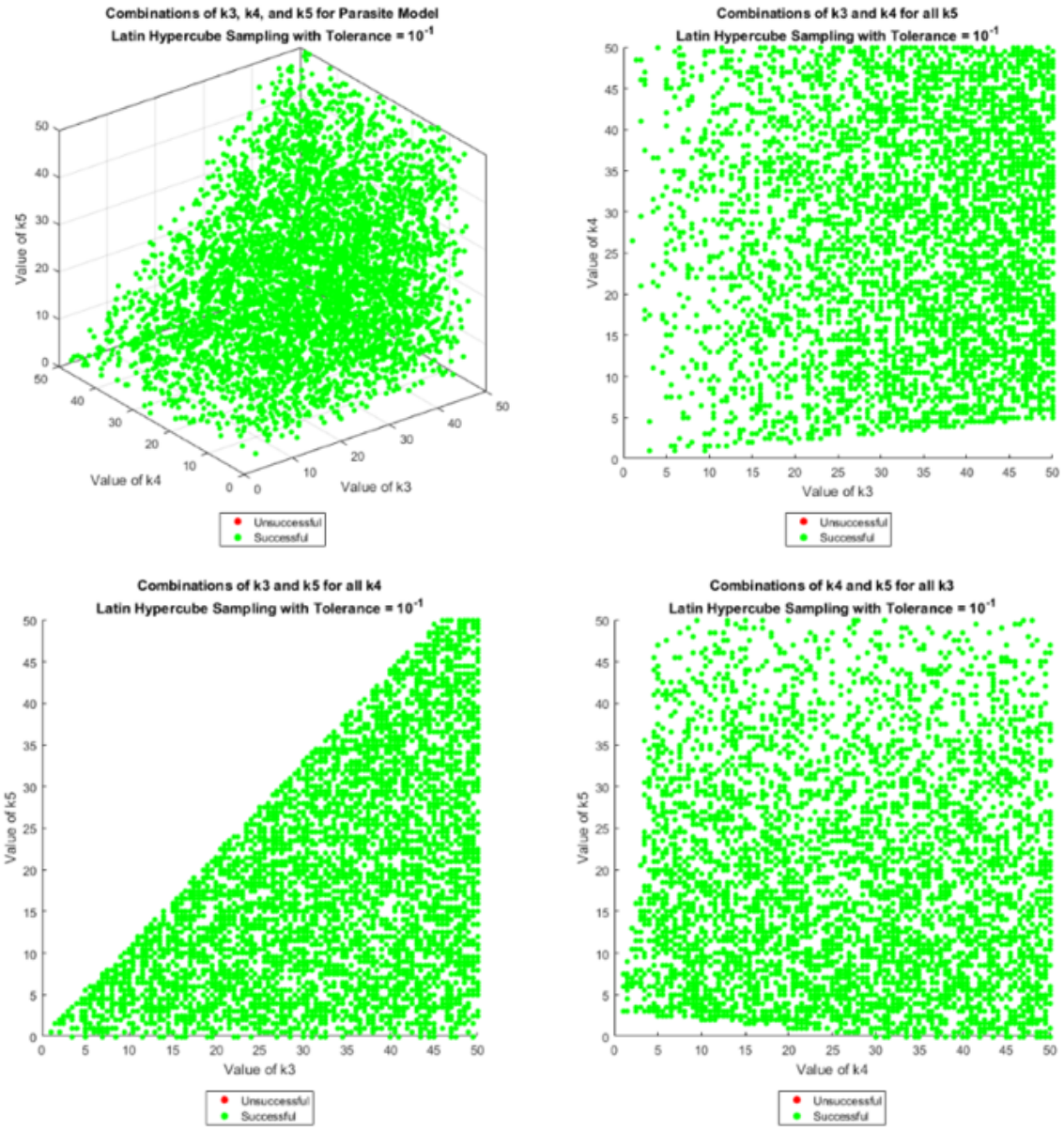


Figure 6: Plot of the results of 100 LHS trials on (k_3, k_4, k_5), with all inequalities applied.

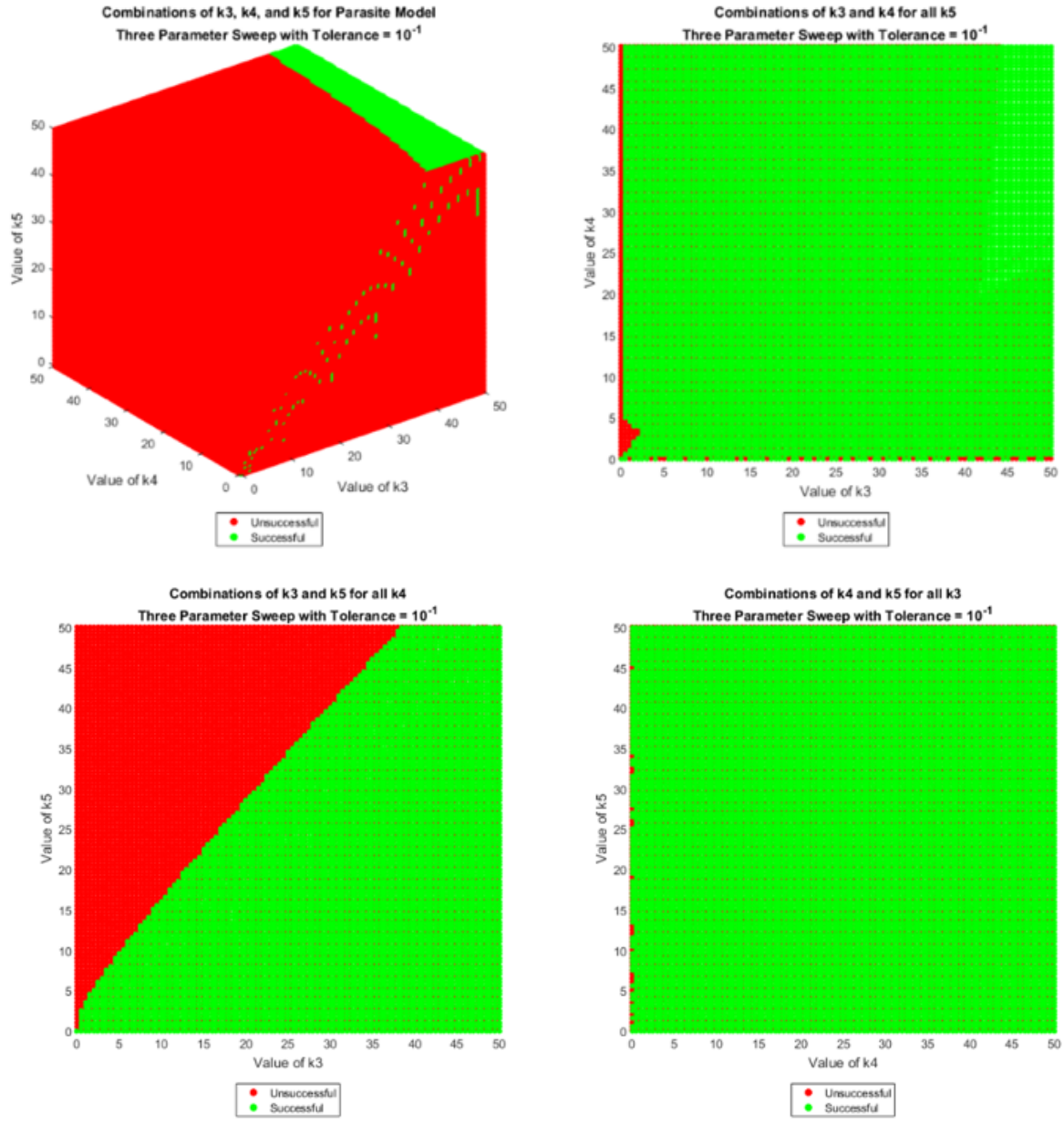


Figure 7: All combinations of (k_3, k_4, k_5), colour coded to indicate success/failure.

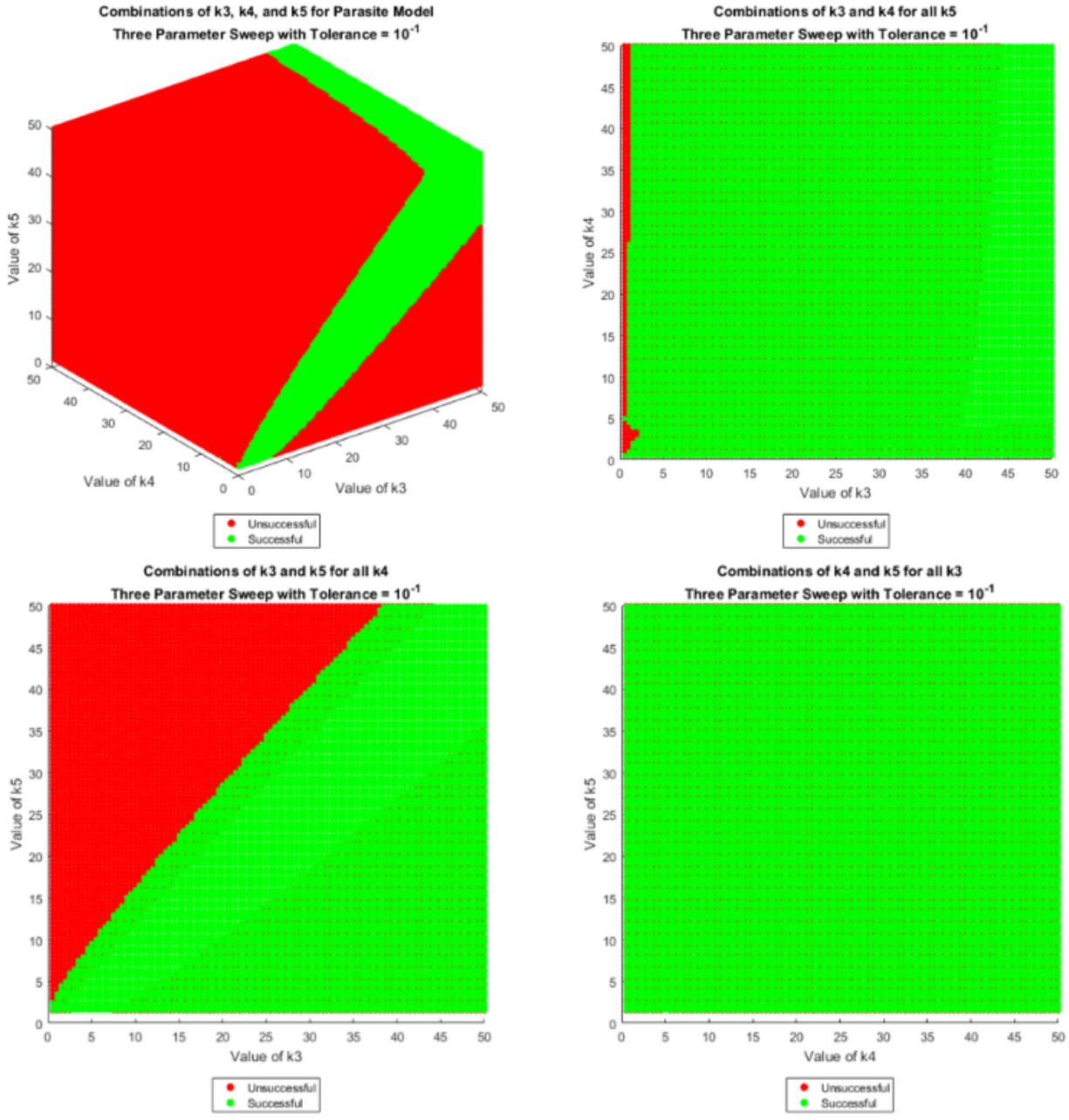


Figure 8: All combinations of (k_3, k_4, k_5), colour coded to indicate success/failure, where $k_5 \geq 1.5$, $k_3 \geq 0.5$, and $k_4 \geq 0.5$.

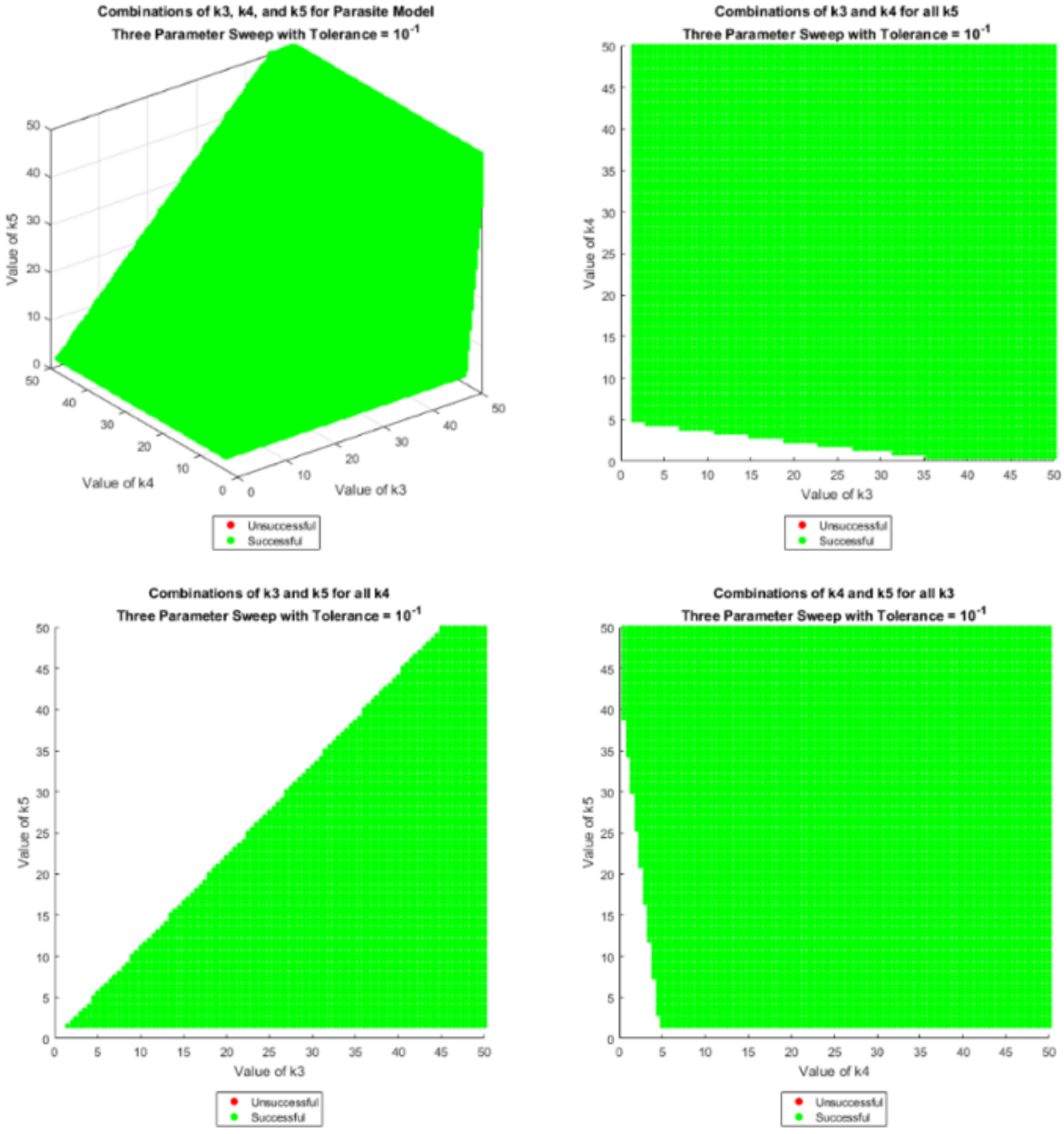


Figure 9: All combinations of (k_3, k_4, k_5), with inequalities applied.

8.2 Appendix 2 – Task Three Full-Sized Figures 10-11

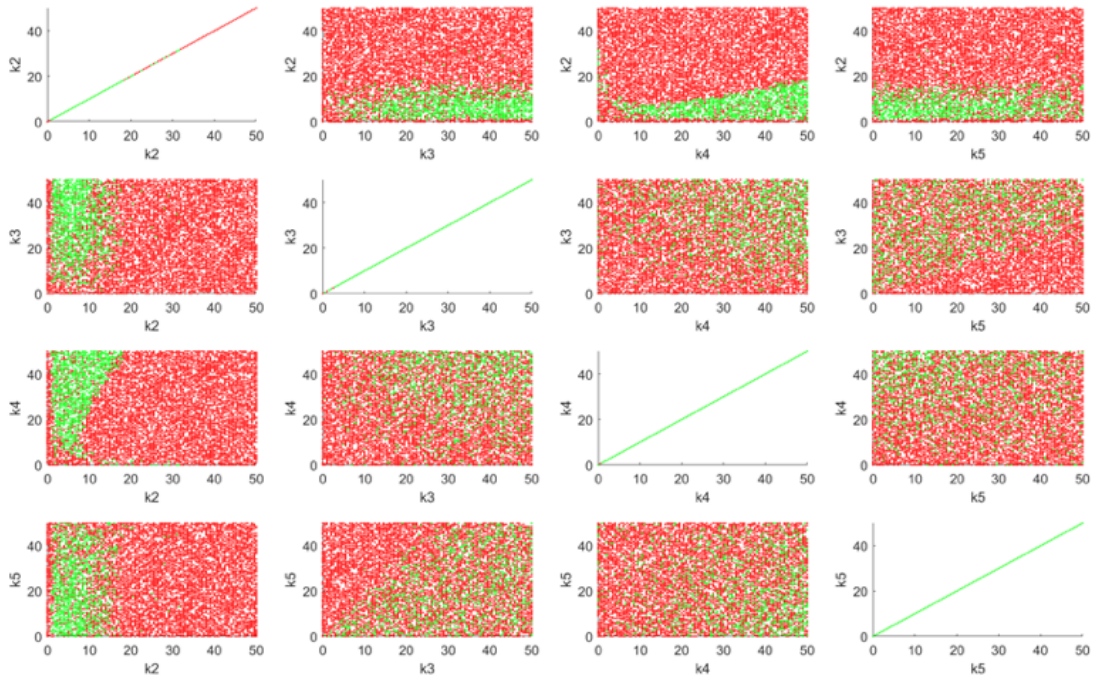


Figure 10: Comparison of k_2, k_3, k_4 and k_5 variables in combined scatter plots. Red depicts **unsuccessful** models, and green for **successful**.

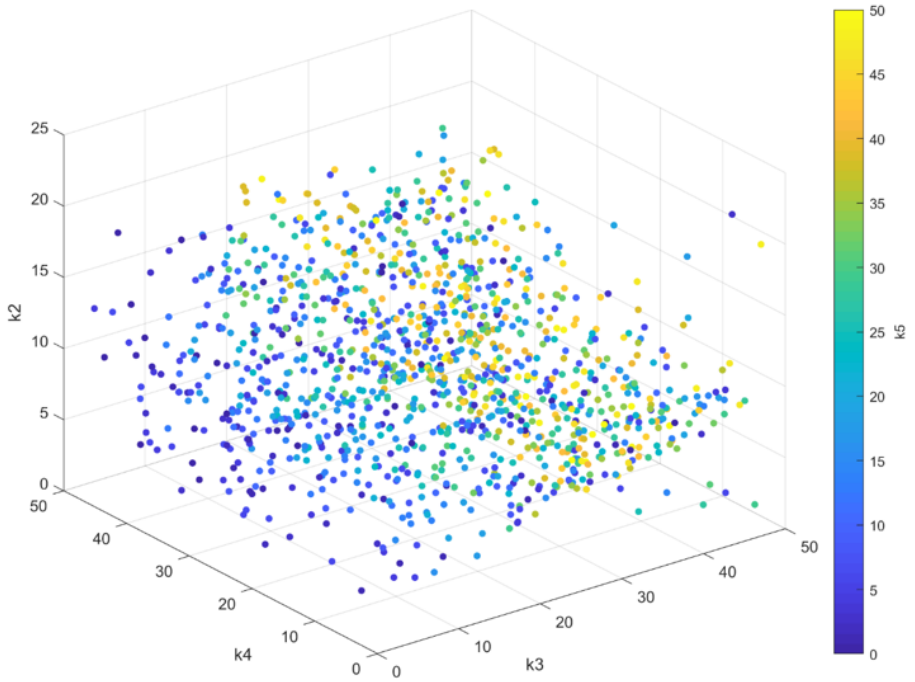


Figure 11: Successful models with k_2, k_3 and k_4 plotted in a 3D scatter plot, with colour corresponding to

8.3 Appendix 3 – Task Four Full-Sized Figure 12

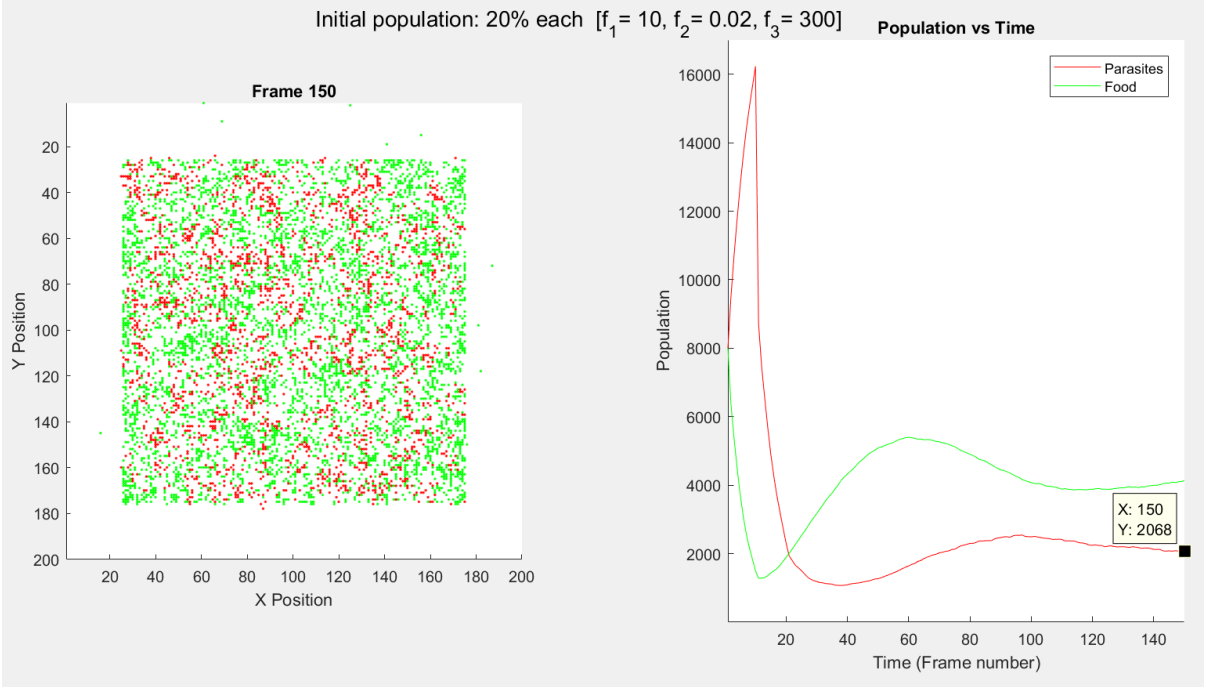


Figure 14: Food-placement strategy for centralised large area (22,500 units)

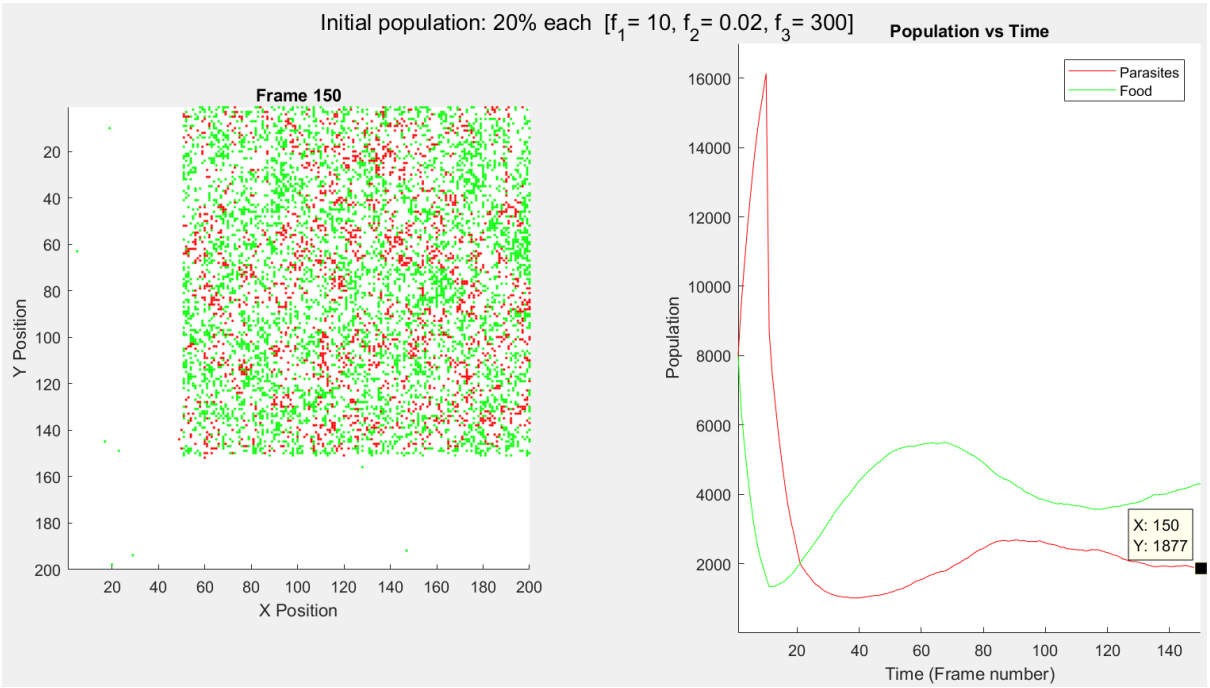


Figure 15: Food-placement strategy for top-right large area (22,500 units)

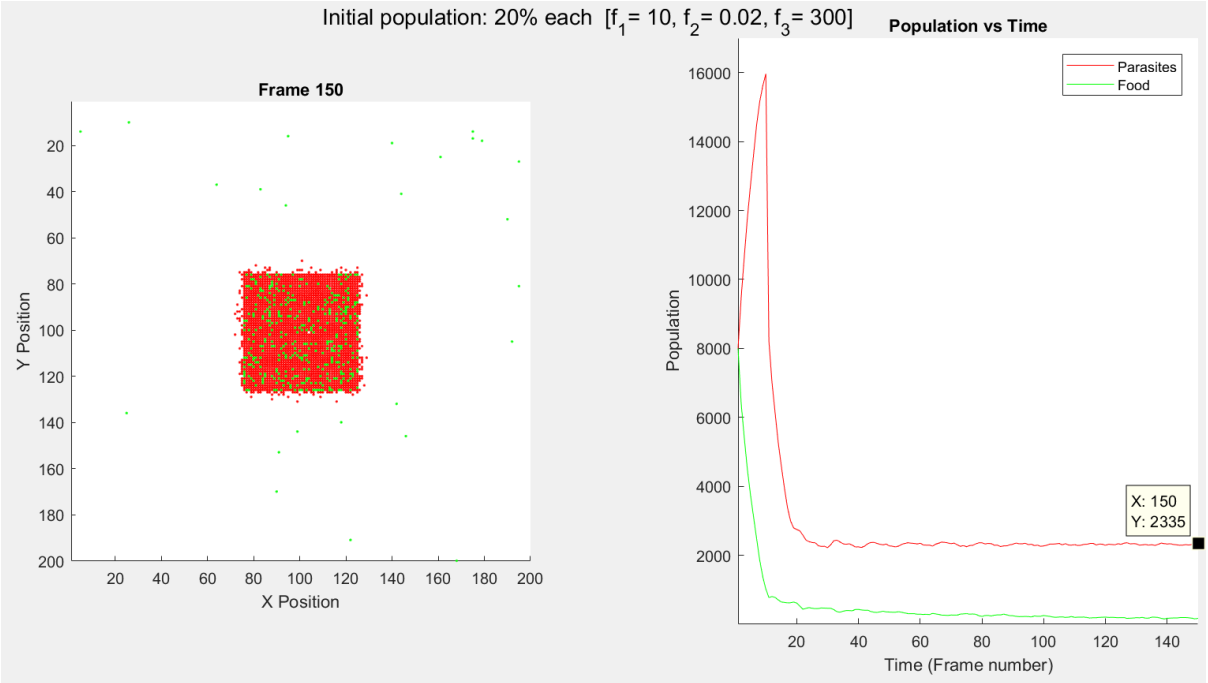


Figure 16: Food-placement strategy for centralised small area (2500 units)

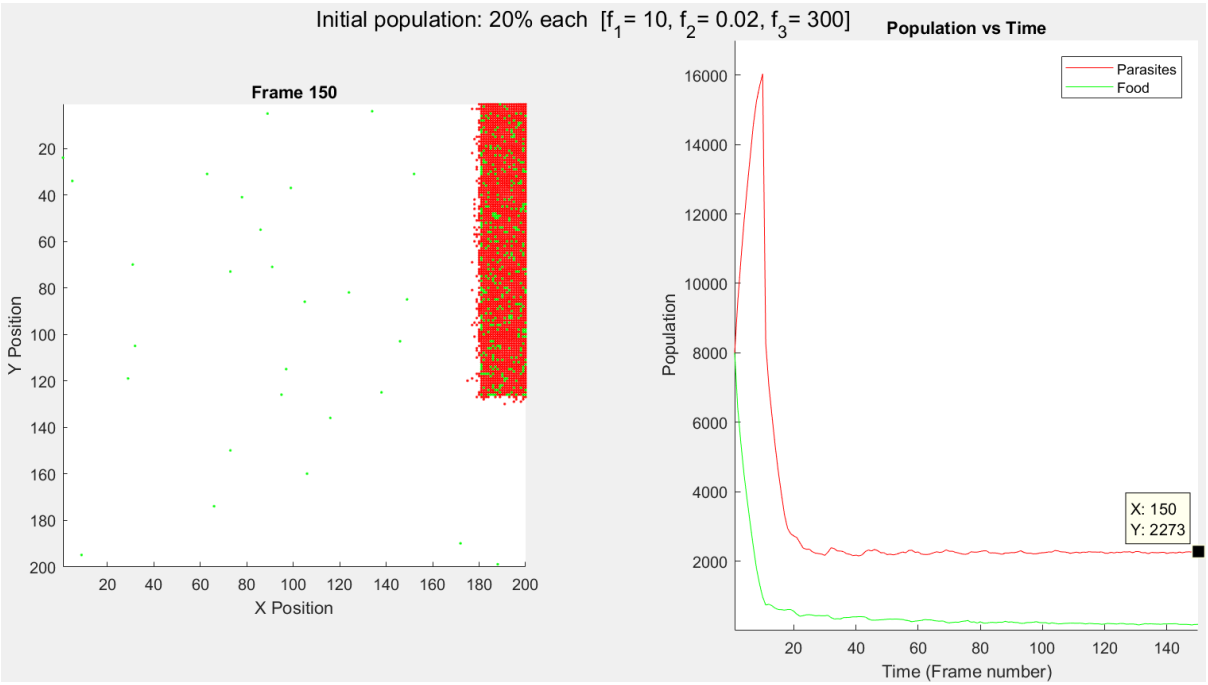
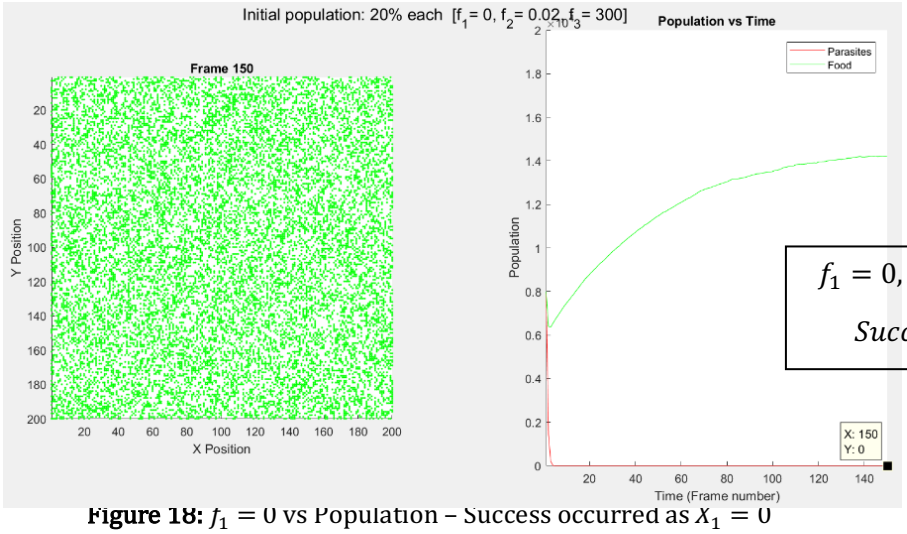


Figure 17: Food-placement strategy for top-right small area (2500 units)

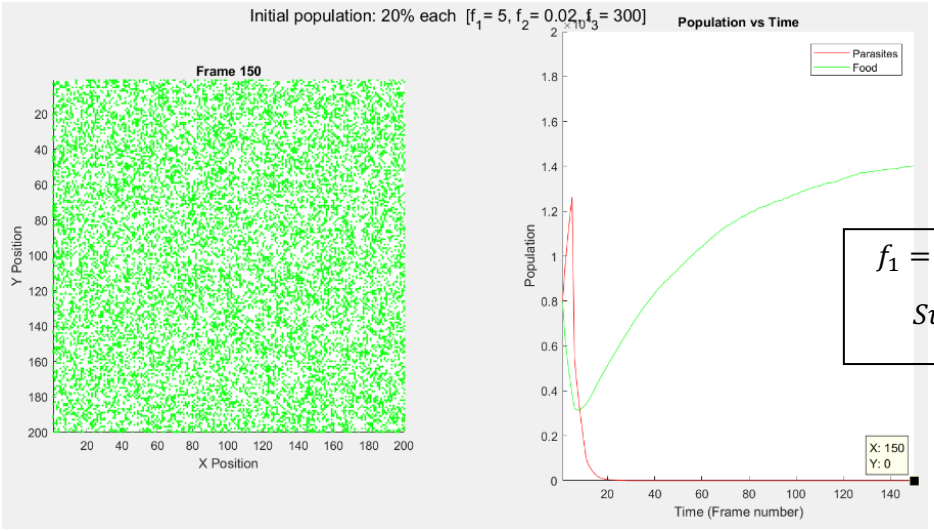
8.4 Appendix 4 – Task Four f_1, f_2 and f_3 Parameter Sweeps



$f_1 = 0, X_1 = 0$

Success

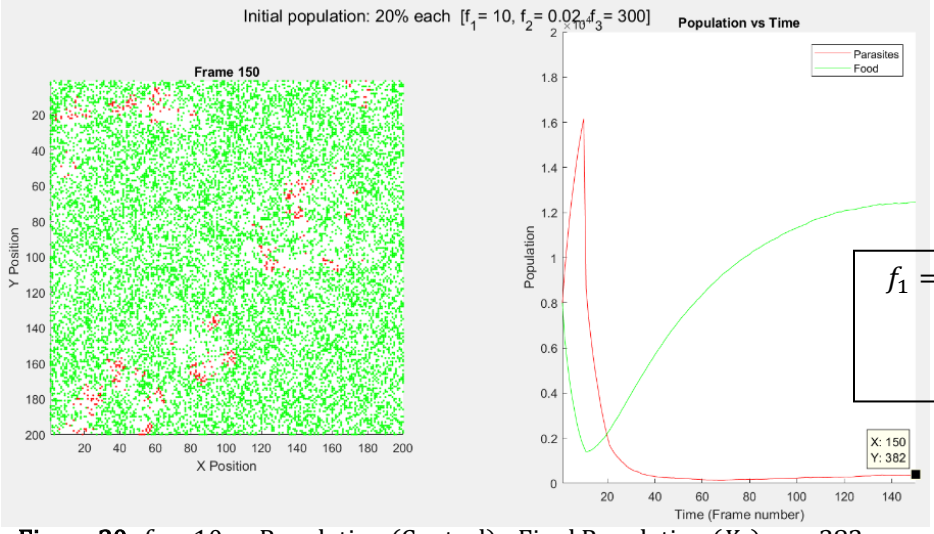
Figure 18: $f_1 = 0$ vs Population – Success occurred as $X_1 = 0$



$f_1 = 5, X_1 = 0$

Success

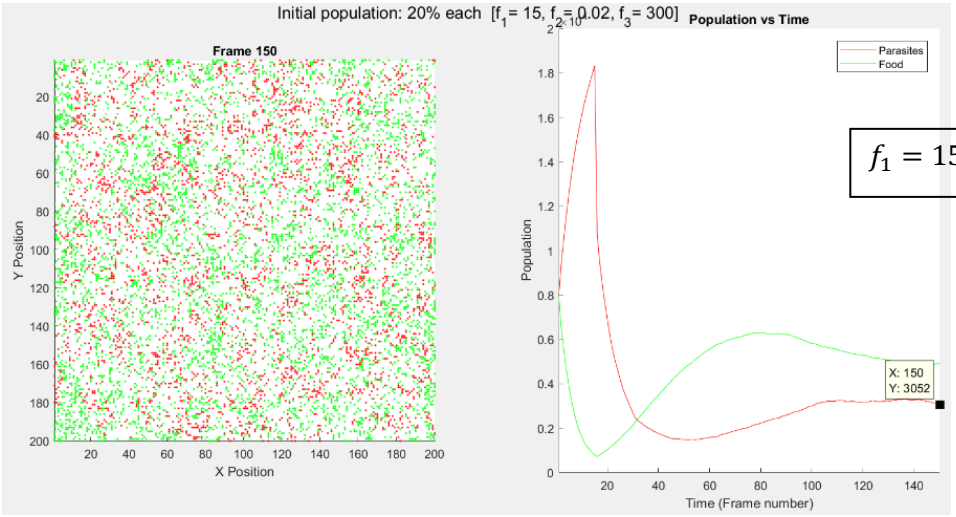
Figure 19: $f_1 = 5$ vs Population – Success occurred as $X_1 = 0$



$f_1 = 10, X_1 = 382$

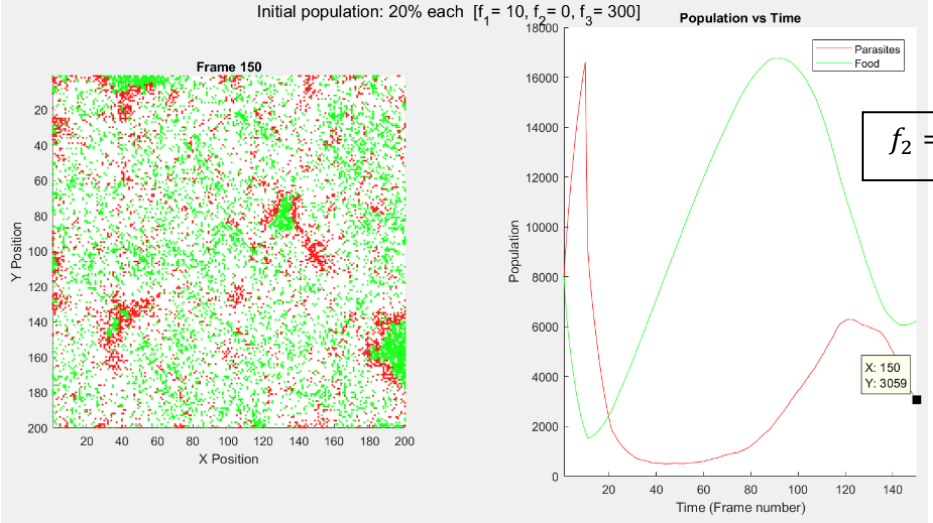
(Control)

Figure 20: $f_1 = 10$ vs Population (Control) - Final Population (X_0) was 382



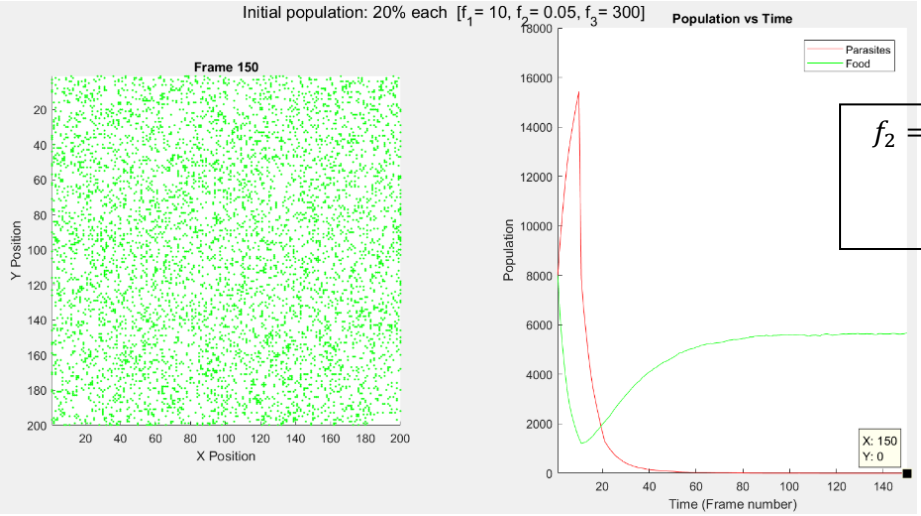
$f_1 = 15, X_1 = 3052$

Figure 21: $f_1 = 15$ vs Population – Final Population (X_0) was 3052



$f_2 = 0, X_1 = 3059$

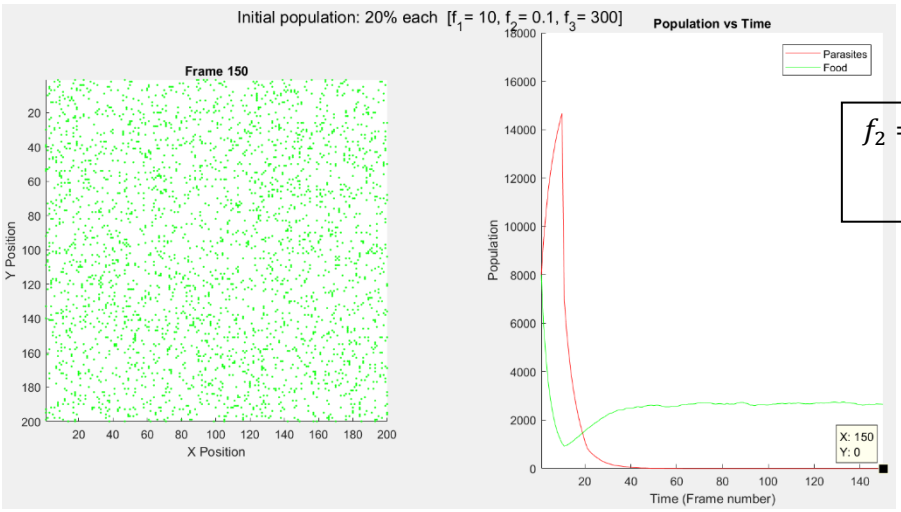
Figure 22: $f_2 = 0$ vs Population – Final Population (X_0) was 3059



$f_2 = 0.05, X_1 = 0$

Success

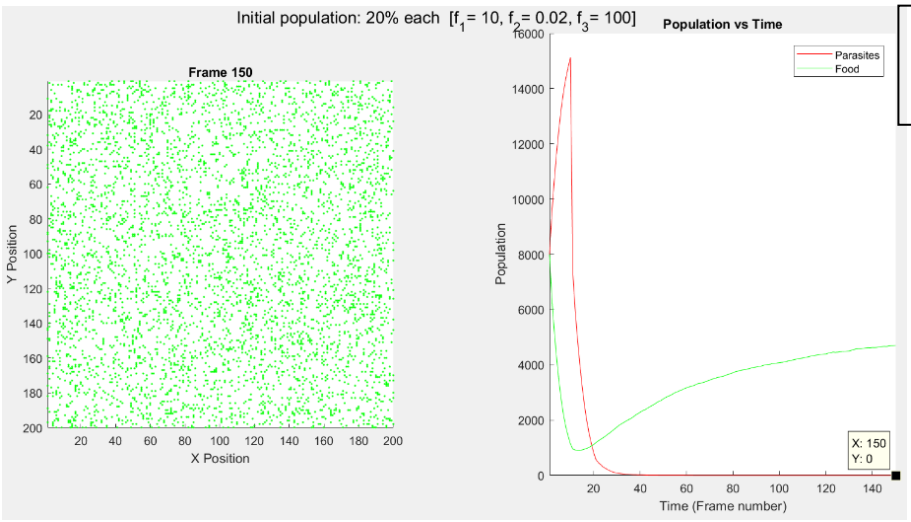
Figure 23: $f_2 = 0.05$ vs Population – Success occurred as $X_0 = 0$



$f_2 = 0.10, X_1 = 0$

Success

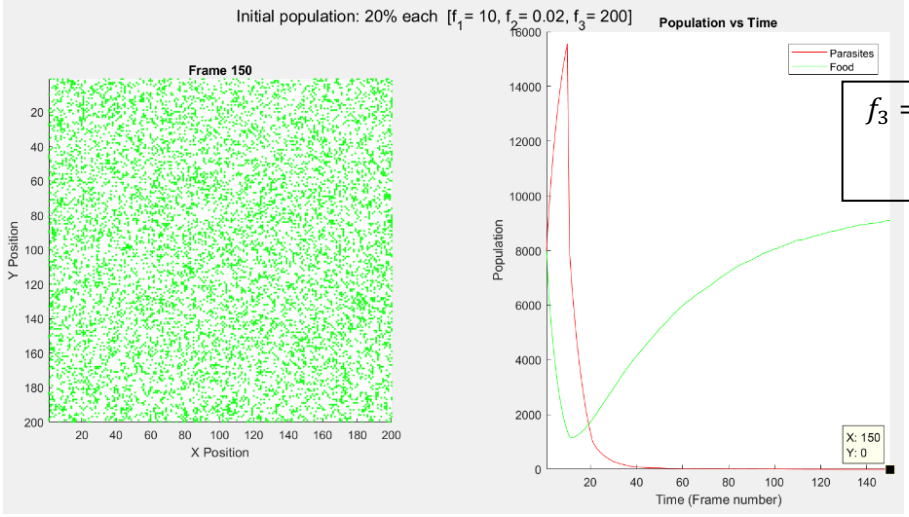
Figure 24: $f_2 = 0.10$ vs Population – Success occurred as $X_0 = 0$



$f_3 = 100, X_1 = 0$

Success

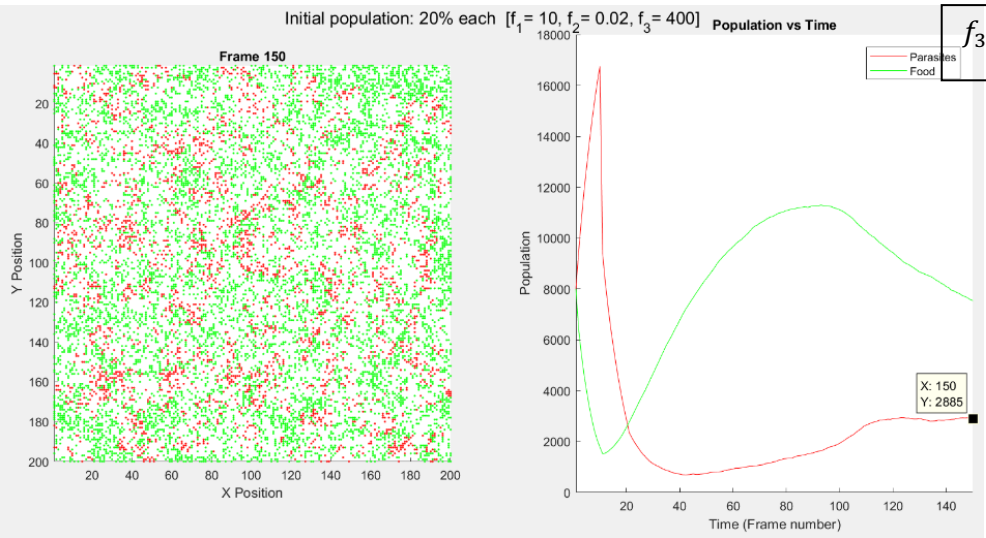
Figure 25: $f_3 = 100$ vs Population – Success occurred as $X_0 = 0$



$f_3 = 200, X_1 = 0$

Success

Figure 26: $f_3 = 200$ vs Population – Success occurred as $X_0 = 0$



$f_3 = 400, X_1 = 2885$

Figure 27: $f_3 = 400$ vs Population – Final Population (X_0) was 2885