Department of Computer Science

Learning Population Dynamics in Ant Colony Emigrations

James Collerton

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering

September 2015 | CSMSC-15

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

James Collerton, September 2015

Executive Summary

Equation discovery [87] is a set of techniques taken from the field of machine learning which automate the process of generating formulae and equations for modelling data. Within this project we consider a new application of equation discovery: deriving systems of differential equations for modelling social insect behaviour. Previously research has been limited to finding equations modelling single quantities, or rediscovering systems of differentials from data generated by well-defined underlying equations [79, 80, 88]. The work done within this thesis replicates these findings and contributes to the field by extending past these aims and into trying to derive systems of differential equations from complex data generated by processes with probabilistic components. As opposed to previously, within these new applications we do not know the underlying equations used to generate data, or even if such equations exist. All of this is done under the hypothesis that we can use equation discovery to automate the process of deriving differential equation models from data.

These investigations into equation discovery are done within the context of modelling emigrations of the ant species *Temnothorax albipennis*. A method is required to derive differential equation-based summaries of existing agent-based models of emigrations, and equation discovery has been forwarded as one possible method of achieving this: specifically using the Nutonian Eureqa system [79]. Eureqa was chosen as it is free-form in its approach, a property thought to be advantageous over the stronger declarative biases required by other systems.

Within the project we show that using Eureqa it is not possible to find differential equation-based summaries of the agent-based models. The findings of the project demonstrate a relationship between the complexity of equations within a model and overall model accuracy. When there are not simple, stable, previously defined equations to discover, the formulae required to form a differential equation-based model from data become increasingly complex in order to encapsulate the necessary behaviours. As equations become more complex and the number of variables remains the same, the sensitivity of the equations to the values of the variables increases. Then as the equations for different variables interact within a system, small errors begin to propagate across the model and grow rapidly, leaving the result inaccurate and unstable. To summarise: complex individual equations combine to form an unstable model.

These results inspire a set of experiments which attempt to stabilise the models derived using Eureqa. Initially we identify the reason that the Eureqa system does not derive stable models: it does not consider the interaction of the equations it derives. We address this problem using optimisation, iterative, and brute force approaches. Brute force methods were most effective but are computationally costly and may not be applicable across multiple emigrations. The success of the brute force approach also underlines one of the most interesting findings of the project: sacrificing accuracy for lower complexity on an individual equation basis can lead to more accurate overall models.

The main contributions and achievements of this project can be considered as:

- **An analysis of Nutonian Eureqa:** The project verifies the most recent research within our considered context.
- **A mathematically tenable method of generating general behaviours of agent-based models:** The agent-based models output data with complex differentials, making it difficult to find representative equations. A nonparametric regression-based approach to defining general behaviours of emigrations and simplifying these differentials has been defined within the work done on the thesis.
- **The identification, demonstration and rationalisation of current weaknesses in equation discovery applied to discovering systems of differential equations from complex data:** The thesis explores an existing weakness in the application of Eureqa to deriving systems of differential equations modelling complex data.
- **The results of experimentation done to address the weaknesses in the Nutonian Eureqa system:** A number of approaches were tried in order to stabilise the differential equation model returned by Eureqa. Their results will motivate and provide the basis for further work in the area.

Acknowledgements

I would like to acknowledge Dr. Oliver Ray for his support and advice during the completion of my thesis. Also, Gleb Kolpakov, Gregory Southgate, Martin Garrad, Alisdair Venn and all of the members of the Ant Labs, both computational and real. Acknowledgements also go to Richard Grafton for his help with the University of Bristol High Performance Computing Laboratory and University College, London for access to their facilities.

Thanks especially goes to my parents for their support during the year, and my Grandfather for his assistance.

Contents

Part I
Introduction

Within this project we will be looking to investigate the application of *Equation Discovery* [87] from the field of machine learning to systems of differential equations. Equation discovery covers techniques which employ genetic algorithms to automatically distil data into analytical laws, turning numeric values into representative formulae. We will analyse the ability of these techniques to find systems of differential equations, demonstrating and evaluating the strengths and weaknesses in the field. We will also discuss experiments and techniques employed to circumvent the weaknesses and consider the most important steps going forward in the area.

These investigations will be contextualised in terms of the population dynamics of the ant species *Temnothorax albipennis*, a species that demonstrates a decentralized decision making process for emigrating from a single original nest to one or more replacements [46]. These population dynamics are an exciting area of study relevant to both biology and computer science and could potentially hold the key to important discoveries in each of these fields. They also are an appropriate choice for the application of the project as there exists a challenge within the field requiring the derivation of systems of differential equations from data.

This challenge is related to a number of different models that were constructed in order to understand more about the behaviour of *Temnothorax albipennis* colonies. Some of the models use systems of differential equations to summarise the behaviour of ants during an emigration. Other models, however, use agent-based modelling to simulate the emigration process. Within this project we consider two differential equation-based models, *Pratt* [71] and *Planqué* [66], and two agent-based models, the *Simulated Positional Ant Colony Emigration (SPACE)* [78] model, based at the University of Bristol, and the *Ant House-Hunting Algorithm (AH-HA)* [49]. Each modelling approach has both advantages and disadvantages, and the applications of the project revolve around the ongoing aim in the fields of biology and computer science of unifying both approaches by being able to derive representative systems of differential equations from the agent-based models.

Equation discovery was forwarded as one potential method to achieve this [74]. Therefore we explore the ability of equation discovery to derive systems of differential equations through an attempt to discover differential equation-based summaries of the agent-based models. To achieve this, we use the Nutonian Eureqa system [79]. Other equation discovery systems were considered, but the free-form nature of Eureqa was deemed advantageous over other systems with a more rigid declarative bias.

Within the work done on the project we first recreate the results of the most recent research in our considered context (*Section* 7). Following this we move past current research in order to make our own contribution to the field, applying equation discovery to deriving systems of differential equations from complex data generated by processes with probabilistic components. We demonstrate that the derivation of differential equation-based summaries from the agent-based models is not possible using current equation discovery techniques due to a relationship between data complexity, equation complexity and model stability (*Section* 8). We attribute this to the Eureqa system not considering the interaction of equations that it derives. These findings are then used in experiments attempting to address the problems in the field (*Section* 9).

In the following thesis we will examine the steps taken to achieve the aims of the project, and the results discovered in the process.

1 Aims and Objectives

The aim of this thesis is to **investigate the application of equation discovery to modelling the behaviour of social insects**, specifically demonstrating, rationalising and attempting to address the current weaknesses in the application of equation discovery to deriving systems of differential equations from complex data. We distil this aim and the points made in the introduction into the following objectives for the project:

- **A preliminary analysis of Nutonian Eureqa:** Primarily, we wish to learn more about the capabilities of the Eureqa system. This will involve testing the system in order to verify the most recent research in the field within the context of our project [20, 55, 79, 80]. We aim to provide a thorough evaluation of our experimentation in order to better understand the merits and flaws of contemporary equation discovery techniques.
- **A demonstration and rationalisation of the weaknesses of equation discovery applied to deriving systems of differential equations from complex data:** We then aim to expand away from the boundaries of current research, and into deriving systems of differential equations from complex data. In doing so we hope to demonstrate and rationalise why, as data becomes more complex, the Eureqa system fails.
- **Experimentation attempting to address the current weaknesses in the application of equation discovery to deriving systems of differential equations from complex data:** On having experimented with both the strong and weak points of Nutonian Eureqa, we will attempt to employ the strengths in addressing the weaknesses of the system.
- **Findings relevant to future attempts to derive differential equation-based summaries of the agent-based models of ant population dynamics:** We consider the prior aims within our context of interest. In addressing the weak points of the Eureqa system applied to deriving systems of differential equations, we can use this work to motivate future research towards the ultimate aim of deriving differential-equation based summaries of the agent-based models of *Temnothorax albipennis* emigrations.

The structure of the thesis will be based around these objectives and is outlined in *Section* 3.

In terms of how the thesis will approach the discussion of these objectives, we will focus on the theoretical aspects of the project. An overview of all implementation details (languages, technology requirements *etc.*) can be found in the *Appendix* and in the source code, and will be referenced when necessary. This approach is employed as this project focusses mainly on concepts and theory and less on developing software or writing code.

We now build upon these aims, outlining the motivation for the project.

Figure 1: Forward facing (*left*) and top down (*right*) views of *Temnothorax albipennis* heads, the ant whose population dynamics will form the focus of this project. (Images sourced from [59, 61])

2 Motivation

This project incorporates elements from the fields of biology and computer science, and the results will add value over both disciplines. *Section* 2.1 covers how the goals of the thesis will bring benefit to computer science, whilst *Section* 2.2 offers a discussion of the added value to biology.

2.1 Benefit to the Field of Computer Science

The following points of added value to the field of computer science will be provided by the work done within the project:

- **An evaluation of the strengths and weaknesses of equation discovery applied to discovering systems of differential equations.**
- **A set of results from experimentation done in an attempt to address the current weaknesses in the application of equation discovery to systems of differential equations from complex data.**
- **Findings relevant to future attempts to derive differential equation-based summaries of the agent-based models of ant population dynamics.**

The focus of this project will be on understanding the limitations of contemporary equation discovery techniques applied to systems of differential equations. This will offer added value in two main areas. Initially, and specific to the field of equation discovery, we attempt to apply equation discovery techniques to data with no known underlying trends. Previously when systems of differential equations have been looked for there has been a well-defined underlying system used to generate the data [80,88]. The aim of the equation discovery has then been to rediscover that model. This project acts as a first attempt at discovering a system of differential equations to model data with no known underlying system.

By demonstrating that it is not possible to use available techniques for our purposes, and evaluating and assessing why, we create an area for future improvement. The experimentation done in an attempt to address these weaknesses then offers a direction for these improvements.

Secondly, we consider the contextual aim of this project, discovering differential equation-based summaries from agent-based models of ant population dynamics. The work done during the thesis will show that equation discovery is not immediately suitable for achieving this objective. The reasoning behind this will also help shape future efforts in the area, by acting as a warning of similar problems that may occur when using related techniques.

When finally discovered, the differential equation-based summaries of the agent-based models will offer several points of added value in themselves. Primarily, the development of one of the agent-based models, SPACE [78], is an ongoing project at the University of Bristol computer science department. The project is constantly searching for evidence of the model demonstrating realistic behaviours, or for points of improvement. By being able to generate differential equation-based summaries of the model it will be possible to approach the evaluation of SPACE from a brand new direction.

Additionally, the study of decentralized decision making systems has an application in computer science. The population dynamics of *Temnothorax albipennis* are studied within the contexts of *artificial decision making algorithms* [5] and *anytime algorithms* [13,95] (a relationship covered more fully in *Section* 4.3.1). By finding accurate differential equation-based summaries of the ant population dynamics, it may be possible to derive analytically colony properties that link to these areas. The experimentation done within this thesis will aid in future work attempting to achieve these aims.

2.2 Benefit to the Field of Biology

Insights into equation discovery applied to systems of differential equations in the context of ant population dynamics also add value to the field of biology. The main source of added value is:

- **A set of results that can inform future efforts associated with deriving differential equation-based summaries of the agent-based models, SPACE and AH-HA.**

The ability of social insects to function as a single information-processing unit has long been a subject of study in the field of biology [8, 32, 81]. The ant species *Temnothorax albipennis* is a prime example of one such insect, and demonstrates the capacity to combine simple behaviour on an individual level to generate more complex behaviour on a colony-wide scale. Because of this, it is often chosen as the subject of studies on naturally occurring decentralized decision making processes [46, 49].

Setting up experiments in order to examine this behaviour is a complex task. One common problem is the marking of individual ants for tracking, which involves their immobilization using CO_2 and the application of paint to their shells. This process is time consuming, can damage ants, and is by no means a perfect tracking system. For example, in the work of *Pratt et al.* [71], whilst attempting to mark six colonies, they found that by the start of the emigration $15\% \pm 12\%$ of the colony had either lost their markings or had been missed when the marking had taken place.

Figure 2: A colony of marked ants. (Image sourced from [2])

The SPACE model [78] was introduced by Master's students at the University of Bristol in order to offer a computationally based method of experimentation, removing the difficulties of running physical experiments. SPACE looked to incorporate the latest biological findings into a model that could be run through the *Unity3D* game engine [1]. This could then serve as an initial test and as a reasoning tool for biologists' hypotheses, before going to the effort of implementing experiments with real ant colonies.

The first iteration of SPACE was completed in 2013 [78] and was enhanced during the 2014 academic year [30]. The resulting model was evaluated to realistically simulate ant colony emigrations by checking it displayed known properties of emigrations, comparing it to other successful models and by verifying it with expert opinion [30, 78].

An area of active interest (and the motivation behind studying the application of equation discovery to systems of differential equations in an ant population dynamics context) is to take the strengths demonstrated in the agent-based version of SPACE and to summarise the behaviours of the model using a set of differential equations. This will bring with it a number of sources of added value for the field of biology. Primarily, it will allow researchers to harness the useful properties demonstrated by SPACE in its agent-based form, as well as a number of useful properties that are inherent in a differential form (for more details see *Section* 5.1).

For the biological community this will allow access to differential equation models representing both SPACE and AH-HA that will be fast to run and analyse and can be used to derive deterministic properties of ant population dynamics. It will also help address some of the weaknesses of the agent-based version of SPACE, one of which being that it cannot easily be compared to the existing models in the field due to their contrasting forms. The work done on this project will inform later efforts which hope to achieve these aims.

3 Methodology & Document Structure

This section will cover how the remainder of the thesis will be structured. As we detail the structure of the document, we will also outline the methodology employed in approaching the aim of the project: investigating the application of equation discovery to modelling the behaviour of social insects. This is since the thesis is structured around the methodology's component steps.

The remainder of this thesis will be split into four parts, each of which will be split into further sections. The first of the four parts will cover the necessary background for an understanding of the thesis. The second will cover the execution of the project, including the experimentation carried out, the results and their analysis. The third section covers an evaluation of our work. The final section will cover the further work that can be done in relation to the project, and the conclusions that have been drawn from the findings of the thesis.

Project Background (Part II)

The project background will be split into three:

- **Biological Background (*Section* 4):** This section will document the population dynamics of the ant species *Temnothorax albipennis*.
- **Modelling Background (*Section* 5):** Here we will cover the models and model properties used within the project.
- **Equation Discovery Background (*Section* 6):** The final part of the background section will introduce equation discovery, and the important factors of the equation discovery system we have chosen for the project.

Project Execution (Part III)

The project execution section will also be split into three:

- **A Preliminary Analysis of Nutonian Eureqa (*Section* 7):** In this section we attempt to verify the previous successes of Nutonian Eureqa by showing that it is capable of discovering approximating equations for both functions for single quantities and known systems of differential equations.
- **Discovering the Differential Equation-Based Summaries of SPACE and AH-HA (*Section* 8):** Here we begin to make our contribution to the field, and attempt to apply Eureqa to discover systems of differential equations for data where there is no known underlying set of equations. We show that discovering differential equation-based summaries of the agent-based models using current techniques is not possible and conclude with an explanation of why the Eureqa system cannot achieve this.
- **Experiments on Stabilising the SPACE Differential Equation-Based Summaries (*Section* 9):** The weaknesses demonstrated in the previous section are then used to inform a set of experiments in order to stabilise the models derived from Eureqa. This section contains the methodologies, results and analysis of these experiments.

Project Evaluation (Part IV)

The evaluation of the project is split into the following:

- **An Evaluation of Contributions and Achievement of Objectives (*Section* 10):** In *Section* 2 we laid out a number of proposed contributions to the fields of biology and computer science for the project. In this section we will review whether we made these contributions, simultaneously evaluating if we achieved the objectives laid out in *Section* 1.

- **An Evaluation of the Wider SPACE Project (*Section* 11):** SPACE is an ongoing project at the University of Bristol. In this section we evaluate how what we achieved within the thesis will impact the work surrounding SPACE.
- **An Evaluation of Nutonian Eureqa as a Tool for Deriving Differential Equation-Based Summaries of the Agent-Based Models (*Section* 12):** Within this section we critically examine the suitability of Nutonian Eureqa as a tool for the contextual aim of this project. We also suggest and discuss alternative methods to both Eureqa and equation discovery for achieving this aim.
- **An Evaluation of the Aims of the Project (*Section* 13):** The final evaluation comprises of an examination of the aims of the project, and the implications of trying to derive systems of differential equations in the manner we have attempted.

Further Work and Conclusion (Part V)

The further work and conclusion part of the project will be split into the following sections.

- **Further Work (*Section* 14)**: This section converts the findings of the thesis into recommendations for future research.
- **Conclusion (*Section* 15):** This final part concludes and summarises the project, bringing together everything we have learnt.

Part II
Project Background

To achieve the aims set out in the introduction, it will be necessary to introduce a number of areas core to the project. These can be split into *biological background*, *modelling background* and *equation discovery background*.

- **Biological Background (*Section* 4):** Within this section we will examine the population dynamics of *Temnothorax albipennis*. Starting from how contemporary literature describes the behaviour of individual ants, we will use this to explain how the actions of individuals combine to generate a complex decision making process on a colony-wide scale. Included within this section will be a description of some of the properties that ant colonies have been known to demonstrate, including their ability to trade speed for accuracy and speed for cohesion.

  A detailed and comprehensive understanding of the behaviours of ant colonies will be vital throughout the entirety of this project. An appreciation of one of the global behaviours of colonies in particular (the speed-accuracy trade off, *Section* 4.3.1) aids an appreciation for the relevance of studying *Temnothorax albipennis* emigrations in relation to anytime algorithms and artificial decision making algorithms.

Figure 3: A dorsal view of a *Temnothorax albipennis* colony member climbing a twig. In *Section* 4 we discuss both their individual and colony behaviours. (Image sourced from [60])

- **Modelling Background (*Section* 5):** Here we will discuss the four models of the population dynamics: Pratt, Planqué, AH-HA and SPACE. The section will start with a discussion of the properties of the three types of model considered within the project: differential equation-based, agent-based and spatial. From there we will use the relevant literature to define each of the four models in turn; covering their features and relative strengths and weaknesses.

  An awareness of the properties of different types of model underpins an appreciation for the added value of the project, where we explore equation discovery applications with the aim of defining models that can exploit the strengths of the different types. A thorough familiarity with the models is then vital as they are used throughout the exploration of equation discovery, in which they are constructed and run for various demonstrations or experiments.

- **Equation Discovery Background (*Section* 6):** This section will cover a number of equation discovery systems. The aim is to give insight into the new ground covered by this project and the relevance of the project's findings to the field of equation discovery.

4 Biological Background

This section will cover the biological background necessary for the project. It will begin with a description of the behaviour of individual ants in a *Temnothorax albipennis* colony during an emigration (*Section* 4.1). The following part will build on this, and explains how the individual behaviours combine to form a decentralized decision making process for choosing between several potential new nest sites (*Section* 4.2). The final piece of this section will be devoted to explaining some of the other properties that emerge as a result of the population dynamics (*Section* 4.3).

4.1 Individual Ant Behaviours

The ant species *Temnothorax albipennis* resides in colonies of less than 500 workers and is indigenous to Europe [65]. It is of such great interest to researchers globally due to its decentralized decision making process, which manifests itself in individual ants taking on one of a set of responsibilities during an emigration [82]. At any point during an emigration an ant can be in one of two classes: *passive* or *active*. Ants in the passive class can take one of three roles; *inactive, being led to a nest* or *being socially carried*. Ants in the active class can take any of the passive roles, or can be *scouting, assessing, forward tandem run recruiting* or *social carrying*. The proportion of ants within specific roles at a given time during an emigration characterises the emigration state.

The two classes differ in that active ants can play a role in the emigration, such as scouting for new nests or recruiting ants to their new found home. Passive ants, on the other hand, do not hold any responsibilities within the emigration, they are just guided or carried to nests by the active ants. An important distinction to be made is that inactive ants in the active class are different to inactive ants in the passive class. Inactive ants in the active class can transition to scouting and to other roles in the emigration whilst passive ants cannot. Because of this, most of the discussion of the ant population dynamics will be in terms of the active ants, as it is their actions that shape the emigration.

Emigrations start with a number of active ants *scouting* for a new nest. Upon a scouting ant discovering a potential nest, it will transition to the *assessing* state, using a mixture of weighted criteria including nest darkness, entrance width, size, and nest height to determine the nest quality [27]. If, through its assessment of the nest, the ant deems it a suitable new home, it transitions to the *recruiting* state. If it rejects the nest, it will start scouting for another one.

Once in the recruiting state, the ant will begin to search for scouting ants to lead back to its nest of choice. This process of leading ants to a new nest is known as *forward tandem running* [53]. On being led to the new nest, an ant will begin its own assessment and either choose to also begin recruitment or reject the nest and continue scouting, with this choice based upon the arriving ant's perception of the nest quality. The transition of an ant's state to a nest's recruitment is not necessarily permanent [67]; the ant can switch back to scouting or begin to recruit to another nest if one of a higher quality is presented.

At this point, we introduce the notion of the *quorum threshold* [71]. If one ant leads another back to the new nest then the following ant will learn the route between the two. This ant will now be capable of recruiting other ants back to the new nest, and the total number of recruiters for that nest has increased. However, this is a slow process. A faster process (shown to be three times as fast [72]) is to return to the old nest and carry passive ants or brood items to the new nest and then leave them there, before going back to the original nest for further ants. The drawback of this is that a carried ant does not learn the route to the new nest and therefore cannot play a part in the recruitment process.

Figure 4: The ant marked in red is leading a forward tandem run to a potential new nest site, with the ant marked in white following. The two maintain contact using tactile signals throughout the process [28]. (Image sourced from [3])

Here we reach a dilemma. If the ant transitions to the carrying state immediately on finding the nest, then the process of getting further individual ants back to the nest will be quick. However, there will only be one ant capable of carrying as the others will not have learnt the route. If the inverse happens and ants only recruit via forward tandem running, then there will be a lot of recruiters, but the process of leading individual ants to the new nest will be slow. The implication is that there exists some optimal point at which recruiting ants should swap from forward tandem running to social carrying.

The quorum is the ant colony's interpretation of this point. Once the number of ants within the new nest has reached the quorum threshold, the recruiting ants will alter their recruiting tactics from forward tandem running to social carrying.

However, tandem running in general does not finish there. Once the quorum threshold has been reached and social carrying has begun, active ants begin to employ *reverse tandem running* [66], where they lead other active ants from the new nest back to the old nest. On reaching the nest the ants then begin socially carrying the passive members of the colony to the new nest site. The biological community is still speculating on why exactly this behaviour is exhibited, but it has been hypothesised that this may increase the speed of emigrations where there is a lack of scouts in the original nest site, limiting the number of forward tandem runs [66].

The underlying rules for individual active ants have been summarised in the below flow chart.

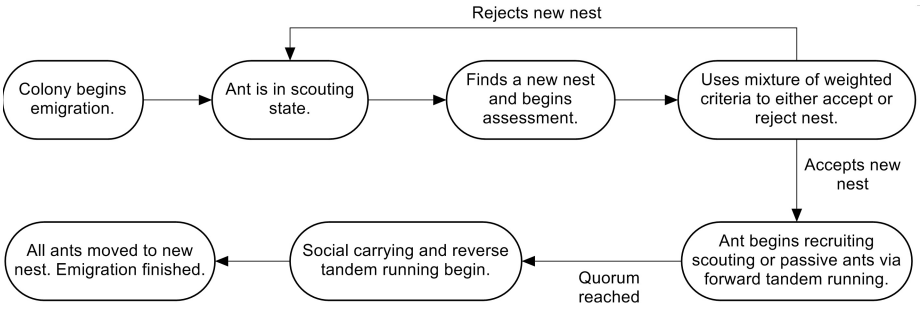


Figure 5: Flow chart representing individual active ant behaviour.

4.2 Decentralized Decision Making in Colonies

Now we explain how the behaviour of individual ants combines to generate a decision making process. This is achieved by altering the recruitment rate to nests depending on the nest quality. In *Section* 4.1, we covered how ants used a mixture of criteria and a set of weightings of each criterion's overall importance to define a quality metric for a given nest. By setting the recruitment rate to a nest proportional to its quality, the colony achieves the effect of recruiting more rapidly to better nests. This will mean that the quorum is exceeded more quickly for higher quality nests and social carrying begins sooner. Overall the colony is able to, as a whole, choose and move to a high quality nest using a decentralized decision making process.

There are two mechanisms by which it has been postulated that this can be achieved. The first is a waiting time, weighted inversely to the quality of the nest [46, 67]. By this mechanism, better nests would have a lower wait time and would accrue more recruiters. In contradiction, some contemporary literatures have argued that, in fact, no waiting time occurs and that ants immediately commit to a nest dependent on its quality [75, 76].

The second suggested mechanism involves varying the quorum threshold [75]. A higher quorum would mean that more ants would need to accept the nest before social carrying began; leading to a higher accuracy. This negates the need for a waiting time whilst offering a solution which maintains the relationship between nest quality and recruitment rate.

4.3 Emergent Properties of Ant Colony Emigrations

As well as forming a process for choosing between nests, the combination of individual behaviours within an ant colony also produces a number of emergent properties. These properties include a **speed-accuracy trade off** [26] and a **speed-cohesion trade off** [29]. An understanding of the speed-accuracy trade off aids an appreciation for the relevance of the project to anytime algorithms and artificial decision making algorithms. The speed-cohesion trade off is important in understanding one of the existing flaws in the Pratt model (covered in *Section* 5.2.1).

The following sections will require the definitions of **emigration time**, **emigration accuracy** and **emigration cohesion**.

- **Emigration time:** The time between the start of an emigration and the point at which the last passive item leaves the nest [29].
- **Emigration accuracy:** The proportion of passive items in the best nests at the end of the emigration [29].
- **Emigration cohesion:** The ability of a colony to, as a whole, have chosen the same nest by the end of an emigration [29]. Mathematically we can define colony cohesion for an emigration concerning n nests as:

$$C = 1 - \frac{H}{H_{MAX}} \tag{4.3.1}$$

Where H is Shannon's Index [63] defined:

$$H = -\sum_{i=1}^{a} p_i log_2 p_i \tag{4.3.2}$$

With a being the number of nest options utilised, p_i the proportion of members of the society who chose option i, and H_{MAX} corresponding to the maximum possible entropy of the considered choice scenario. This H_{MAX} case is when all sites are chosen equally.

4.3.1 Speed-Accuracy Trade Off

Colonies of *Temnothorax albipennis* are capable of trading the speed of their decisions on the nest they wish to emigrate to, with the accuracy with which they choose the best nest [26]. Typically we see this trade off displayed under harsh emigration conditions. When conditions are good, the colony prioritises emigrating to a high quality nest, taking more time in order to ensure that the new nest is the optimal choice. However, when conditions are bad, the ants prioritise the speed of the emigration to ensure colony safety; sacrificing the accuracy of their nest selection.

The mechanism with which this has been suggested to take place is by varying the quorum threshold. Lower quorums mean a smaller number of ants need to accept a potential new nest. In turn this will mean that social carrying begins more quickly, so minimizing the time taken to move to a new site.

However, on lowering the quorum, there is more room for the colony to make a collective error of judgement. As fewer ants need to accept a new nest, this leads to a higher probability of quorum being exceeded at a non-optimal nest. Interestingly, this is a reflection of the ant colony behaving more individualistically as conditions become harsher.

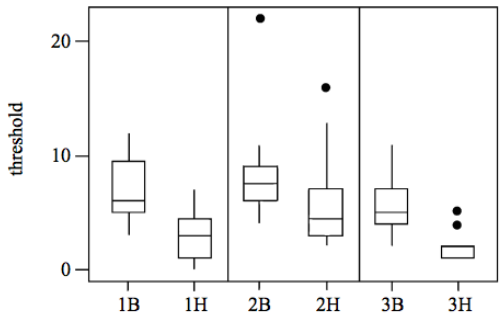


Figure 6: Quorum threshold box plots. Three experiments were carried out $(1, 2, 3)$, each under either benign (B) or harsh (H) conditions. We see that under benign conditions the quorum threshold is higher, increasing the accuracy of the ant emigration. (Image sourced from [26])

This capacity for an ant colony to exhibit a speed-accuracy trade off is of particular interest to researchers in computer science due to a field known as *anytime algorithms*; algorithms that generate higher quality answers the longer that they are left to run [13, 95]. Using these algorithms, if you were only interested in an approximate solution you would leave the algorithm to run for only a short period. If you were more interested in accuracy you could leave it to run for longer. The ability of an ant colony to trade the speed at which it selects a new nest with the accuracy of its interpretation of its quality is analagous to this type of process [49]. By studying the ability of ants to make their own speed-accuracy trade off, it was thought that we could learn lessons to employ within this field.

4.3.2 Speed-Cohesion Trade Off

The speed-cohesion trade off is where colonies are capable of trading speed for cohesion by altering the quorum threshold, where lower quorums mean a lower cohesion [29]. The relationship between quorum and cohesion is due to the speed at which ants begin social carrying to each nest. If the quorum threshold is low then carrying will begin rapidly to different nests; splitting the colony between several sites. However, if the quorum is higher, then it will take longer for ants to accept a nest, and the probability of this splitting will be smaller.

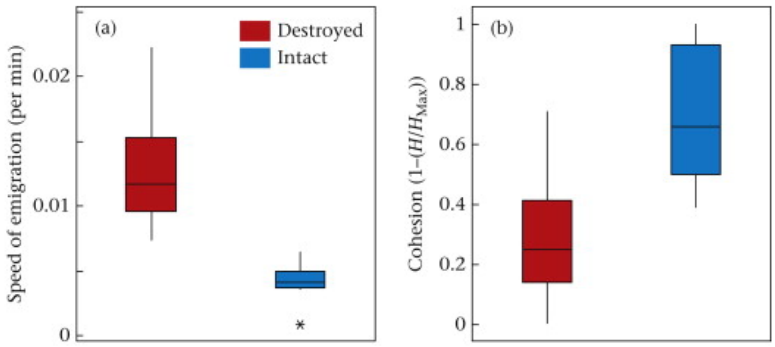


Figure 7: Speed versus cohesion box plots. Speed is defined as one over the emigration time defined earlier. Sixteen emigrations were recorded where the original nest was destroyed (*red*) and sixteen where the original nest was intact (*blue*). We see from the results that as speed became more important in the emigration, cohesion suffered. (Image sourced from [29])

4.4 Biological Background Summary

The biological background to the project covers a number of important points to be used going forward. Initially an understanding of the different roles ants play during an emigration and the function of the quorum threshold will be fundamental to understanding the models discussed in *Section* 5.2. The speed-accuracy trade off displayed by emigrating colonies is significant as it helps explain why the decentralized decision making process demonstrated by *Temnothorax albipennis* is studied in the field of computer science. A grasp of the speed-cohesion trade off is then integral in understanding one of the weaknesses of the Pratt model covered in *Section* 5.2.1.

We now define and discuss the models and model types that will be used within the project.

5 Modelling Background

Within this section we will discuss the necessary modelling background for the project. This will focus on the four models of ant population dynamics: Pratt (*Section* 5.2.1), Planqué (*Section* 5.2.2), AH-HA (*Section* 5.2.3) and SPACE (*Section* 5.2.4). We will start with a discussion of the different properties that the models can display: *agent-based, differential equation-based*, and *spatial*.

5.1 Types of Model and their Properties

Agent-based, spatial, and *differential equation-based* are three properties of models central to the project. Going back to one of the goals of the thesis, we look to explore the application of equation discovery to the problem of standardising models from different forms to one consistent form. The problem we consider is to start with two agent-based models and look to derive their differential equation-based summaries. One of these models, SPACE, is different to the others as it holds a spatial element, whilst the remaining models do not. Within this section, we discuss the relevant literatures in order to explain what these model properties entail and what their comparative advantages are.

5.1.1 Agent-Based Models

We begin with agent-based models [33]. Agent-based infers that each ant is treated as a separate entity and is given its own set of rules to employ during the emigration process. At any one time, an ant will be classified as a discrete individual and can be categorised as being in one of the distinct emigration states. Both the AH-HA and SPACE models fall into this category.

Agent-based models bring a number of benefits. Primarily they are easy to reconcile with biological observations, since drawing a comparison between the behaviour of an individual ant in an agent-based simulation and in real life is relatively simple. This method of modelling is also a lot more intuitive for our given application. We think of ants as individuals; each with a set of rules that they follow regarding their current state and how they transition to other states. It makes most sense to model them as such and let the more complex colony level behaviour emerge as a result of the combinations of the actions of individual ants, rather than to aim towards modelling this complicated behaviour directly.

5.1.2 Differential Equation-Based Models

Differential equation-based models consider the change in the number of ants fulfilling each role through time as a continuous function. However, by choosing to do this we can have fractions of ants in an emigration role. We compare this to agent-based modelling, where we can only have a discrete number of ants per role at any given moment, which more accurately represents meaningful biology.

Having said this, differential equation-based models do harbour a number of very useful properties. They are easy to construct, analyse and run, and their form means that they can be used to derive deterministic properties analytically. In addition, they are widely used and understood, whilst being deeply entrenched within modern biology.

5.1.3 Spatial Models

Finally we cover spatial models. Spatial refers to the fact that the model includes some notion of physical location. In the context of the project, the Pratt, Planqué and AH-HA models see the change in number of ants per role through time as a process distinct from the ant's spatial location. The only exception within our models is SPACE.

In the case of SPACE, a spatial element adds an extra dimension and makes the simulations feel more realistic. As SPACE is both agent-based and spatial; visual simulations of the model will *look* like an ant emigration and so facilitate a quick cursory method of analysis.

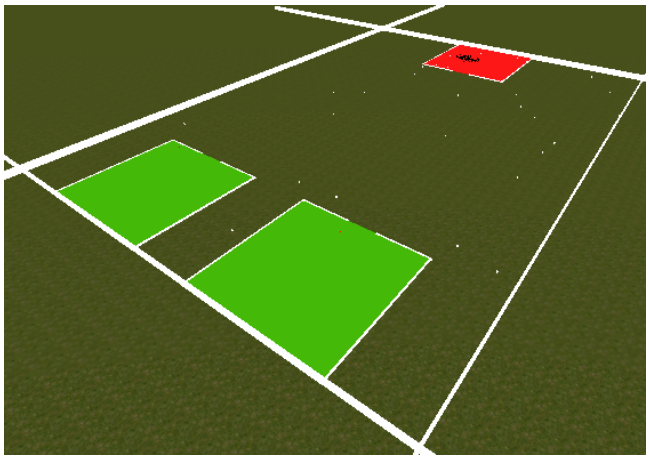


Figure 8: 3D visualisation of the SPACE model. The white particles represent active ants, the black particles passive ants, the red area the starting nest and the green areas two equidistant new nests.

In addition to this, there could be information crucial to understanding the population dynamics which is well represented by ants' physical location. Neglecting to include location within a model might lead to overly complex solutions which could be simplified with the inclusion of spatial information.

5.1.4 Model Types Summary

These explanations of the merits of different model properties offer some insight into the added value of the project. Currently we have one model, SPACE, which offers both the benefits of spatial and agent-based models, and an ongoing aim in the fields of computer science and biology is to define a system of deriving differential equation-based summaries from its data. This will allow us access to at least one model with the potential to exploit the properties of agent-based, spatial and differential equation-based models. The work done within this project then acts to inform future research relevant to that area.

To summarise this section, a table has been included; detailing which models hold which properties. This can be used for reference throughout the project as this subject is considered regularly and in a variety of contexts.

Model	Differential Equation-Based	Agent-Based	Spatial
Pratt	✓		
Planqué	✓		
SPACE		✓	✓
AH-HA		✓	

Table 1: Table detailing the properties of the existing models.

5.2 Existing Models of Ant Population Dynamics

Having covered the relative strengths and weaknesses of the different types of model, we define and discuss the four models that will be used within the project.

5.2.1 Pratt

Originally introduced in order to examine the role of tandem running and transport in colony emigrations, the Pratt model [71] is the first of the two differential equation-based models we will examine.

Pratt is based on a scenario where the colony's previous nest has become uninhabitable and they must emigrate. There are then a series of parameters: M, the number of sites of varying quality available, N, the total population of the colony including brood items and p, the proportion of ants in the colony who are active. Active ants are categorised in one of three roles; S, scouting for a new nest, A_i, assessing nest i and R_i, recruiting to site i. The model follows the set of equations:

$$\frac{dS}{dt} = -\sum_{j=1}^{M} \mu_j S - \sum_{j=1}^{M} \lambda_j I(R_j, S) \tag{5.2.1}$$

$$\frac{dA_i}{dt} = \mu_i S + \lambda_i I(R_i, S) + \sum_{j \neq i} (p_{ji} A_j - p_{ij} A_i) - k_i A_i \tag{5.2.2}$$

$$\frac{dR_i}{dt} = k_i A_i + \sum_{j \neq i} (p_{ji} R_j - p_{ij} R_i) \tag{5.2.3}$$

Where we can define the rate at which scouts discover site i as μ_i and the rate at which scouts are led to sites by recruiters as λ_i. Assessors then become recruiters at rate k_i and assessors and recruiters at site j encounter site i and switch their allegiance at rate p_{ji}.

Active ants are only recruited via tandem runs to site i if R_i is less than a threshold T, if $R_i \geq T$ recruitment switches from tandem runs to transports (social carrying). This is embodied in the function:

$$I(R_i, S) = \begin{cases} R_i & \textit{if } R_i < T \textit{ and } S > 0 \\ 0 & \textit{otherwise} \end{cases} \tag{5.2.4}$$

Which can be found in equations (5.2.1) and (5.2.2). The population of passive ants at site i, P_i, changes according to:

$$\frac{dP_i}{dt} = \phi_i J(R_i, P_0) \tag{5.2.5}$$

Where ϕ_i is the rate of transport to site i. J is used so that carrying only occurs at sites with at least T recruiters:

$$J(R_i, P_0) = \begin{cases} 0 & \textit{if } R_i < T \textit{ or } P_0 = 0 \\ R_i & \textit{otherwise} \end{cases} \tag{5.2.6}$$

Where P_0 gives the number of passive items remaining in the old nest.

Within this project, as within the majority of applications of the model [46, 49], we will consider the scenario where there exists two alternative sites, site one and site two. Site two is of superior quality to site one and both sites are of equal distance from the old nest. This means each nest is discovered at the same rate μ, and recruited to at rates λ and ϕ. We also assume that ants never switch from the superior nest to the inferior nest and so $p_{21} = 0$.

The paper provides a table of parameter estimates derived from their observations.

Parameter	Definition	Estimate	SD
N	Colony population (including brood items).	208	99
p	Proportion of colony population consisting of active ants.	0.25	0.1
λ	Rate of recruitment via tandem runs, per ant.	0.033 tandem runs/ min	0.016
ϕ	Rate of recruitment via transports, per ant.	0.099 transports/min	0.02
μ	Rate of discovery of new sites, per ant, per site.	0.013 min^{-1}	0.006
k_1	Probability per min that an assessor at site 1 begins to recruit.	0.015	0.06
k_2	Probability per min that an assessor at site 2 begins to recruit.	0.02	0.008
p_{12}	Rate of switching allegiance from site 1 to site 2 per ant.	0.008 min^{-1}	0.004
c	Time to move an item from site 1 to site 2 after old nest is empty.	4.6 min	-

Table 2: Parameters for the Pratt model.

From the definition of the model we move to a discussion of its properties. It has been shown that it is very sensitive to one of its switching probablities, p_{ij} and that increasing the switching probability can lead to perfect accuracy with a quorum size of one [49]. Also, the assumption that is made that ants never swap from a superior to an inferior site ($p_{21} = 0$) is not consistent with observations.

On constructing and running the model, we have found that, for low quorums, it does not function effectively. To explain this, we refer back to the notion of cohesion in the colony (*Section* 4.3.2). The higher the quorum, the more ants need to accept the nest before carrying begins and the smaller the margin for error for choosing the non-optimal nest. The lower the quorum, the higher the possibility of the colony beginning social carrying to a lower quality nest and so the higher the probability of the passive items being carried to different nests; splitting the colony.

The model does not deal with the eventuality of a split population well. Once the passive items become split over two potential nest sites, it contains no real method of simulating the recruitment from the lower quality nest to the higher quality one. Instead it relies on simply moving brood items from the lower quality to the higher quality site at a constant rate. This oversimplifies the real behaviour and so, for low quorums, damages the realism of the model.

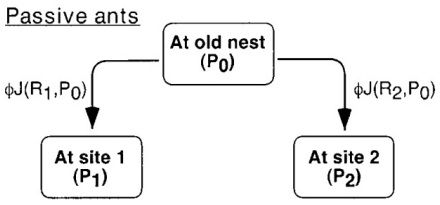
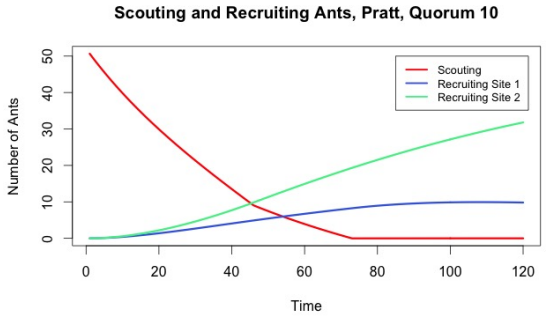


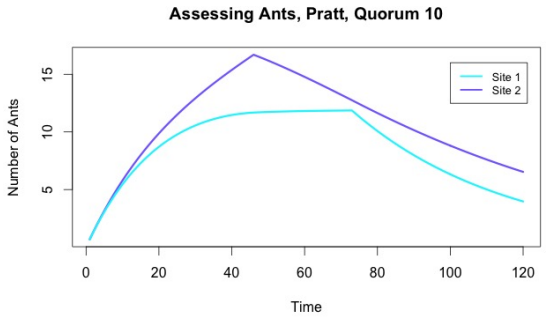
Figure 9: Flow chart representing the Pratt model's approach to the emigration of passive items. Within the model we see that there are defined paths from the original site to the two new nests. However, there is no route between the two new sites. The model moves passive items between the sites at a constant rate, oversimplifying the real world behaviour. (Image sourced from [71])

Apart from these low quorum emigrations, we saw Pratt modelled the expected ant behaviours reasonably well. To demonstrate the model it was run with a quorum size of ten and the following figures were produced (implementation details can be found in *Appendix* C.3).

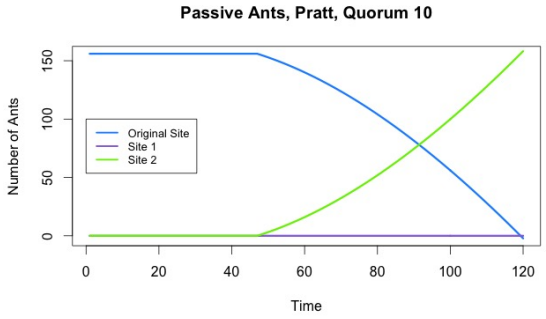
Scouting and Recruiting Ants: At the start of the emigration, a large number of ants are scouting for new nests. This dies away as they begin to accept nests and recruit to one of the two sites. We see that the majority of ants accept the higher quality nest (site two) and begin recruiting. We would expect the proportion of ants recruiting to the higher quality site to increase with the quorum.



Assessing Ants: After a short period ants begin to find the two sites. We note that both sites are found at approximately the same rate, as would be expected as both are equidistant to each other. The reason for the higher number of assessments at site two is due to the fact that some ants will switch allegiance from site one, then must assess site two before beginning recruitment to the nest.



Passive Ants: The emigration we are currently considering has a relatively high quorum, where the cohesion is also high. In the previous figure we saw the number of recruiters rising rapidly for site two. This can then be seen reflected in these results from the Pratt model, where at around time fifty the quorum for site two is reached. From this point the number of passive items at the superior quality nest begins to rise rapidly as social carrying begins.



Within the model we see the number of passive ants, recruiters and assessors rose most rapidly at the superior site, and for higher quorums passive ants were carried almost exclusively to the site of best quality, which matches what we would expect to see from *Section* 4.3.2, where we discussed the relationship between quorum level and colony cohesion. Overall this reinforces our confidence that it is a realistic model of ant behaviour.

5.2.2 Planqué

The Planqué model was introduced in order to explore the role of reverse tandem running in colony emigrations [66]. In contrast to Pratt, it only considers emigrations with one new nest site. The defining equations are as follows.

$$\frac{dA}{dt} = -\mu A - l(\lambda, R, Q, A) \tag{5.2.7}$$

$$\frac{dS}{dt} = \mu A - kS - fr(\lambda, R, Q, S) \tag{5.2.8}$$

$$\frac{dR}{dt} = kS + l(\lambda, R, Q, A) + fr(\lambda, R, Q, S) \tag{5.2.9}$$

$$\frac{dP}{dt} = -(1-f)c(\phi, R, Q, P) \tag{5.2.10}$$

$$\frac{dC}{dt} = (1-f)c(\phi, R, Q, P) \tag{5.2.11}$$

Where A, S, R, P and C are the number of ants active in the old nest, scouting, forward tandem run recruiting, passive and being carried respectively. Q is the colony quorum threshold. The parameter μ is then the rate at which active ants at the old nest begin scouting, k is the rate at which scouts become recruiters, and ϕ the rate at which passive ants and items are carried to the new nest. Forward tandem running takes place at rate λ, and f is the proportion of post-quorum time spent on reverse tandem running. We also define:

$$l(\lambda, R, Q, A) = \begin{cases} \lambda \frac{RA}{R+A} & if\ R < Q \\ 0 & otherwise \end{cases} \tag{5.2.12}$$

$$c(\phi, R, Q, P) = \begin{cases} \phi \frac{RP}{R+P} & if\ R \geq Q \\ 0 & otherwise \end{cases} \tag{5.2.13}$$

$$r(\lambda, R, Q, A) = \begin{cases} \lambda \frac{RA}{R+A} & if\ R \geq Q \\ 0 & otherwise \end{cases} \tag{5.2.14}$$

The justification behind the $\frac{AB}{A+B}$ form of the equations is that if there are two populations of ants with sizes A and B, the number of ants that on average meet is proportional to $\frac{AB}{A+B}$. Similar to the Pratt model, a table of parameter values is provided.

Parameter	Definition	Value/ Range
N	Colony size	250
F	Fraction of active ants	$[0.05, 0.5]$
Q	Quorum Threshold.	$n.a.$
f	Fraction of post-quorum reverse tandem running time.	$n.a.$
μ	Rate at which active ants at old nest become scouts ($ants^{-1}min^{-1}$).	$[0.01, 0.2]$
λ	Rate at which ants following tandem runs become recruiters ($ants^{-1}min^{-1}$).	0.1
ϕ	Rate at which passive ants are carried to a new nest ($ants^{-1}min^{-1}$).	0.2
k	Rate at which scouts independently become recruiters ($ants^{-1}min^{-1}$).	$\{0.0001, 0.001\}$

Table 3: Parameters for the Planqué model.

The initial conditions for the model are set as $(A, S, R, P, C)(0) = (FN - 2\epsilon, \epsilon, \epsilon, (1-F)N, 0)$. The parameter ϵ is used in order to prevent division by zero in l, c and r, and is chosen to be 0.01.

The Planqué model represents the behaviour of ant colonies well. It is capable of displaying certain properties of emigrations such as the speed-accuracy trade off discussed in *Section* 4.3.1, and also correctly predicts that under emergency conditions ants should not spend time recruiting via forward tandem runs. Instead the model proposes a colony should use a low quorum and spend a proportion of their time recruiting scouts using reverse tandem runs, increasing the recruiting population and speeding up the emigration [15].

One of the major points that should be raised in relation to the Planqué paper is that it is theoretically based. No experiments were run by the authors, and any data used was taken from preceding literature. Specifically, the equations were derived using theory with minor influence from the data of *Mallon et al* [46], and the values and ranges of parameters were based on the papers from *Pratt et al* [68, 71].

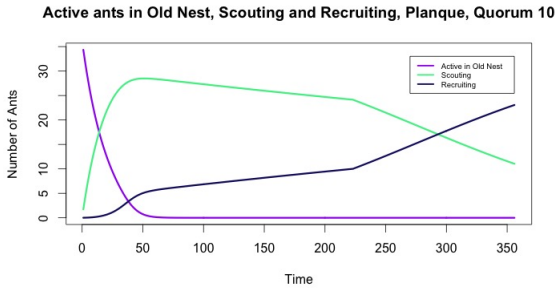
It could be argued that there are both strengths and weaknesses to this approach. Deriving the equations from known theory prevents any extraneous influence from experimental data. One of the potential failings of basing a model heavily on observations can be the temptation to interpolate to the data, and then claim that it is a good model as it fits the observations well.

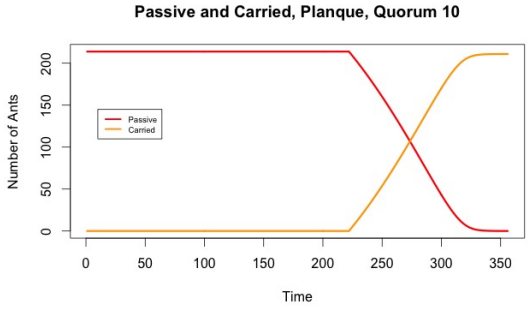
Conversely, a totally theoretical approach is of no use if it does not represent what occurs in reality. The use of the Pratt data to justify parameters for the model is a reasonable way of mediating this. The strategy of defining a sound looking model using theory, then using experimental data to set parameters so it mimics realistic behaviour seems sensible. It would potentially be possible to have qualms with their use of second hand data, as they have no first hand experience of how it was collected, or of its potential failings. However, as the model has been evaluated to have been successful in displaying realistic ant behaviours, this seems unnecessary.

To demonstrate the model figures were generated by running Planqué with the parameters specified in the original paper [66], and quorum size 10 (implementation details can be found in *Appendix* C.3).

Active in the Old Nest, Scouting and Recruiting Ants: Initially we see the number of active ants in the old nest is high. These ants then transition into the scouting state as they leave the nest. As time continues more ants transition into the recruiting stage as they find nests of a suitable quality.

Passive and Carried Ants: At a time between 200 and 250 time steps into the emigration the quorum is reached. In the plot the number of passive ants in the the original nest begins to decrease as they are carried to the new nest.





In summary, the Planqué model is successful in exhibiting the expected population dynamics. The assessing, scouting and recruiting ants all interact as defined by the literature. In addition the point at which quorum is reached is well represented, as the number of ants being carried from the old nest to the new site rapidly increases.

5.2.3 AH-HA

AH-HA [49] was the first model that attempted an agent-based approach, stepping away from the differential equation modelling tactic employed by Pratt and Planqué. It was the study of one of the emergent properties of the population dynamics of *Temnothorax albipennis*, the speed-accuracy trade off (*Section* 4.3.1), that inspired the creation of the model.

The paper AH-HA is based upon includes an argument for the validity of agent-based modelling within ant population dynamics. Encouragingly, this resonates heavily with the one offered in *Section* 5.1.1. It notes that it is unrealistic to model discrete entities as continuous, as occurs in the differential equation-based models. It also argues that this approach can lead to qualitatively different dynamics and equations must become more complex to capture the variety of emergent properties displayed by ant colonies. It is far more intuitive to define simpler rules for individual ants, and let the colony-level complexities emerge from those, than to try to capture them all within a set of equations, which represent every single one of the sophisticated global behaviours simultaneously.

Within the model, two new nests are set up equidistant from the old nest, with ants leaving the old nest with probability s. Once it has left the nest, an ant will discover one of the new nests with equal probability and spend a time steps assessing it. The ant will leave with an interpreted quality of the nest, o_i, where:

$$o_i \sim N(Q_i, \sigma^2) \tag{5.2.15}$$

With Q_i being the defined quality of the nest and σ being assessment noise. The scout then delays before beginning recruitment. This delay is implemented by each time step having associated with it a probability, d_i, that the ant begins recruiting to the site, where:

$$d_i = \begin{cases} \frac{o_i}{Q_{MAX}} & o_i \leq Q_{MAX} \\ 1 & otherwise \end{cases} \tag{5.2.16}$$

Q_{MAX} is a defined upper bound for nest quality. From this relationship we can see that the time taken to begin recruiting to a new site is proportional to its percieved quality.

Whilst waiting to transition to the recruiting stage, the ant can consider the other nest with probability p. If the ant does decide to consider the nest, it spends a further a time steps assessing it; changing its preference if the quality of the site exceeds the previous one.

Once recruiting for a site has begun, the ant chooses between social carrying and tandem running by assessing the quorum size, q_i for the current nest. This is compared to a quorum threshold T, which is low if emigration speed is important, but is high otherwise. If $q_i > T$ then social carrying is used, otherwise the ant reverts to tandem running. Carrying takes place in c time steps and tandem running in r, where $r = 3c$.

The priority of recruitment is:

1. Inactive scouts in the original nest.
2. Recruiters willing to change recruitment allegiance from one nest to another. This willingness is determined by a probability of switching preference, p, with 0 representing never switching and 1 representing always switching. A willing recruiter is selected and immediately stops and assesses the new nest it is being recruited to. The ant then resumes its own recruitment if the new nest was perceived as a worse quality than the current one, otherwise it changes allegiance and recruits to the new nest.
3. Passive ants and brood items.

Once there are no ants left in the recruiting ant's original nest, it returns to its new home and may begin the scouting process again with probability s; otherwise it remains inactive in the new site. If the ant does begin scouting, it can only consider the other new nest, as the nest it originally came from is deemed uninhabitable by the algorithm.

However, if there are still ants remaining in the ant's original nest, then with probability p the ant will consider the alternative nest and spend a time steps assessing it. After the assessment, the ant will switch preference if its evaluation of its quality is higher than its currently associated nest quality.

This leads to the fact that an ant can return to the same site several times during an emigration. On returning, the ant has no memory of the quality of the nest and uses its most recent survey of it to define its perceived quality.

True to the biology, an ant becomes inactive once carried to a site and plays no part in the recruitment process. However, at each time step, the ant can become active again with probabilty s. If, on the other hand, it has been recruited via tandem running and accepts the site, then it delays time d_i, as defined earlier, before beginning its own recruitment.

The explanation so far has been solely in terms of active ants. We finish by covering passive ants, which are moved between the nests by scouts if there are no other scouts that can be recruited.

All of the above explanation can be found summarised in *Figure* 10.

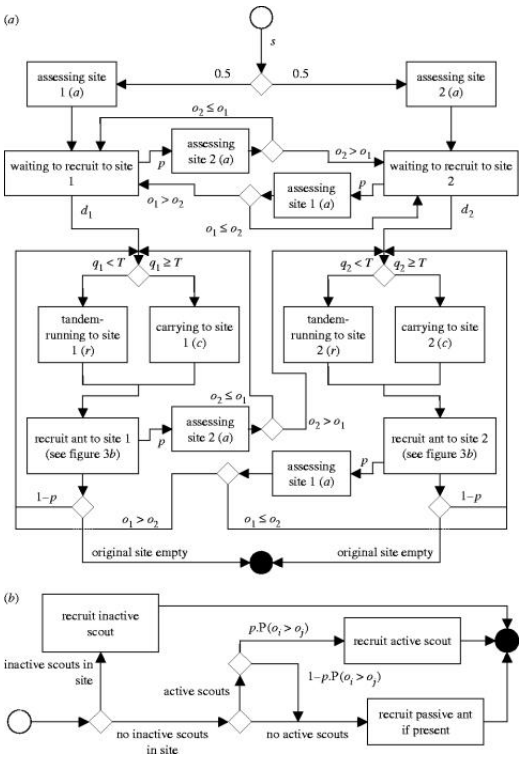


Figure 10: Flow chart representing the behaviour of ants in the agent-based AH-HA model. Section a) is the state transition diagram for active ants. The number of time steps per state is indicated where applicable, and the quantities associated with arrows are transition conditions or probabilities. Section b) is the recruitment strategy. Active and inactive are defined differently to previous. Active scouts are recruiting ants to their chosen nest site, inactive scouts are still scouting. o_i is the recruiting scout's assessment of their preferred nest quality (site i), o_j is the potential recruit's assessment of their preferred best quality (site j, where $i = j$ is possible). (Image sourced from [49])

From current publications we know there are a number of problems with the model. Primarily, it is non-spatial. Once ants find a nest, they essentially teleport to that location, without any exploration cost. This is not reflective of the real world process and underlines one of the failings of including no spatial information. Furthermore, the model was built with an explicit waiting time after nest assessment, which has been shown to be unrepresentative of true biology [75, 76]. In addition, by increasing one of the parameters of the model (the switching preference p) the optimal nest choice is made with 90% probability, even when the quorum is one [49].

This last property can be counteracted by increasing the noise and cost associated with nest decisions, after which an increase in p only increases accuracy at the cost of speed. This speed-accuracy trade off is an observed property of ant colonies, and so adds some weight to the AH-HA model's validity.

However, this ability to demonstrate the speed-accuracy trade off should be tempered with the necessity for the introduction of increased decision noise and cost. In the conclusion of the original AH-HA publication, it states that assessment noise and assessment cost are crucial to modelling the ant population dynamics. We know this not to be true as it is possible to recreate the same speed-accuracy trade off they claim requires this cost and noise, but by including spatial information and reverse tandem runs instead (refer to *Section* 5.2.4 on SPACE for details). This produces the same behaviour but using observed biological findings, not by altering parameters of the model.

5.2.4 SPACE

SPACE [78] is a model that has been built up over a number of years by students at the University of Bristol, and is the first model of *Temnothorax albipennis* population dynamics that attempts to build in a spatial element. It integrates the AH-HA decision process with ant navigation techniques taken from the Ant Box model [52]. The Ant Box model will not be discussed within the thesis, as all of the relevant information is contained in this section on SPACE.

The original aim of the SPACE project was to incorporate the latest biological research into a model with a high level of flexibility for investigating different experimental hypotheses. For example, the behaviour of reverse tandem running can be turned on or off in order to examine its role in the emigration process.

The explanations relating to SPACE will be a lot more high level than the ones for previous models, since the model is sophisticated and will require many pages to explain fully. The exact technical details of SPACE are also less important within this project, as we will not be dealing with them directly. Instead we will be looking to use it as an example in our investigation of equation discovery. The most auspicious tactic going forward will be to explain the model in terms of its higher level behaviours, and then refer the reader to *Developing A Spatially Realistic Simulation of Ant Colony Emigration*, Sampson, N. (2013) to find the full details.

Within SPACE, inactive ants do very little. If an ant is in the passive class, it waits to be recruited from the nest by an active ant. If it is in the active class, it makes occasional noisy assessments of the nest, and if this assessment yields a perceived quality above a certain threshold, the ant transitions to scouting. Once scouting, the ant uses a random walk mechanism to determine its movement.

Aside from employing a random walk, pheromones laid by previous ants are also considered in the ant's movement pattern. These pheromones act as an attractive force, which in light of the latest findings, could be considered controversial. It has been suggested that in fact pheromones have a repellent effect on ants when scouting for valuable resources [36]. This is due to the fact that they demonstrate where ants have already investigated, and so it is deemed to be more advantageous to the colony to investigate more uncharted territories.

Once a scouting ant is within a specified range of the entrance to a nest, it will begin to navigate towards it and, on entering the nest, it transitions to the assessing state. When the ant enters the nest

it begins a random walk around its interior until it finds the entrance again and leaves. When it leaves the ant receives a number between zero and one inclusive, representing its perceived quality of the nest. This number is drawn from a distribution:

$$Z \sim N(\textit{true nest quality, nest assessment noise}) \tag{5.2.17}$$

If this is greater than the ant's quality threshold then it begins to recruit to the nest.

On transitioning to the recruiting stage, an ant continually goes between its new and old nest recruiting until the old nest is empty. Recruitment is done via forward tandem running if the last visit to the nest resulted in a quorum measurement less than the quorum threshold; otherwise carrying is employed. When tandem running, the ant leads at a reduced speed and the follower follows. If the follower falls behind, then the leader waits for a set period of time for the following ant to catch up. If the ant does catch up, recruitment continues; otherwise the leader looks for a new ant to recruit.

A separate problem is then what happens should a recruiting ant encounter a nest it is not recruiting from. In this scenario, it assesses the new nest. If the assessment yields a higher perceived quality than the previous nest then it changes allegiance, otherwise it continues from where it left off.

The final point to be covered in this overview of the SPACE model is how it deals with interactions between ants. The only ants that actively interact with each other within the model are recruiters. There are two probabilities associated with these interactions, the first is a switching probability if an ant is tandem running (*tandRecSwitchProb*), and the second is a switching probability if an ant is social carrying (*carryRecSwitchProb*). The interactions of ants can be found summarised in *Figure* 11.

As only recruiting ants actively interact, SPACE also provides a mechanism by which ants avoid each other. When an ant encounters another directly in its path it selects an offset from its current direction, where the offset, Q, is distributed:

$$Q \sim U(\textit{-maxVar, maxVar}) \tag{5.2.18}$$

Where $maxVar$ is a parameter defined by the user. The use of a uniform distribution means that the change in path is not weighted in a particular direction.

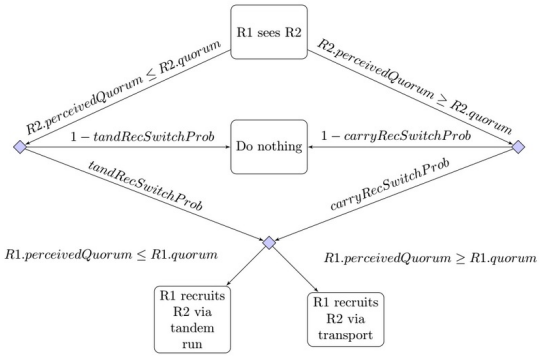


Figure 11: Flow chart representing how two actively recruiting ants interact. If a label does not contain a conditional expression then it represents one of the defined probabilities. The *perceivedQuorum* variable corresponds to the recruiting ant's last quorum reading, *quorum* is then the quorum threshold for the simulation. (Image sourced from [78])

This marks the end of the properties included in the first iteration of SPACE. Comparing the above to the explanation of ant population dynamics in *Section* 4, we see that it incorporates a lot of the known biology into a spatial and agent-based model.

Now we will cover how SPACE has evolved from its first iteration. *Decentralised Decision Making in Temnothorax albipennis Colonies*, Garrad, M. (2014) was a Master's thesis at the University of Bristol focused on augmenting SPACE [30] . In particular it looked to prove that, at the time, the model was not capable of displaying one of the emergent properties of ant colonies: an ability to trade decision speed for decision accuracy. It looked to improve SPACE by incorporating some of the missed findings from the latest literatures into the model, then prove that with these new features that SPACE could display the speed-accuracy trade off.

On evaluation the project had succeeded in its aims. At the time of writing the model now follows the prior logic, as well as including:

- **Reverse Tandem Running:** On a recruiter completing a social carrying act, it begins searching in the new nest for any inactive scouts. If one is found, it is led back to the old nest, after which both ants begin carrying back to the new nest.
- **Initial Scouting:** In the original model, within a few seconds of beginning the emigration all inactive ants became scouts. According to observations, the number of inactive ants transitioning to scouting should vary with the quorum [28]. Low quorums are inherent in emergency emigrations and so will lead to more scouts leaving the nest more rapidly, whilst the inverse is true for high quorums. This was then incorporated into SPACE.
- **Fixed Assessment Cost:** The method the original model used for ants' assessment of nests was based on a random walk carried out until, by chance, they left the nest. This results in large fluctuations in assessment times for different ants, which is unrealistic when compared to biological observations of ants using Buffon's Needle [47]. The random walk mechanism was replaced with a fixed assessment cost. In the current model, ants enter the nest, remain there for a period of sixty seconds, and then leave with a measurement of the nest's quality.
- **Frustrated Recruiters:** The final addition to the model was the notion of frustrated recruiters. In the original model there were cases in high quorum emigrations where the ant colony did not reach quorum, even when the best nest had been accepted by all the recruiters. This is because although the sufficient number of ants had accepted the nest, there were not enough of them present in the nest at any one time for ants to register the quorum being achieved.

  The implication of this is that recruiters would begin searching for scouting ants to recruit via tandem running; however, there would be none available. To combat this, frustrated recruiters were introduced. When recruiters cannot find an ant to tandem lead, they now return to the new nest and reassess the quality and quorum.

Now we move from the definition of the model, to a discussion of its properties. Primarily, we know SPACE to be a good model of the ant population dynamics we are interested in. This has been demonstrated in terms of the number of ants per state progressing through time, expert opinion, and the model displaying emergent colony properties [30, 78]. However, it does include some minor failings that it will be necessary to be aware of going forward.

Firstly, the necessity of including frustrated recruiters, a departure from observed biology, can be seen as one such failing. Secondly, although the model makes reasonable approximations of ant behaviour, it does not include some of the most recent findings. For example, there is no inclusion of the Buffon's needle nest assessment [47, 48] or of how ants display a sense of memory [9] or have been shown to use landmarks and edges in their navigation [69, 70].

Aside from omitting these biological details, SPACE also has one practical flaw. On occasion, emigrations can get stuck and have to be terminated when a nest wall gets between a leading and following ant during a tandem run.

To address this, the focus of one of the projects running alongside this thesis is to start incorporating these biological behaviours and to remove this bug [84]. Therefore, although they are weaknesses in the current model, as the model develops they will be removed.

The above would lead us to believe that SPACE is the stronger of the two agent-based models. It can display the emergent properties found in AH-HA and does so through incorporating valid biology. The inclusion of a spatial element also gives it an advantage in realistically modelling the exploration cost of ants scouting for new nests. These factors contribute to the motivation for this project. We know that as an agent-based model, SPACE demonstrates accurate ant colony behaviours. Therefore, we look to derive a differential equation-based summary of the model, so we can examine the equations for the properties of the model that demonstrate this. The findings of this project then contribute towards that aim.

5.3 Modelling Background Summary

To summarise this modelling background section we relate what has been discussed to what will be covered in the remainder of the thesis. Initially we outlined the different types of model: differential equation-based, agent-based and spatial. In doing so we gave insight into the motivation for the project, where we aim to derive models that can exploit the merits of the different model types.

Following this we defined the various models we will use throughout the project. In *Section* 7.2.2 we will use Pratt and Planqué to verify the most recent research on applying equation discovery to systems of differential equations. We will show that when there are systems of differential equations generating data it is possible to discover formulae accurately approximating the original equations' dynamics from the generated data.

In *Section* 8 we will use the SPACE and AH-HA models we discussed and attempt to derive their differential equation-based summaries, demonstrating that this is not possible using current equation discovery techniques. In *Section* 9 we will use SPACE in experiments attempting to stabilise models derived using Eureqa.

We now discuss equation discovery and the equation discovery systems related to the project.

6 Equation Discovery Background

Processes for generating, collecting and then storing data are becoming increasingly powerful and autonomous. However, the methods for taking this data and deriving analytical rules from it are nowhere near as sophisticated. There is an increasing demand for a complimentary field to automated data generation which takes this data and turns it into useful formulae or equations. The proposed answer to this demand is equation discovery. The underlying principle of equation discovery is to employ genetic algorithms to search through the data generated from a system, and to use this data to identify useful analytical relations associated with the dynamics of the investigated system.

This section will focus on a review of three different equation discovery systems. The first two, LAGRANGE [18, 19] and LAGRAMGE [86, 88] (*Section* 6.1.1) are related; with LAGRAMGE being a development of LAGRANGE. The last equation discovery system is Nutonian Eureqa [79] (*Section* 6.1.2) and is different in its approach to the previous two systems, whilst also being more developed. Within this project we implement Eureqa in order to explore its application to systems of differential equations, hence the coverage of Eureqa will be considerably longer than the discussion of LAGRANGE and LAGRAMGE.

These equation discovery systems were chosen as required background for the project as they serve as examples of the different approaches taken to equation discovery. LAGRAMGE and similar systems (GOLDHORN [39] and SDS [90]) use a strong declarative bias to shrink the search space of the equation discovery, whereas Eureqa is less restricted in the solutions it aims to find. Both systems have been used previously to discover systems of differential equations. Therefore the aim of this section is to give context to the work done within the project, and explore how it is unique in its aims.

6.1 Equation Discovery Systems

The following section of the thesis will be used to discuss each of the equation discovery systems we have selected. We begin with LAGRANGE and LAGRAMGE in order to discuss equation discovery systems that use a strong declarative bias. We then cover Eureqa, the most developed of the modern day equation discovery implementations and one that uses a free-form approach.

6.1.1 LAGRANGE & LAGRAMGE

LAGRANGE [18, 19] was developed in 1993 and is a system for discovering differential and ordinary algebraic equations involving more than two variables. It is both open source, and written in *C* [37]. The system functions by keeping all but two variables constant, and then running experiments that vary these values. By cycling through all of the possible variable pairs one can attempt to identify the relationships between the variables in the system.

The main failing for LAGRANGE is that it deals badly with noisy data. It would be possible to apply smoothing techniques, however this would only reduce the noise, not remove it entirely. In addition to this LAGRANGE has no way of removing redundant laws in an intelligent way (the need for this will be expanded on in *Section* 6.1.2 on Nutonian Eureqa).

LAGRAMGE [86, 88] is a development of LAGRANGE. The difference between the two is that LAGRAMGE uses context-free grammars to introduce a declarative bias and reduce the search space of the equation discovery. Within LAGRAMGE, the user defines a grammar for the expected solution. The equation discovery system uses this to look for relationships within the data by minimising an error metric between a function that uses the specified grammar, and the provided data. The process terminates after a certain time limit, or after the error has reached a certain threshold.

The main flaw with LAGRAMGE links to the use of a grammar to limit the search space of functions. This provides two obstacles. Firstly it requires specialist knowledge in the domain of the application in order to define a sensible grammar. Secondly, it is very limiting. If the solution does not fall in the defined grammar, then it will not be found.

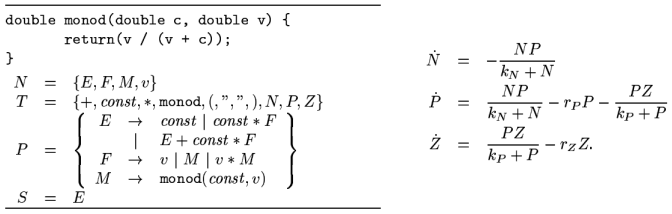


Figure 12: On the left we have an example grammar for the differential equation system on the right, to be used in the LAGRAMGE system. Within this project we aim to be able to define systems of differential equations without the necessity of defining a grammar beforehand. (Image sourced from [88])

LAGRAMGE, and similar systems with a strong declarative bias, have previously been successfully used in order to discover sets of differential equations. However, the tests run on these equation discovery systems have been with either pre-existent differential equations, or simple real world examples with well-defined grammars. Within this project, we consider deriving sets of differential equations from data of high complexities, with no known underlying equations, with limited domain specific knowledge with which to form a grammar. Therefore these types of equation discovery systems are not appropriate for our aims.

6.1.2 Nutonian Eureqa

Nutonian Eureqa [79] is the most developed of the current equation discovery systems and is regularly used in a variety of contexts; both academic and commercial [17,56,93]. It is based on an earlier system, BACON [40, 41], with LAGRANGE and LAGRAMGE also providing some of its basis. In terms of implementation, Eureqa is the most advanced system to date. It addresses many of the problems of previous systems such as needing to declare a grammar, or not being able to deal with noisy datasets, and is much more user friendly. Previously, it has been used to discover both single equations of various complexities, and also to rediscover known systems of differential equations.

Eureqa employs *symbolic regression* [38] to achieve its aim. The symbolic regression algorithm functions by forming initial expressions randomly from algebraic operators $(+, -\times, \div, ...)$, analytical functions $(sin, cos, tan, ...)$, constants, state variables and other mathematical functions and operators. New equations are then formed by recombining the existing ones and probabilistically varying their subexpressions. Equations that minimise a specified error metric are retained, and the process continues; terminating at the point when the measurement of error has reached a certain threshold, or after a certain time.

Within the paper on which Eureqa is based [79] several important factors relating to the algorithm are discussed. The first of these factors is the importance of finding *meaningful* and *non-trivial* invariants. In relation to the project, this speaks to what we have previously discussed about spatial information. The principle of equation discovery is to identify useful analytical relations associated with the dynamics of the investigated system. We can only discover these analytical relations given the necessary components to form them. Therefore, a failure to include a sense of spatial information could result in a failure to identify the relationship between that, and the population dynamics of the colony.

Conversely, we could also include too much information, and identify meaningless relationships. For example, if we were to take the function $f(y) = 1$ then we could identify the function f in several different forms:

$$f(y) = sin^2(y) + cos^2(y) \tag{6.1.1}$$

$$f(y) = y + 1 - \frac{yz}{z} \tag{6.1.2}$$

$$f(y) = ln(e^y)) - y + 1 \tag{6.1.3}$$

Each of which is equivalent to the true form, but with varying levels of complexity that could be simplified away. Eureqa deals with this by not presenting a single solution, but a small set of solutions that lie across an *accuracy-parsimony Pareto front*. *Parsimony* is defined to be the inverse of the number of terms in the expression and *accuracy* is defined in terms of validation data. This addresses one of the weaknesses of LAGRANGE which we saw in *Section* 6.1.1.

Something noted within the literature [79], and that has been witnessed in experimentation with Eureqa, is that this Pareto front contains a point where predictive ability jumps rapidly with some minimum complexity, after which accuracy only increases minimally with added complexity. When searching for solutions we will often aim for them to fall on this point, as it represents the optimal balance between accurate and complex solutions.

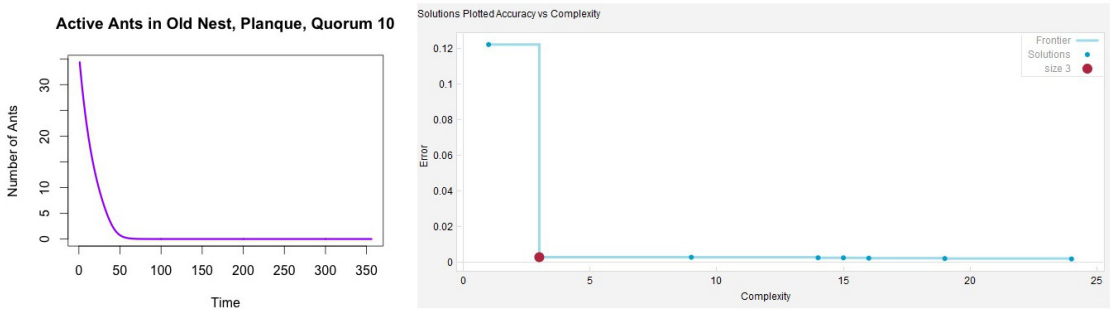


Figure 13: The accuracy-parsimony Pareto front for rediscovering one of the Planqué model equations from its simulated data. On the left is a plot of the number of ants active in the old nest through time. On the right is the plot of error versus complexity for the discovered solutions. Highlighted with a red dot is a solution with optimal complexity and error.

Another useful facet of the paper on which Eureqa is based is the methodology employed when using the algorithms. Within the publication they talk at length about how they validated Eureqa using two systems, an air track oscillator, and single and double pendulums.

Single and double pendulums can be justified as a reasonable testing choice, as within these two similar experiments it can be demonstrated that the algorithm can discover both simple and complex equations. The harmonic motion generated by single pendulums is a well understood, simple dynamic. However on attaching a second pendulum we transition into the realm of chaos after reaching certain energies.

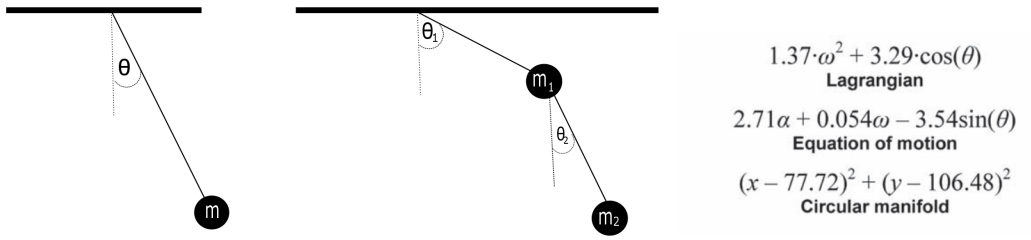


Figure 14: On the left is the single pendulum. The θ angle is used in equations of motion along with the mass, m of the ball. On moving to the double pendulum we introduce another mass and angle, creating chaotic dynamics at higher energies. On the right are examples of equations discovered using Eureqa to model the single pendulum. We look to see if we can extend the application from modelling single variables to modelling systems of differential equations. (Equations taken from [79])

Their experiments underline a few key points relating to the project work. Initially, they discuss how, dependent on the form of the data, the system will discover different laws. Contextualising this in terms of the pendulums; given position and velocity, the algorithm discovered the Hamiltonian and Lagrangian energy equations. Given the acceleration data, it discovered the differential equation of motion corresponding to Newton's second law. Finally, given only position data, it discovered the equation of a circle for the single pendulum (the circle is due to the pendulum being confined to this trajectory).

This relates to the project in that the form of the data given to the algorithm will determine its output, therefore it will need to be picked carefully. The form of the data will be discussed in detail in *Section* 8.1.1 on deriving the differential equation-based summaries of SPACE and AH-HA, but has been noted here as a point of interest.

What it is also important to note is the form of the experiments Eureqa is shown working with. In the majority of cases they use the system on single, complex equations such as those for the pendula. Eureqa is primarily designed for these types of application, and excels at finding equations for modelling individual quantities. One exception is the application of Eureqa to analytical models for metabolic networks, where they mirror our aims of using Eureqa to define systems of differential equations [80].

Within this experiment, they consider a system of differential equations associated with a glysotic oscillation model. They show that, using a symbolic regression based-system, it is possible to discover very good approximations to the underlying equations for the model using their generated data.

In terms of the project this shows that, when systems of differential equations are used to generate data, it is possible to discover very close approximations to those equations. In *Section* 7.2.2 we aim to reinforce this impression by demonstrating Eureqa can discover approximating equations for the Pratt and Planqué models from their generated data.

This also marks the boundary of modern research into discovering systems of differential equations using equation discovery. In methods both with and without a strong declarative bias, equation discovery has been applied to data from known systems of differential equations or simple real life examples. One of the contributions of the project will be extending past this and into trying to derive systems of differential equations from complex data generated by processes with probabilistic components.

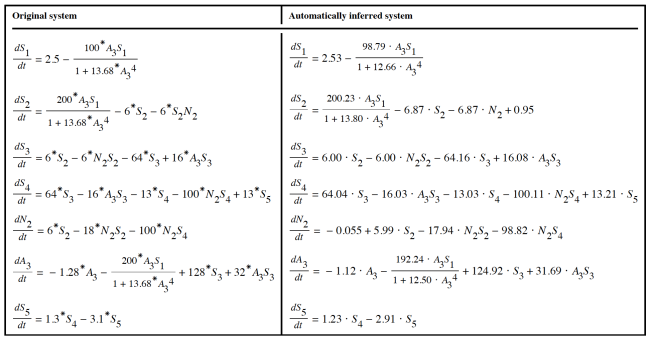


Figure 15: Previously the Eureqa system has been used to rediscover systems of differential equations from their generated data. On the left we have a system of differential equations representing a glysotic oscillation model. On the right we have the system of equations Eureqa discovered representing the model. We see that, analytically, they are very close. (Figure taken from [80])

The final piece of information it is possible to take away relating to the Eureqa system is the findings in simulations when they did not give the algorithm the correct building blocks. Since the algorithm incrementally builds solutions from initial combinations of mathematical operators and functions, if the correct operators and functions are not presented to the algorithm, it will struggle to find the most accurate solution.

On testing this, they note that the algorithm can develop reasonable approximations to functions. For example we can expand *sin* into its Taylor series:

$$sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + ... \tag{6.1.4}$$

If we were to look for a solution from the algorithm which included *sin* but did not give it the necessary building block, then it may approximate the function using this. Being aware of approximations and equivalent forms will become useful in our own employment of the algorithm.

6.2 Equation Discovery Background Summary

We conclude this section on equation discovery by relating its content to the rest of the project.

Initially we covered the LAGRAMGE and LAGRANGE systems. In part this was to demonstrate the difference between systems with a strong declarative bias and those without. In addition, in the evaluation section we will present arguments that defining a context free grammar for solutions could allow us to derive more stable systems of differential equations from data (*Sections* 12.1 and 13.2). This could be achieved by limiting the solution search space to solutions with a more stable grammar. These arguments will lead to experimenting with LAGRAMGE being one of the recommend courses for future research (*Section* 14.2).

Nutonian Eureqa will then form the focal equation discovery system for the project, and will be used throughout the following section on project execution. The information surrounding the optimal point on the accuracy-parsimony Pareto front will be used in *Section* 7.2.1 to help us to choose solutions from the front. In *Section* 8.1.1 we will use what we have learnt about defining solution types using the type of data supplied to the algorithm to derive the correct form of solutions from the agent-based models using Eureqa.

We now move from the project's background to its execution.

Part III
Project Execution

We now revisit the aims and objectives of the project. Currently we are in possession of four models of ant population dynamics: Pratt and Planqué, which are differential equation models, and AH-HA and SPACE, which are agent-based. The goal of the thesis is to explore equation discovery applied to deriving differential equation-based summaries from the two agent-based models. We break down this objective into the following steps:

1. **A preliminary analysis of Nutonian Eureqa.**
2. **Identifying, demonstrating and rationalising the current weaknesses in equation discovery applied to discovering systems of differential equations from complex data.**
3. **Experimentation attempting to address these weaknesses.**

1. (*Section* 7) As discussed in *Section* 6.1.2, the examples seen in the paper on which Eureqa is based center around the system's application to single equations. We initially run tests to verify how accurately Eureqa can discover single equations from data, extending the tests to cover simple, non-differential equation systems. Finally we attempt to apply Eureqa and discover approximations to the Pratt and Planqué models. On discovering these approximations this will bring the work done within the project in line with the most recent work in the field, where close approximations to well-defined systems of equations (similar to Pratt and Planqué) have been discovered using Eureqa.

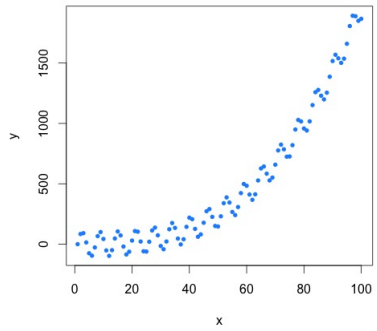


Figure 16: In *Section* 7 we aim to discover simple functions such as this polynomial with an embedded *sin* term. We also analyse how we can extend Eureqa to cover systems of differential equations.

Included within this section will be the methodology used to derive systems of differential equations from data. The results from this part will then pave the way for the following section, where we attempt to derive differential equation-based summaries from the agent-based models.

2. (*Section* 8) In the previous part we looked to discover systems of known differential equations from their numeric data. We now seek to apply the same techniques to discover systems of differential equations where there is no known underlying system. For this project, this manifests itself in discovering the differential equation-based summaries of the agent-based models. In this section we identify a weakness in Eureqa, and reason as to why this weakness exists, exploring the relationship between data complexity, equation complexity and model stability.

The work done within this part of the project will then motivate the following section where we attempt to address the failings of Eureqa.

3. (*Section* 9) This section reconciles the successes and failures of Eureqa in the previous two parts and turns them into a set of experiments aiming to address the weaknesses of the system. These experiments will then be used to motivate the further work in the area.

7 A Preliminary Analysis of Nutonian Eureqa

This section will first delineate the methodology for the preliminary analysis of Nutonian Eureqa (*Section* 7.1). Following this will be the implementation, results and evaluation of the analysis (*Section* 7.2), and a discussion of the findings (*Section* 7.3).

7.1 Analysis Methodology

In order to analyse Nutonian Eureqa, it is proposed to define functions of varying complexities, generate data from them, add Gaussian noise of differing degrees and look for Eureqa being able to rediscover the underlying functions. We define rediscovering the functions as either finding their exact analytical form, an equivalent analytical form, or a function that has a satisfactorily small margin of error between itself and the generated noisy data.

The methodology will be to run equation discovery, and initially look for it finding the exact form of the function used to generate the data. If this does not occur, the solution the discovery derives from the data will have the Mean Squared Error (MSE), Mean Absolute Error (MAE) and R^2 Goodness of Fit measured between itself and the original function (for definitions see *Appendices* D.2.1, D.2.2 and D.2.3 respectively). These measurements of error will be used to reason whether or not the equation discovery has been successful. As noise has been added it makes sense to be forgiving and allow the algorithm to discover functions that do not have the same exact form, but accurately model the data. This will act as a generic first test of the software, and give us confidence in Eureqa being able to approximate basic functions.

We then move to apply Eureqa to systems of differential equations using the Pratt and Planqué models from *Sections* 5.2.1 and 5.2.2. We analyse the ability of Nutonian Eureqa to rediscover these models for both fixed parameter values and in their general form. This is achieved by generating data from the system, then using the algorithm to rediscover the models. Successful rediscovery is defined as in the previous case: deriving the exact form of the original equations, or an equivalent solution with satisfactorily good fit and minimal error. If the algorithm is successful at this stage, this adds evidence to the findings seen in *Section* 6.1.2 showing Eureqa is capable of discovering systems of well-defined differential equations, and brings the project in line with the most recent work in the field.

7.2 Analysis Implementation, Results and Evaluation

This section covers the implementation, results and evaluation of the testing and will be split into two parts. The first part will deal with the results from the initial, basic testing functions. The second will be used to cover the results from testing Eureqa with the Pratt and Planqué models.

7.2.1 Basic Testing Functions

Implementation details for this section can be found in *Appendices* C.1 and C.2. Throughout this section, we will denote the noise added to functions:

$$Q \sim N(0, \sigma^2) \tag{7.2.1}$$

Where the variance of the noise, σ^2 will be given separately for each function. As a starting point, each test will use building blocks of constants, input variables, addition, multiplication, division and subtraction. Additional components will be determined by inspection. For example, if it is reasonable to assume a function has trigonometric terms embedded within it, then they will also be fed to the algorithm. If the underlying function was found on the Pareto front, then it will be chosen as a solution. Otherwise the solution at the optimal point on the Pareto front will be chosen.

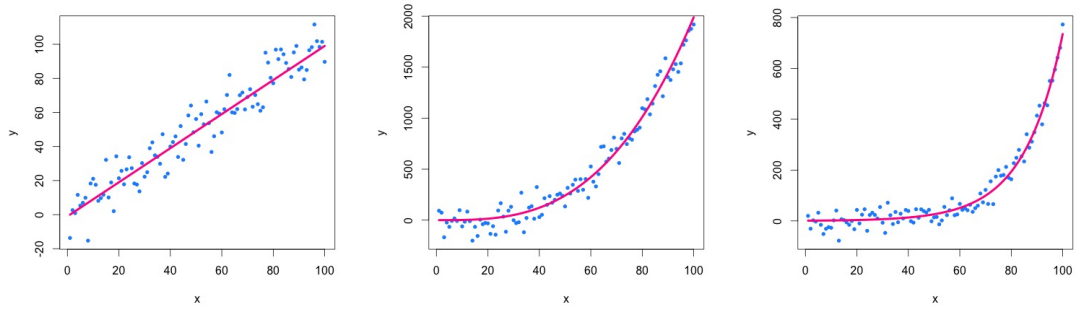


Figure 17: From left to right: $a), b), c)$. Blue points represent the underlying noisy data, pink lines are the Eureqa derived functions.

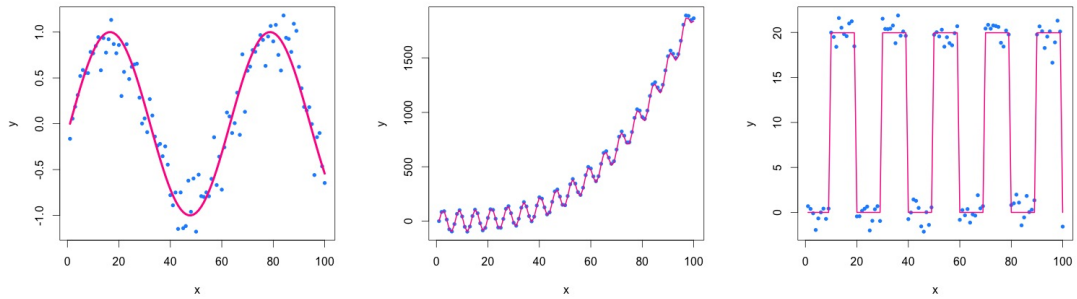


Figure 18: From left to right: $d), e), f)$. Blue points represent the underlying noisy data, pink lines are the Eureqa derived functions.

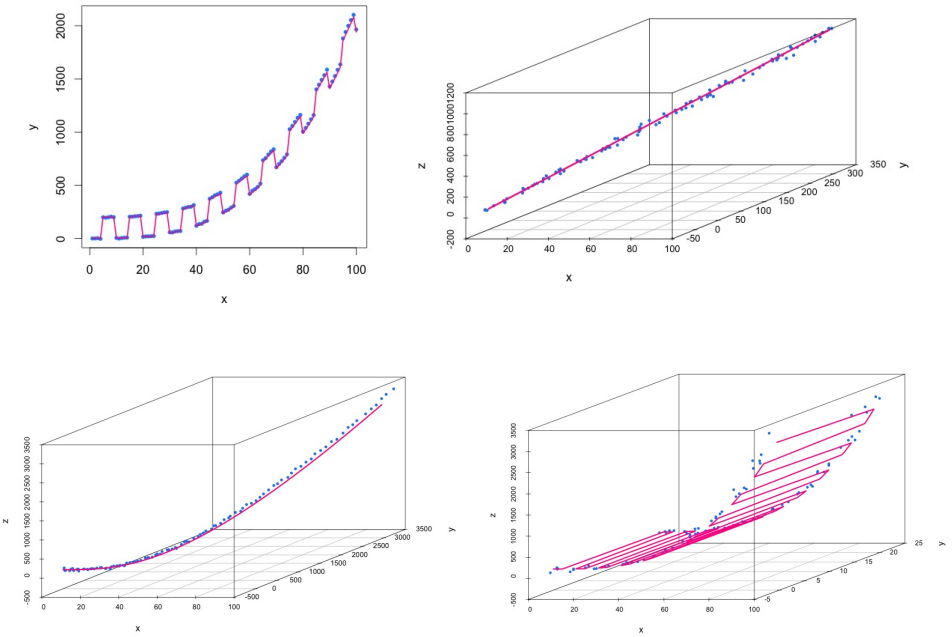


Figure 19: From left to right, top to bottom: $g), h), i), j)$. Blue points represent the underlying noisy data, pink lines are the Eureqa derived functions.

Function	σ^2	Original	Eureqa Derived
a)	10	$y = x + Q$	$y = x$
b)	100	$y = \frac{1}{500}x^3 + \frac{1}{400}x^2 + \frac{1}{100}x + Q$	$y = 0.00205 \times x^3$
c)	30	$y = e^{\frac{1}{15}x} + Q$	$y = e^{\frac{1}{15}x}$
d)	0.25	$y = sin(\frac{1}{10}x) + Q$	$y = sin(0.101x)$
e)	0	$y = \frac{1}{500}x^3 + \frac{1}{4000}x^2 + \frac{1}{1000}x + 100sin(x) + Q$	$y = 0.002 \times x^3 + 100.00549 \times sin(x)$
f)	1	$y = \begin{cases} 0 & 0 \leq x < 10 \text{ or } 20 \leq x < 30 \text{ or } ... \\ 20 & otherwise \end{cases} + Q$	$y = 19.96893 \times \mathbb{I}[sin(3.57221 - 12.25059 \times x) > 0]$
g)	1	$y = \frac{1}{500}x^3 + \frac{1}{4000}x^2 + \frac{1}{1000}x + \begin{cases} 0 & 0 \leq x < 5 \text{ or } 10 \leq x < 15 \text{ or } ... \\ 200 & otherwise \end{cases} + Q$	$y = 0.002 \times x^3 + 199.65703 \times \mathbb{I}[(sin(5.61195 + 11.93316 \times x) > 0]$
h)	20	$y = 3x + Q$	$y = 3.1x$
	20	$z = 4y + Q$	$z = 11.91289 \times x = 3.84287 \times y$
i)	20	$y = \frac{1}{500}x^3 + \frac{1}{400}x^2 + \frac{1}{100}x + Q$	$y = 0.00339 \times x^3$
	20	$z = \frac{1}{500}y^3 + \frac{1}{400}y^2 + \frac{1}{100}y + Q$	$z = 0.00339 \times y^3$
j)	20	$y = \frac{1}{500}x^3 + \frac{1}{400}x^2 + \frac{1}{100}x + Q$	$y = 0.00339 \times x^3$
	20	$z = \begin{cases} 0 & 0 \leq x < 5 \text{ or } 10 \leq x < 15 \text{ or } ... \\ 20 & otherwise \end{cases} + Q$	$z = 19.73796 \times \mathbb{I}[sin(-0.64033 \times x) > 0]$

Table 4: The underlying and Eureqa derived equations from the basic testing.

Function	MAE	MSE	R^2
b)	0.7626	0.8267	0.9762262
e)	0.1750	0.0417	0.9999999
f)	0.6614	0.7761	0.9923506
g)	2.6814	10.1637	0.9999707
i) y	0.9165	1.2482	0.9994732
z	0.7589	1.0405	0.9995623
j) y	0.8924	1.1590	0.9999833
z	0.6988	0.6374	0.9999993

Table 5: Normalised results for equation discovery applied to the basic testing functions. If the exact analytical form, or an equivalent analytical form has been discovered, the MAE, MSE and R^2 values are not given, as we know equation discovery has been successful. If a different function has been discovered, we include these values in order to reason around the results.

For many examples we can see that Eureqa has been able to find the exact underlying function (a, c, d, h). For the cases when Eureqa has not found the exact function, we give the R^2 and normalised MAE and MSE between the original function and discovered solution in *Table* 4. From these results it is clear that, even if it cannot find the exact analytical form of a function, Eureqa is capable of making very intelligent approximations.

When reviewing the errors and measurements of fit we note that, due to the noise added, we would expect the MAE to be proportional to the variance of the noise added to the underlying function, and the MSE to be proportional to the square of that. This has been taken into account and the MAE and MSE have been normalised by being divided by the variance and variance squared respectively. We then see many of the approximations made by Eureqa offer a reduction in error when considered proportionally to the variance. In the cases when it does not offer a reduction (g) it does offer a significant cutback in complexity, whilst maintaining the overall shape of the function. This is a side effect of our strategy of choosing solutions from the optimal point on the Pareto front, where we sacrifice error for complexity.

Within these approximations, we see that in the case of the functions containing polynomials, the system tends to take a polynomial containing only the highest order terms (b, e, g, i, j). This is due to the fact that the level of noise obscures the effect of the lower order terms. By doing this, the system captures the main shape of the function with minimal complexity.

The remaining approximations use trigonometric functions to approximate periodic behaviour (f, g, j). Although experiments were run using modulus building blocks as a method of capturing periodicity, sin, cos and tan offered solutions of equal or greater simplicity as the modulus approach, with the same level of accuracy (for details see *Appendix* A). Therefore in the remainder of the project we will use trigonometric functions to capture periodic behaviour.

Overall, we see that Eureqa is capable of accurately finding functions to model simple trends. We also see that Eureqa is capable of discovering simple sets of interacting equations (h, i, j). When it does not find the exact underlying function, it instead finds a very reasonable approximation. This gives us confidence in the basic ability of Eureqa. We expand on this in the following section where we examine Eureqa applied to systems of differential equations.

7.2.2 Rediscovering the Pratt and Planqué Models

This section will be used to cover testing Nutonian Eureqa on discovering systems of differential equations, and will be split into two parts. The first section will cover the testing of Eureqa on the Pratt model with fixed parameter values, where we will generate data from the model using the parameter values in the original paper and look for Eureqa being able to discover equations that can approximate the dynamics of the system. Following this we will generate data from the model with varied parameters over several emigrations and look for Eureqa discovering Pratt in its general form. The second part of this section will cover the same experimentation, but applied to the Planqué model. The implementation details for this part of the project can be found in *Appendices* C.1 and C.3. Before discussing the results, we outline the methodology for the experiments with both fixed and variable parameters.

7.2.2.1 Fixed Parameter Methodology

In order to demonstrate being able to rediscover the Pratt and Planqué models with fixed parameters, they were written with the parameter values as specified in the original papers (found in *Sections* 5.2.1 and 5.2.2) and the models were built and run. Following this, the data generated was fed to the Eureqa system. The system was given all of the building blocks known to be in the solution, and the form of the solution was specified over all variables in the model. We demonstrate with an example from Pratt.

Within the Pratt model we have term:

$$\frac{dS}{dt} = -2\mu S - [\lambda I(R_1, S) + \lambda I(R_2, S)] \tag{7.2.2}$$

Where:

$$I(R_i, S) = \begin{cases} R_i & \textit{if } R_i < T \textit{ and } S > 0 \\ 0 & \textit{otherwise} \end{cases} \tag{7.2.3}$$

To discover the solution for $\frac{dS}{dT}$ we would specify solution form $\frac{dS}{dT} = f(S, A_1, A_2, R_1, R_2, P_0, P_1, P_2)$ with building blocks of constants, variables, addition, subtraction, multiplication, division, indicator functions and comparators (less than, less than or equal to, *etc.*).

Eureqa was left to run for an hour per equation and all solutions were chosen from the Pareto front according to the minimum error between the underlying function and the discovered solution. On finding a solution, the discovered differential equations were built and run as a model with the same initial conditions as the original model, and the results were compared.

7.2.2.2 General Form Methodology

The same tactics were then employed in deriving the differential equations for the general forms of the model. However, the parameters of the model were also provided as variables. Returning to the Pratt example, the solution for $\frac{dS}{dt}$ would be specified as:

$$\frac{dS}{dt} = f(S, A_1, A_2, R_1, R_2, P_0, P_1, P_2, \mu, \lambda, p_{12}, k_1, k_2, \phi, N, Q) \tag{7.2.4}$$

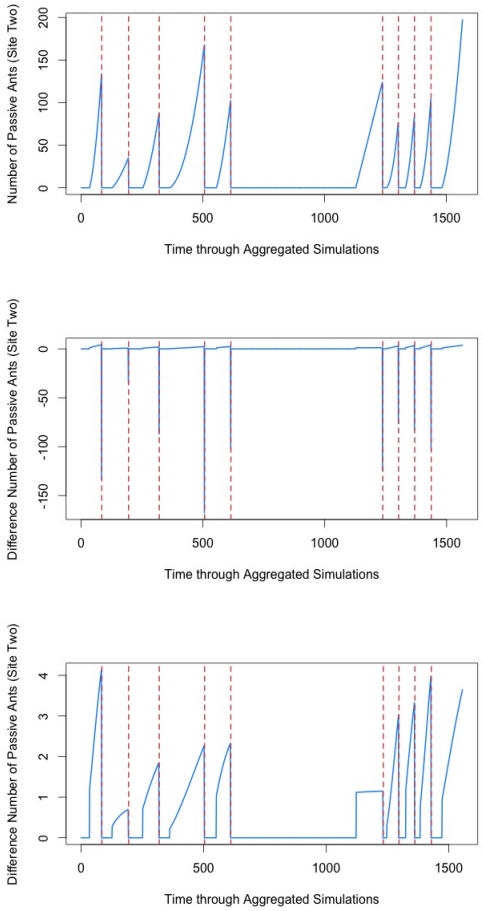
Where $\mu, \lambda, p_{12}, k_1, k_2, \phi, N, Q$ are the parameters as defined in the original paper.

Ten simulations were ran for each of the models, with randomly generated parameter values. The methods of generating these parameter values are given in the relevant section. Aggregating the simulations was then carried out by calculating the differences, and appending them back to back. Using the Pratt example, data was fed to the algorithm in the form:

$$(dS, dA_1, dA_2, dR_1, dR_2, dP_1, dP_2, S, A_1, A_2, R_1, R_2, P_0, P_1, P_2, \mu, \lambda, p_{12}, k_1, k_2, \phi, N, Q) \tag{7.2.5}$$

This method of aggregation offers a number of advantages. First of all, it fully captures each of the emigrations with different parameters in their entirety. As there is no averaging within the aggregation we lose no information.

Secondly, by calculating the differences and asking Eureqa to model them directly, rather than setting the Eureqa system to calculate the differential from the provided data, we increase the accuracy of the system. When we append the simulations, we create join points. As the number of ants per role tends to either increase or decrease significantly between the initial and final conditions, the value of the number of ants in a role changes sharply at the join points. When the Eureqa system calculates the differential of a variable, these points confuse the algorithm, and solutions become harder to find. By calculating the differences beforehand we can increase the accuracy of discovered solutions. This is demonstrated in *Figure* 20.



Top: Plot of the number of passive ants in site two over ten simulations appended back to back to be fed to the Eureqa system. Join points between simulations are shown in red.

Middle: Plot of the differential as seen by the Eureqa system when calculated directly from the data. We see the number of passive ants in the second site returns to zero at the beginning of each emigration, which in turn sharply affects the differential. When the Eureqa system is left to calculate the differential directly from the data, the join points confuse the algorithm.

Bottom: Plot of the differences calculated directly from the data and appended back to back. By calculating the differences for each simulation directly from data, appending them and looking for Eureqa to model this, we remove the confusion associated with the join points, and simplify the modelling problem.

Figure 20

This tactic works as all of the variables explicitly change at the join points. For example if a simulation ends at point i in *vector* 7.2.5, then at $i+1$ the values of all of the variables in *vector* 7.2.5 will alter accordingly: $\mu, \lambda, p_{12}, k_1, k_2, \phi, N$ and Q will change to the values for the following simulation, $S, A_1, A_2, R_1, R_2, P_0, P_1$ and P_2 will revert to the initial conditions and t will return to zero.

Each equation was then left to run for either one hour, or until the solution converged on the Eureqa system, based on which of the two options happened latest. Once the solutions were found, they were built and run as a complete model for each of the original sets of parameter values and initial conditions and compared to the underlying data. The results were then averaged for all simulations, and it is those results that are contained within this section.

7.2.2.3 Rediscovering the Pratt Model Results (Fixed Parameters)

The parameters for all of the simulations were fixed at the values in *Table* 2, *Section* 5.2.1 and two different quorum levels were investigated. First a quorum of one, and then a quorum of ten. A threshold of one was used in order to simplify the Pratt equations as the I and J functions (*Equations* 5.2.4 and 5.2.6) essentially revert to 0 and R_i respectively. Ten was then chosen to represent a full emigration without the simplification of the indicator terms.

In all cases, parts of the simulation in which the quorum becomes split at the end of an emigration and the ants are passively carried between sites are ignored, as they are separate from the underlying differential model (refer to *Section* 5.2.1 for details).

In both cases Eureqa discovered a system of equations capable of displaying the same dynamics as the original system. When the model derived by Eureqa was built and run the measurements of fit and error between the Eureqa derived system and the underlying data were well within a tolerable range for both quorums.

Term	MAE	MSE	R^2
$\frac{dS}{dT}$	0.01086604	0.001313848	0.9999623
$\frac{dA_1}{dT}$	0.02487834	0.002800469	0.9995676
$\frac{dA_2}{dT}$	0.06071242	0.018211421	0.9985154
$\frac{dR_1}{dT}$	0.04514428	0.004506528	0.9993875
$\frac{dR_2}{dT}$	0.84162938	0.766014120	0.9950044
$\frac{dP_1}{dT}$	0.51755457	0.319845302	0.9970333
$\frac{dP_2}{dT}$	0.81604023	1.136618195	0.9982515

Term	MAE	MSE	R^2
$\frac{dS}{dT}$	0.04487266	0.027818548	0.9990895
$\frac{dA_1}{dT}$	0.05644447	0.030815908	0.9965134
$\frac{dA_2}{dT}$	0.05527985	0.005572916	0.9996049
$\frac{dR_1}{dT}$	0.10386072	0.031004419	0.9966537
$\frac{dR_2}{dT}$	0.59128650	0.396517542	0.9972667
$\frac{dP_2}{dT}$	1.47210703	2.312234543	0.9985841

Table 6: On the left we have measurements of error and fit for the Pratt model with quorum one. The values on the right are for quorum ten. Note, $\frac{dP_1}{dt}$ has been missed from the quorum ten emigration as the number of passive ants at the inferior site remains at zero throughout the emigration. This is characteristic of higher quorum emigrations where accuracy is high.

All R^2 values are suitably close to one, with the MAE and MSE values either close to, or below one. This gives us that the fit of the model is close in terms of its shape and error. We reinforce this impression with *Figure* 21, where we plot the model as given to Eureqa, and the Eureqa derived model for the quorum one case.

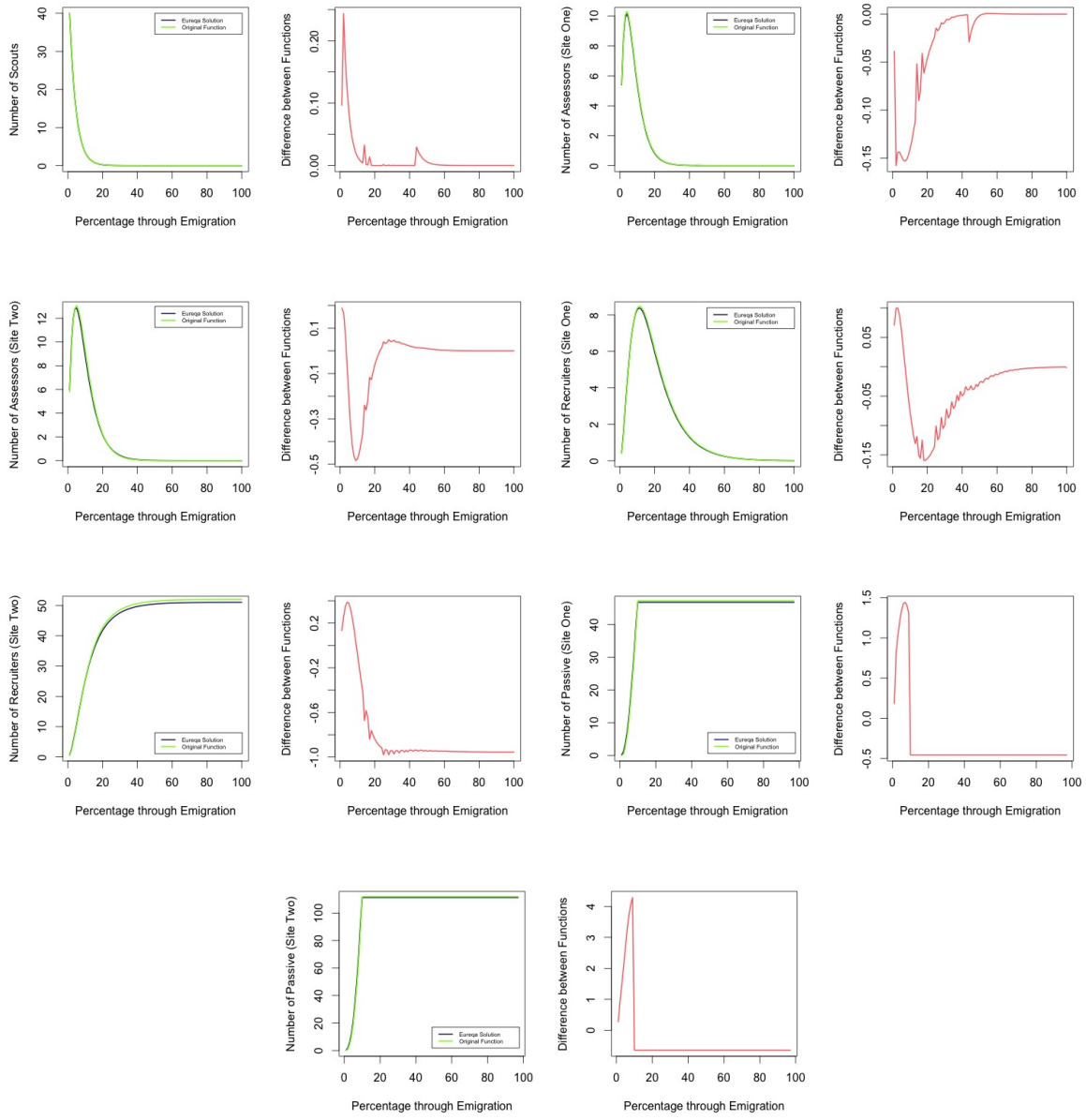


Figure 21: The seven equations determining the number of ants per role during a simulation, as dictated by the Pratt model with fixed parameters and a quorum of one. Green lines represent the underlying function, whereas dark blue lines are used to represent the model derived using Eureqa. Plots are also included charting the difference between between the two and are represented using red lines (*note:* the data plots and difference plots are scaled differently).

7.2.2.4 Rediscovering the Pratt Model Results (General Form)

The Pratt model was built and run as specified in *Section* 7.2.2.2 on the variable parameter methodology. Random parameter values were generated with a normal distribution with mean and standard deviation as specified in *Table* 2, *Section* 5.2.1. The data was passed to Eureqa and the solution system of differential equations was built and run, then compared to the original models. The average results per role and their standard deviations are contained in *Table* 7.

Term	MAE	SD	MSE	SD	R^2	SD
$\frac{dS}{dT}$	0.1139649	0.1442393	0.07035742	0.1729044	0.9997179	0.0004851093
$\frac{dA_1}{dT}$	0.2832943	0.3088348	0.19681104	0.3219927	0.9772385	0.0150865707
$\frac{dA_2}{dT}$	0.3406168	0.2938838	0.22209563	0.2405429	0.9905018	0.0076276350
$\frac{dR_1}{dT}$	0.3086328	0.2176744	0.19890819	0.2098117	0.9711313	0.0370525820
$\frac{dR_2}{dT}$	0.2644154	0.1468475	0.13575935	0.1340842	0.9930836	0.0144480116
$\frac{dP_1}{dT}$	1.0982261	0.7570850	4.28354531	7.0528300	0.9830426	0.0363834952
$\frac{dP_2}{dT}$	1.3749406	0.6791810	3.87799175	2.6717954	0.9961079	0.0033629878

Table 7: Averaged results for equation discovery applied to the Pratt model with varying parameter values. SD refers to the standard deviation of the average values over all simulations. The standard deviation for the MSE of P_1 may appear high, but we note that this is the standard deviation of the squared error. Therefore small differences between the underlying and approximating models will grow large. The number of passive ants in an emigration is also a lot greater than the number of active ones, therefore we would expect the standard deviation to be greater for the passive roles.

Reflecting on these values, we see that over all simulations the error is low, and the fit is high, with minimal variation. This gives us confidence in Eureqa being able to derive sets of differential equations that accurately represent the dynamics of well-defined systems.

We also examine the form of the equations given by the Eureqa algorithm, and compare them side by side with the underlying Pratt equations. The underlying formulae, and Eureqa derived formulae are given in *Table* 8.

Generally, we see that the equations are, analytically, very close to the originals. Aside from minor deviations in coefficients, the results for $\frac{dS}{dt}$ are very similar. $\frac{dA_1}{dt}$ shares the μS and $-k_1 A_1$ terms, whilst the inner term offers a reasonable approximation to the underlying expression.

Moving to the $\frac{dA_2}{dt}$ expression, we see that the majority of the equation is captured. Initially, we have the matching μS and λR_2 dependencies. The λ^2 term in the Eureqa derived expression can be seen as negligible, as λ is constrained to be very small. The $\frac{dA_2}{dt}$ expression also shares the $p_{12}A_1$ and $A_2 k_2$ terms. Although the Eureqa derived function also contains a $p_{12}\mathbb{I}[p_{12} > P_2]$ expression, p_{12} is constrained to be small, whereas P_2 grows rapidly once quorum is reached, therefore its effect is limited.

The remaining expressions are then rearrangements of each other. In total this gives the impression that, when provided data representing the general form of the model, Eureqa is capable of discovering good approximations of the underlying functions.

We contextualise this in terms of the current state of the field of equation discovery. The most recent work in the field has shown that, given the underlying data for a known differential equation system, Eureqa can develop very reasonable approximations of the underlying dynamics [80]. This is the same point we have reached within our own experimentation. In a seperate application to the glysotic oscillation model, we have shown Eureqa to be capable of discovering an approximation of a system of differential equations, where underlying relationships are known to exist.

Original Function	Eureqa Derived Function
$\frac{dS}{dt} = -2\mu S - [\begin{cases}\lambda R_1 & \text{if } R_1 < T \text{ and } S > 0\\ 0 & \text{otherwise}\end{cases} + \begin{cases}\lambda R_2 & \text{if } R_2 < T \text{ and } S > 0\\ 0 & \text{otherwise}\end{cases}]$	$\frac{dS}{dt} = -2\mu S - [1.02\begin{cases}\lambda R1 & R_1 < T \text{ and } S > 0\\ 0 & \text{otherwise}\end{cases} + 1.06\begin{cases}\lambda R_2 & R_2 < T \text{ and } S > 0\\ 0 & \text{otherwise}\end{cases}]$
$\frac{dA_1}{dt} = \mu S + \begin{cases}\lambda R_1 & \text{if } R_1 < T \text{ and } S > 0\\ 0 & \text{otherwise}\end{cases} - p_{12}A_1 - k_1 A_1$	$\frac{dA_1}{dt} = \mu S - A_1\begin{cases}p_{12} & Q + k_1 - A_1 k_1 \leq R1\\ p_{12} - S\mu\mathbb{I}[P_1 > 0] - 0.62(k_1\lambda t)\mathbb{I}[S > 0] & Q + k_1 - A_1 k_1 > R1\end{cases} - k_1 A_1$
$\frac{dA_2}{dt} = \mu S + \begin{cases}\lambda R_2 & \text{if } R_2 < T \text{ and } S > 0\\ 0 & \text{otherwise}\end{cases} + p_{12}A_1 - k_2 A_2$	$\frac{dA_2}{dt} = \mu S + \begin{cases}1.04\lambda R_2 + \lambda^2 & [P_2 < 0.001S]\\ 0 & \text{otherwise}\end{cases} + p_{12}A_1 - 1.003A_2 k_2 + p_{12}\mathbb{I}[p_{12} > P_2]$
$\frac{dR_1}{dt} = k_1 A_1 - p_{12}R_1$	$\frac{dR_1}{dt} = k_1 A_1 - p_{12}R_1$
$\frac{dR_2}{dt} = k_2 A_2 + p_{12}R_1$	$\frac{dR_2}{dt} = k_2 A_2 + p_{12}R_1$
$\frac{dP_1}{dt} = \phi\begin{cases}0 & \text{if } R_1 < T \text{ or } P_0 = 0\\ R_1 & \text{otherwise}\end{cases}$	$\frac{dP_1}{dt} = \phi\begin{cases}R_1 & \text{if } R_1 \geq T \text{ or } P_0 > 0\\ 0 & \text{otherwise}\end{cases}$
$\frac{dP_2}{dt} = \phi\begin{cases}0 & \text{if } R_2 < T \text{ or } P_0 = 0\\ R_2 & \text{otherwise}\end{cases}$	$\frac{dP_2}{dt} = \phi\begin{cases}R_2 & \text{if } R_2 \geq T \text{ or } P_0 > 0\\ 0 & \text{otherwise}\end{cases}$

Table 8: Table of resulting equations derived from Eureqa using the Pratt model.

7.2.2.5 Rediscovering the Planqué Model Results (Fixed Parameters)

We now extend the application of the equation discovery to the Planqué model from *Section* 5.2.2. The results are very similar to the Pratt case, and so we do not cover them in as much detail. As with Pratt, the Planqué model was ran with fixed parameters (specified in *Table* 3) and data was generated for emigrations with both quorums one and ten. A quorum of one was chosen as it simplifies the $l(\lambda, R, Q, A)$ function (*Equation* 5.2.12) to zero, and the $r(\lambda, R, Q, A)$ and $c(\phi, R, Q, P)$ functions (*Equations* 5.2.13 and 5.2.14) to $\phi\frac{RP}{R+P}$ and $\lambda\frac{RA}{R+A}$ respectively. Quorum ten then represents a more realistic emigration. The data was given to Eureqa to determine if it could find equations approximating the dynamics of the system, and the results are contained below.

Term	MAE	MSE	R^2
$\frac{dS}{dT}$	0.12629126	0.017852837	0.9994426
$\frac{dA}{dT}$	0.03042366	0.003670258	0.9999044
$\frac{dR}{dT}$	0.11669034	0.021057456	0.9993843
$\frac{dP}{dT}$	0.74929501	1.030477376	0.9998354
$\frac{dC}{dT}$	1.79186518	4.286957257	0.9993054

Term	MAE	MSE	R^2
$\frac{dS}{dT}$	0.02538995	0.0011332140	0.9999632
$\frac{dA}{dT}$	0.01107095	0.0008035884	0.9999704
$\frac{dR}{dT}$	0.15685622	0.0273998007	0.9992256
$\frac{dP}{dT}$	0.78730925	0.8059080483	0.9995930
$\frac{dC}{dT}$	0.45988509	2.5210640291	0.9998674

Table 9: On the left we have measurements of error and fit for the Planqué model with quorum one. To the right are the measurements of error and fit for the Planqué model with quorum ten.

7.2.2.6 Rediscovering the Planqué Model Results (General Form)

Tests were also run with the Planqué model in order to attempt discovering it in its general form. These tests were done following the methodology in *Section* 7.2.2.2 and the results are found in *Table* 10. The parameter values for the general form were randomly generated using a uniform distribution with upper and lower bounds defined in appropriate ranges, the details of which can be found in *Appendix* B. Data for emigrations using these parameter values was then generated from the model and fed to Eureqa to see if it could discover Planqué in its general form.

Term	MAE	SD	MSE	SD	R^2	SD
$\frac{dS}{dT}$	0.52290488	0.4978433	0.76897439	1.5192723	0.9928690	0.0117055529
$\frac{dA}{dT}$	0.07923314	0.1335152	0.05192556	0.1024016	0.9998051	0.0003867595
$\frac{dR}{dT}$	0.31731283	0.2069660	0.17996313	0.1887314	0.9970360	0.0033663114
$\frac{dP}{dT}$	1.17220809	0.8493422	4.73547872	5.2305977	0.9989176	0.0010920121
$\frac{dC}{dT}$	1.90328074	1.5214514	11.19337939	11.8455340	0.9976884	0.0022099998

Table 10: Averaged results for equation discovery applied to the Planqué model with varying parameter values. SD refers to the standard deviation of the average values over all simulations. Although the standard deviation may appear high for the MSE of both P and C we note that the this is the standard deviation of the squared error, and so small differences between the underlying and approximated models will quickly become large as they are squared. We couple this with the fact that throughout an emigration of a colony of size N, with fraction of active ants F, both P and C will take values within the range $[0, (1-F)N]$ as opposed to the other roles, which will take values within a much smaller range. Therefore it is acceptable for the standard deviations and errors of these roles to be a lot higher.

The results match that of the Pratt experiments. Even in its general form, Eureqa is capable of discovering differential equations that accurately model the underlying dynamics of the system.

7.3 Discussion

We finish this section with a discussion of the results. We begin by revisiting the success of Eureqa when applied to singular equations, not forming a part of a system (*Section* 7.2.1). The ability of Eureqa to discover a variety of accurately approximating equations for a range of data with varying complexities supports all previous research in the area, where Eureqa has shown to be capable of achieving such goals [20, 55, 79].

We now move to the discussion of the results from the Pratt and Planqué model testing (*Section* 7.2.2). We consider two practical points related to the experimentation done with the Pratt and Planqué models. We then conclude the discussion by contextualising the findings in light of the existing research in the field, and outline our next steps.

The first of the two practical points relates to the sensitivity of discovered systems to quorum levels. We recognize the number of ants in a nest exceeding the given quorum threshold as being a crucial point in altering the dynamics of the systems. Once quorum has been reached, carrying begins and the number of passive ants at the site starts to increase. The point at which this occurs in both models is related to an indicator function, where when the number of recruiters exceeds the quorum threshold, the dynamics of the system alters.

If the approximation of the expression for the number of recruiters has a small error, and in the underlying data this point is only just reached, then the error can mean that the quorum level for the emigration in the approximated system is never exceeded. This means the dynamics of the system never switches between states. This is an example of how small-scale errors, when applied to systems with a high sensitivity to certain terms, can cause drastically different behaviours. In *Section* 8 we will see that recognising such sensitivities can be crucial in understanding the performance of approximating systems of differential equations.

The second of the two practical points to be discussed is the Eureqa approach to expression simplicity. In *Section* 6.1.2 we discussed the definition of parsimony, and how Eureqa employs an accuracy-parsimony Pareto front to narrow down the number of solutions it presents. However, the Eureqa system includes a failing in that it has no consideration for initial conditions of differential equations. We demonstrate this with an example from the Pratt model.

The expression that Eureqa finds for the $\frac{dP2}{dt}$ term in the Pratt model with quorum ten is:

$$\frac{dP2}{dt} = 0.09952 \times R_2 \times \mathbb{I}[P_2 > 0] \times \mathbb{I}[P_0 > 0] \tag{7.3.1}$$

Which directly mirrors the true underlying expression, where the number of passive ants at site two increases at a rate proportional to R_2 if there are ants at the old nest left to carry, and carrying has begun. However, the point at which carrying has begun is simulated by the $\mathbb{I}[P_2 > 0]$ term as at this point the number of passive ants at the second site will be greater than zero. This correctly simulates the emigration when we have access to the training data, but when we have initial condition $P_2 = 0$ and we build the model from the initial conditions, then we never leave the $P_2 = 0$ state.

This underlines one of the points we must be aware of going forward with the Eureqa system. To the Eureqa algorithm this represents an optimal solution as it has both low error and low complexity. The system has no awareness of the types of problems dependent on initial conditions. However, they are easy to solve. By seeding the solution with the time point in the training data when P_2 exceeds zero, we can simulate the start of carrying at the correct juncture, and the approximation becomes accurate again.

We now summarise the findings and set the scene for the following section. In *Section* 7.2.2 we saw that in both models, Eureqa was capable of discovering systems of differential equations representing the underlying dynamics of the supplied data when the parameters of the models were fixed. Again, in both cases, this extended to finding the general form of the model given the required information. In the field of equation discovery, we have reached the cusp of modern research in the context of the project. In the following sections we look to extend the application away from these well-defined systems, and into more complex data.

8 Discovering Differential Equation-Based Summaries of SPACE and AH-HA

In this section we move from discovering systems of differential equations from data generated by well-defined differentials, to the application of the same techniques but to more complex data. *Section* 7.2.2 was used to show that, when there exists well-defined underlying functions, Eureqa can discover accurate approximations of the original system. We now extend this work to see if it is possible to find approximations of systems where data is no longer generated using simple underlying functions, but instead contains all of the complexity and variation of more realistic situations.

In *Part* I we discussed one of the existing challenges facing the fields of computer science and biology: unifying existing models of ant population dynamics by deriving differential equation-based summaries of agent-based models. It is through this aim that we explore the application of equation discovery to discovering systems of differential equations from complex data.

This part of the project is split into two sections. *Section* 8.1 covers the preparation of the SPACE and AH-HA data for use with Eureqa. *Section* 8.2 then covers the process of attempting to derive systems of differential equations from the SPACE and AH-HA data, and the weaknesses unveiled in the Eureqa system during these experiments.

8.1 Preparing the SPACE and AH-HA Data

This section is divided into two parts. The first deals with the form of the data provided to the Eureqa system, and the definition of the solutions we hope to derive. The second covers the process of simplifying down the SPACE and AH-HA data into their approximate general behaviours in order to try to define differential equations representing this behaviour. The implementation details relating to the SPACE and AH-HA models can be found in *Appendices* C.4 and C.5.

8.1.1 Deciding the Data and Solutions Form

As discussed in *Section* 6.1.2, the form of the data provided to the Eureqa algorithm is crucial to the type of solutions that will be derived by the system. Therefore, we specify the form of the solution we would like to derive, and then define the form of the data for the algorithm around this.

To specify the solution form, we consider one of the motivations for the project. The purpose of deriving differential equation-based summaries from existent agent-based models is to standardise the models and ease their comparison. Therefore, we look to derive differential equation-based summaries of an analytical form similar to the existent models. The scenario on which the majority of models are based is the case of two equidistant nests of differing quality [49,71,78]. One of these models, Pratt, is already in differential equation form, and therefore to ease the comparison of models we define the desired form of the differential equation-based summaries identically. This gives us solutions of the type:

$$\frac{dS}{dt} = f(S, A_1, A_2, R_1, R_2, P_0, P_1, P_2, t, N, Q) \tag{8.1.1}$$

$$\frac{dA_1}{dt} = g(S, A_1, A_2, R_1, R_2, P_0, P_1, P_2, t, N, Q) \tag{8.1.2}$$

$$\frac{dA_2}{dt} = h(S, A_1, A_2, R_1, R_2, P_0, P_1, P_2, t, N, Q) \tag{8.1.3}$$

$$\frac{dR_1}{dt} = k(S, A_1, A_2, R_1, R_2, P_0, P_1, P_2, t, N, Q) \tag{8.1.4}$$

$$\frac{dR_2}{dt} = l(S, A_1, A_2, R_1, R_2, P_0, P_1, P_2, t, N, Q) \tag{8.1.5}$$

$$\frac{dP_1}{dt} = m(S, A_1, A_2, R_1, R_2, P_0, P_1, P_2, t, N, Q) \tag{8.1.6}$$

$$\frac{dP_2}{dt} = n(S, A_1, A_2, R_1, R_2, P_0, P_1, P_2, t, N, Q) \tag{8.1.7}$$

Where $S, A_1, A_2, R_1, R_2, P_0, P_1$ and P_2 are the number of ants scouting, assessing site one, assessing site two, recruiting to site one, recruiting to site two, passive at the original site, passive at site one and passive at site two respectively. The variables t, N and Q are time, colony size and quorum threshold. We now extend the form of data shown to have worked with well-defined underlying systems and define the structure of the data for Eureqa as:

$$(S, P_0, P_1, P_2, A_1, A_2, R_1, R_2, dS, dP_0, dP_1, dP_2, dA_1, dA_2, dR_1, dR_2, t, N, Q) \tag{8.1.8}$$

From the structure of the solutions and the data form required to derive these solutions we move to the definition of the general behaviours of the models.

8.1.2 Defining General Behaviours of the SPACE and AH-HA Models.

The implementation details of this section are contained in *Appendix* C.6. Having defined the type of solution we are looking to generate, and the form of the data we will use to generate it, we aim to define the general behaviours that we would like to see exhibited in a system of differential equations. Capturing the behaviour of each emigration in its entirety in a model is overly ambitious due to the complexities of the data associated with each role, and the effect of the probabilistic components of both AH-HA and SPACE. The probabilistic elements of the models are problematic as they mean that emigrations with the same conditions can have vastly different behaviours. This in turn means that systems of differential equations will have to capture large ranges of behaviours with only small differences in parameters and inputs. Therefore we require a process to remove these data complexities and probabilistic effects. In order to do this we implement a number of techniques from the field of nonparametric regression.

Nonparametric regression is an area of mathematics used, amongst other applications, to smooth data and reduce noise in signals. In the context of our project, the signal will be taken to be the consistent behaviours generated by our models given different emigration conditions, and noise will be taken as the variance between emigrations given the same emigration conditions. By applying nonparametric smoothing methods we isolate the key trends in the emigrations, and reduce the effect of the probabilistic elements of both models on the results.

A variety of nonparametric smoothing techniques were considered for this purpose including: the kernel method using the Nadaraya-Watson formula [58,91], k nearest neighbours ($k-N-N$) [11,21], isotonic regression [4] and spline smoothing [6]. Primarily we reject isotonic regression, as it relies on the assumption that our data is isotonic, which for the most part it is not. Both Nadaraya-Watson and $k-N-N$ were seriously considered, but did not offer one of the main assets of spline smoothing, and so were discarded for the final method.

The advantage offered by spline smoothing is inherent in the approach to smoothing it takes. The technique functions by partitioning the dataset into intervals at certain join points and choosing a spline of order p to be fitted to the data (where we define a spline to be a piecewise polynomial of order p, which is connected such that it is $p-1$ times differentiable). Following this, the intervals are connected to create a smooth curve.

In order to choose where to partition the data and the degree of the polynomial, p, a measurement of fidelity to the data is composed, which measures the closeness of the data to the model and the local variation. It can be shown that the solutions to the minimisation of the measurement of fidelity are always continuous piecewise cubic polynomials with continuous first and second derivatives and knots (join points) located on each of the x_i values (where we assume data is in form (x_i, y_i)). In terms of this project, this fits our intentions well. In our smoothing method we, for the most part, prioritise the method being impervious to outliers and generating smooth curves over being able to capture complex behaviours. By approximating data with low order polynomials spline smoothing does just this.

There does exist, however, one exception to the suitability of spline smoothing as a denoising technique. In the AH-HA model, the number of assessors rapidly reaches a peak, after which it begins to decrease steadily (demonstrated in *Figure* 22). The majority of denoising techniques tend to smooth over sharp peaks in data, losing their shape [6, 58]. Preserving these types of local extrema in signals is an acknowledged problem [85] and a number of methods have been devised that recognise certain types of outlying data as important to the shape of the underlying signal [16]. For our specific application, existing methods are unsuitable as they return a curve comprised of conjoined linear approximations or level sets. They are very efficient at reducing noise and identifying the underlying signal, however the result is not continuously smooth. This means that the underlying differential becomes difficult to model.

In order to assure smoothness whilst accurately capturing the underlying behaviour and the peak in the number of assessors, we must define our own, application specific, smoothing method. This comprises of separating out the curve into two parts. An initial part, which contains the first peak, and then a second part, which contains the steady decrease in the number of assessors.

This separation is achieved using the maximum point of the assessor curve. We specify a bandwidth around the maximum point consisting of a start point at time zero, and an end point at a time five time steps after the maximum is achieved. Five was chosen as in all cases this offered an accurate end point for the initial peak. The first part of the curve was then smoothed using spline smoothing with a high sensitivity to the data (to preserve the peak but induce a smoother curve), whilst a smooth with a lower sensitivity was employed on the end data (to remove the remaining noise).

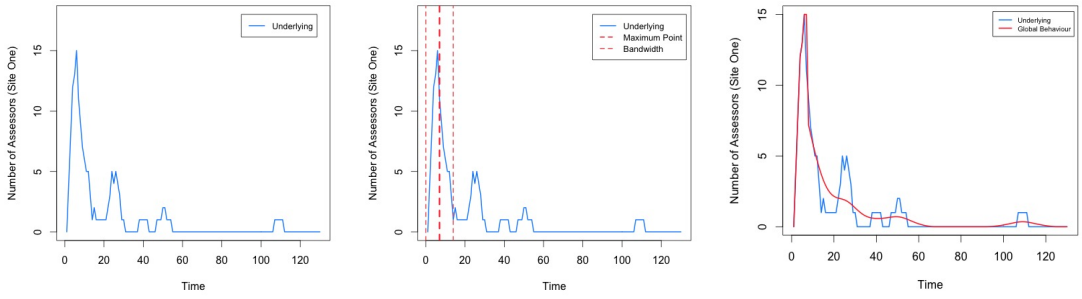


Figure 22: Figure to demonstrate defining the general behaviours of the AH-HA model for the assessors state. On the left is the raw data, we look to produce a smooth curve representing the underlying trend. The central plot represents the method of generating the bandwidth around the maximum point in order to separate the signal into two parts. The plot on the right shows the original curve, with the smoothed curve superimposed.

The curves generated from this process were defined as the general behaviours we looked to derive from AH-HA and SPACE (*Figures* 23 and 24). The methodology used for discovering the Pratt and Planqué models was then implemented in order to discover systems of differential equations representative of these general behaviours for a single emigration. The results are contained in *Section* 8.2.

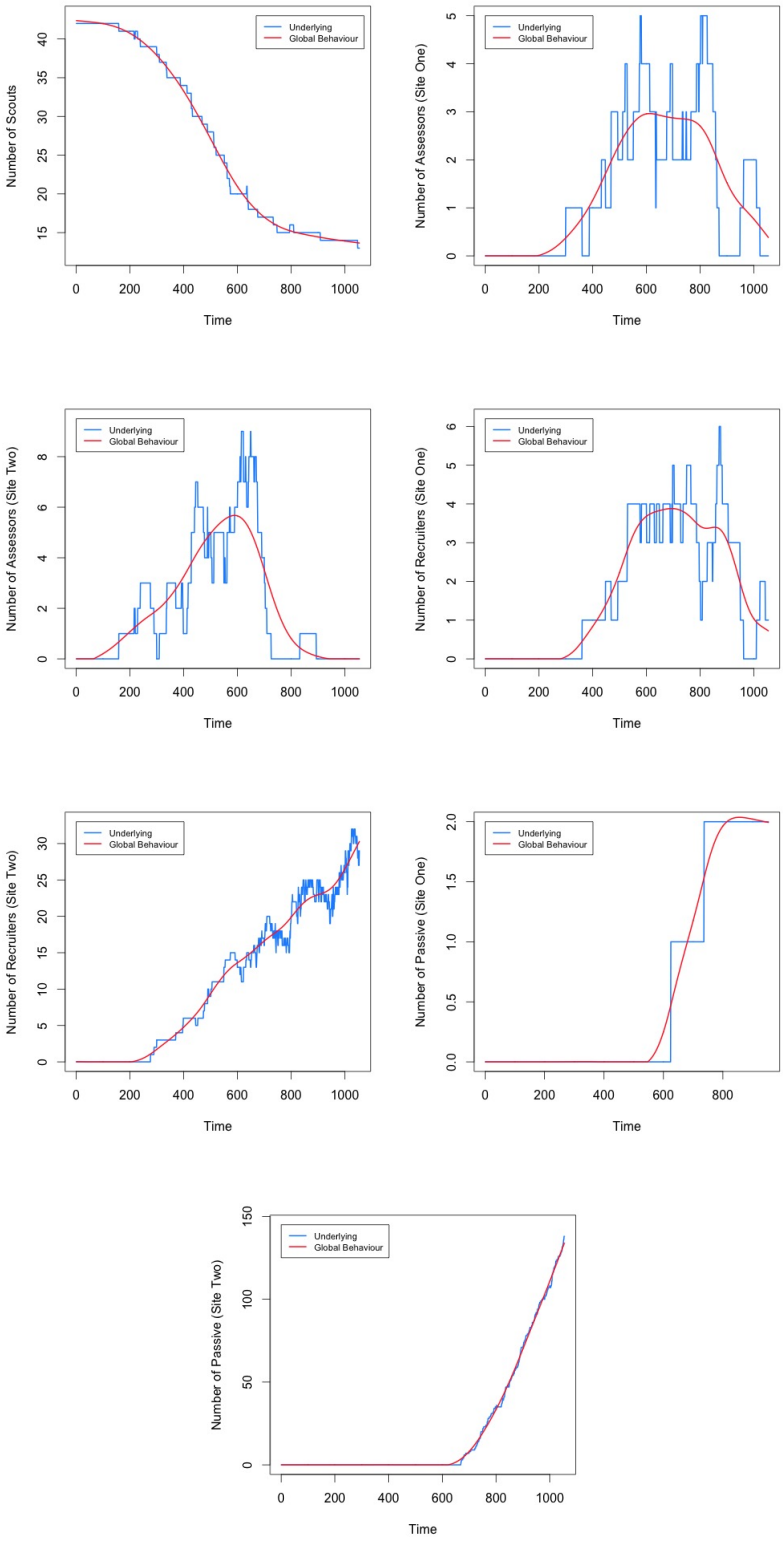


Figure 23: Demonstrating the definition of general behaviours for the SPACE model. Within this approach, we look to reduce noise and focus on the derived signal. For the simpler trends (S, R_2, P_1, P_2) we see that this is reasonably straight forward. There appears to be a higher degree of noise on the A_1, A_2 and R_1 curves. We define the simplest appropriate curves for these trends, eliminating the maximum amount of noise. In later sections we see that regardless of the decision to define the simplest possible curves, the same result would hold.

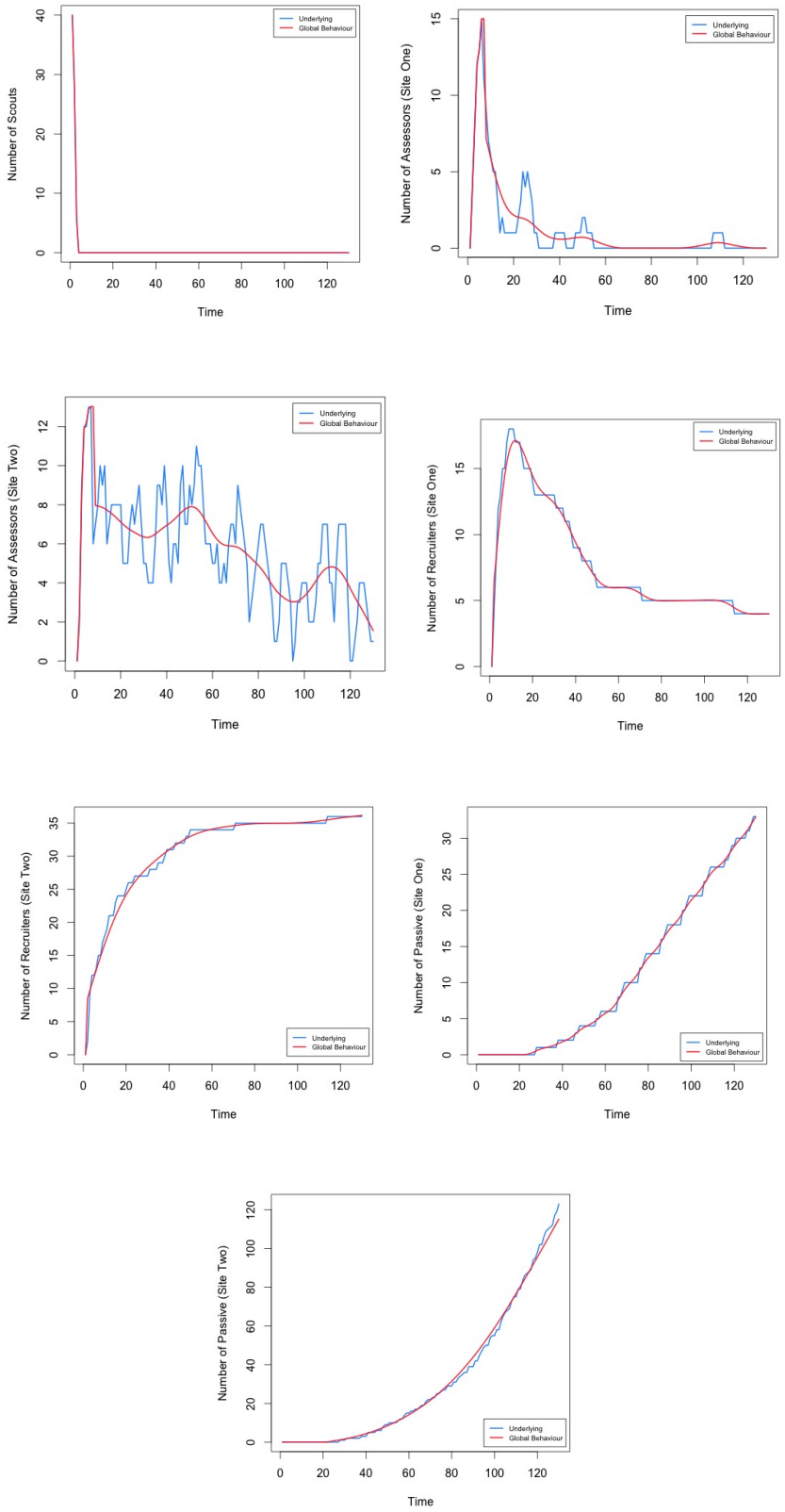


Figure 24: Demonstrating the definition of general behaviours for the AH-HA model. A similar compromise exists with the assessors, where we must try and decide what is significant and what is noise. The sensitivity of the smoothing was chosen to give the results as shown.

Aside from lessening the effect of the variation between emigrations with the same conditions, smoothing the curves also fulfils a secondary aim. The less smooth functions become, the more complex the differential becomes to model. By removing noise from the curves, we also reduce the complexity of the differential.

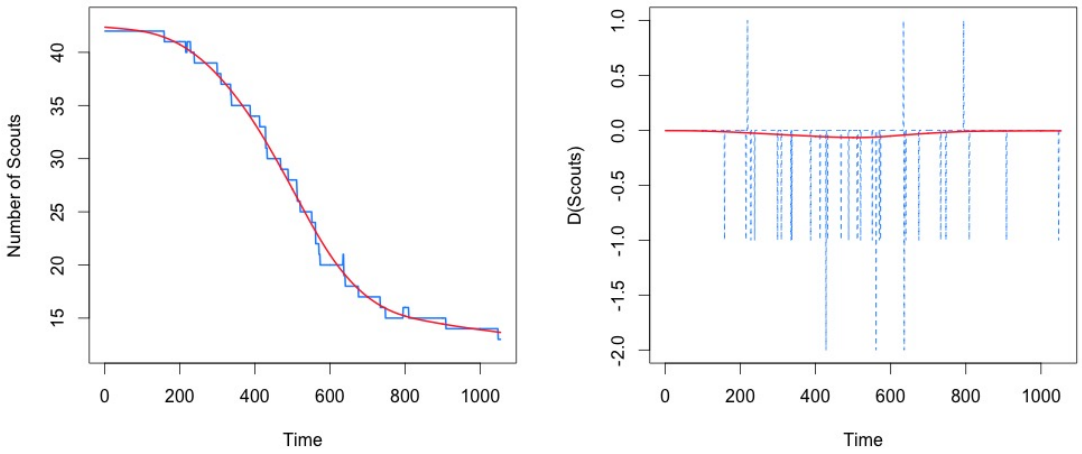


Figure 25: Demonstrating how smoothing the curves also reduces the complexity of the differential, even in the simplest cases. In both images the blue line represents the underlying data, whereas the red line is representative of the smooth curve. We see that, although the difference in shape on the left is minor, the difference in complexity of the differential equation we are looking to model (pictured on the right) is greatly reduced between smooth and non-smooth.

We have reached a definition of the general behaviours of the different roles in both the SPACE and AH-HA models, as well as the form of solutions we would like to derive and the necessary data form to derive such solutions. We now look to use these resources and employ the methodology from the previous sections in order to derive differential equation-based summaries of the agent-based models.

8.2 Weaknesses of the Nutonian Eureqa System Applied to Deriving Systems of Differential Equations from Complex Data

In *Section* 7.2.2 we devised a methodology capable of discovering systems of differential equations using data from systems with well-defined underlying equations. In this section we look to implement this methodology in order to attempt finding the differential equation-based summaries of both SPACE and AH-HA. In *Section* 8.1.2 we demonstrated a method for defining the general behaviours of the agent-based models. These act as the curves we look to approximate using differential equations. Implementation details for this section can be found in *Appendix* C.7.

8.2.1 Demonstrating the Weaknesses of Nutonian Eureqa

Within this section, we will focus on the most simple case: deriving the differential equation-based summary of SPACE for a single emigration. This is because any weaknesses that exists in Eureqa applied to systems of differential equations will affect both SPACE and AH-HA and extend over multiple emigrations. Hence it is only necessary to use one SPACE emigration to demonstrate the point.

Equation discovery was run on each of the differentials for one hour, or until Eureqa converged on a solution. The discovered equations were then combined and run as an entire model, and the results are given in *Figure* 26.

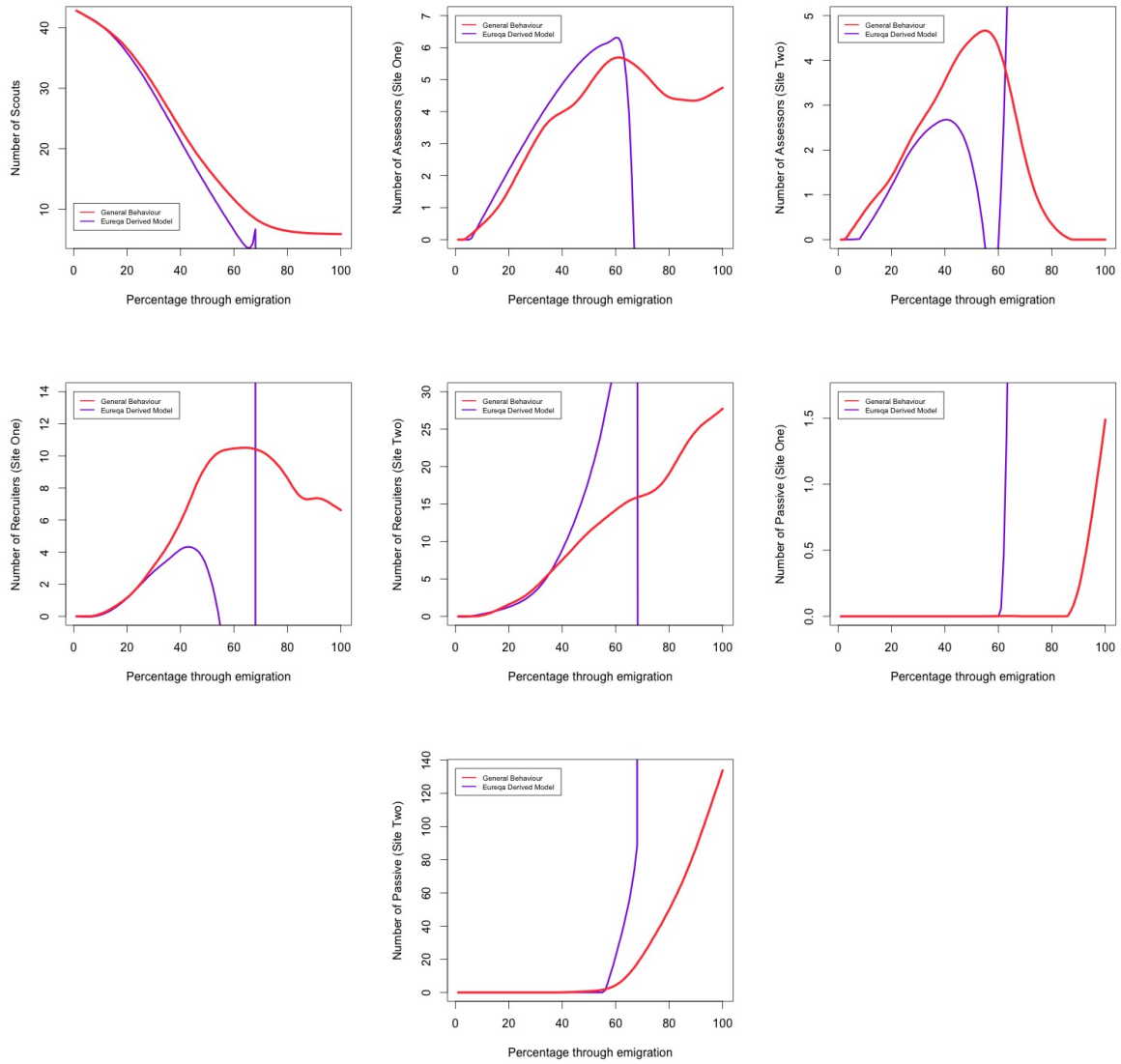


Figure 26: Results from attempting to discover the differential equation-based summaries of the SPACE model. The purple line represents the role according to the discovered differential equation system, whereas the red line is the underlying data.

From *Figure* 26 we see that discovering the differential equation-based summaries has been unsuccessful and the equations discovered by Eureqa poorly represent the underlying data. This seems unexpected for a number of reasons.

Initially, and exactly as in the Pratt and Planqué cases, each of the functions was reported by Eureqa to accurately model the training data. Therefore, we would expect a close fit between the data and the approximations, as was seen in the previous cases. In addition, we saw that this methodology worked well for both Pratt and Planqué. This combination of factors would lead us to believe that the same would occur for the SPACE and AH-HA models. However, it has not, and we now explore the reason why.

8.2.2 Rationalising the Weaknesses of Nutonian Eureqa

In the Pratt and Planqué cases, as in all of the previous applications of equation discovery, the techniques have been applied to well-defined sets of equations, or simple, real-life systems [80, 88]. This is the first attempt to apply these equation discovery techniques to more complex data. As we move away from rediscovering well-defined systems, and increase the complexity of the data we wish to model, we require more complicated functions to capture the curves in their entirety. As we increase the complexity of the model, we also move up in its sensitivity to error in the equations for different roles.

The sensitivity to error in the equations for different roles occurs as in order to create complex curves out of the limited number of variables available, the variables must be used with a greater frequency and in more complicated ways. Previously the Pratt and Planqué models contained a small number of terms, with mostly linear dependencies. However to capture the SPACE and AH-HA behaviours we require a larger number of more complex terms.

This means that small differences in the values of roles can have a large impact on the results we obtain. The overall effect is small errors begin to build up and propagate throughout the system of equations. We demonstrate this effect in *Figure* 27, where we purposefully introduce errors per time step and show the difference in the equation results for a given role.

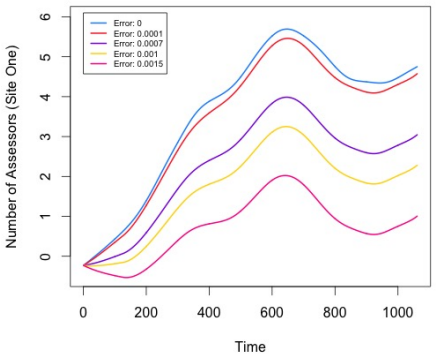


Figure 27: Demonstrating sensitivity in the differential equation system. A purposeful error per time step was introduced into each of the variables in one of the expressions in the model ($\frac{dA_1}{dt}$). We see that even with errors of a very small scale, the difference in the function is pronounced.

To demonstrate that the cause of the errors in the approximating model is the interaction of the derived equations, we plot the differential equation model in the case of each variable having perfect information on each of the others. This removes any error caused by the interaction between roles. We simulate this by allowing the differential functions access to the training data used to derive the model. This is demonstrated in *Figure* 28.

Within these plots, as there is perfect information, there is no error, and we see that the system behaves well and how we would expect it to, given the previous results and the claimed accuracy of Eureqa. When we remove this perfect information, variables begin to interact and realistic levels of error are allowed to build up and propagate through the system, spiralling out of control. In both *Figures* 26 and 27 we can see the difference in functions growing with time, demonstrative of how the errors propagate and grow large as the model is run.

We have shown that small errors in the values of variables cause large differences in the results from the equations (*Figure* 27). We have also shown that this is caused by the interaction of equations within the model (*Figure* 28). Finally we relate this to complexity. We compare the average complexities per equation over the different models we have attempted to derive, and show the correlation between complexity and fit (*Figure* 29). We define complexity as the inverse of parsimony as defined by Eureqa, therefore complexity is the number of terms in an equation for a role.

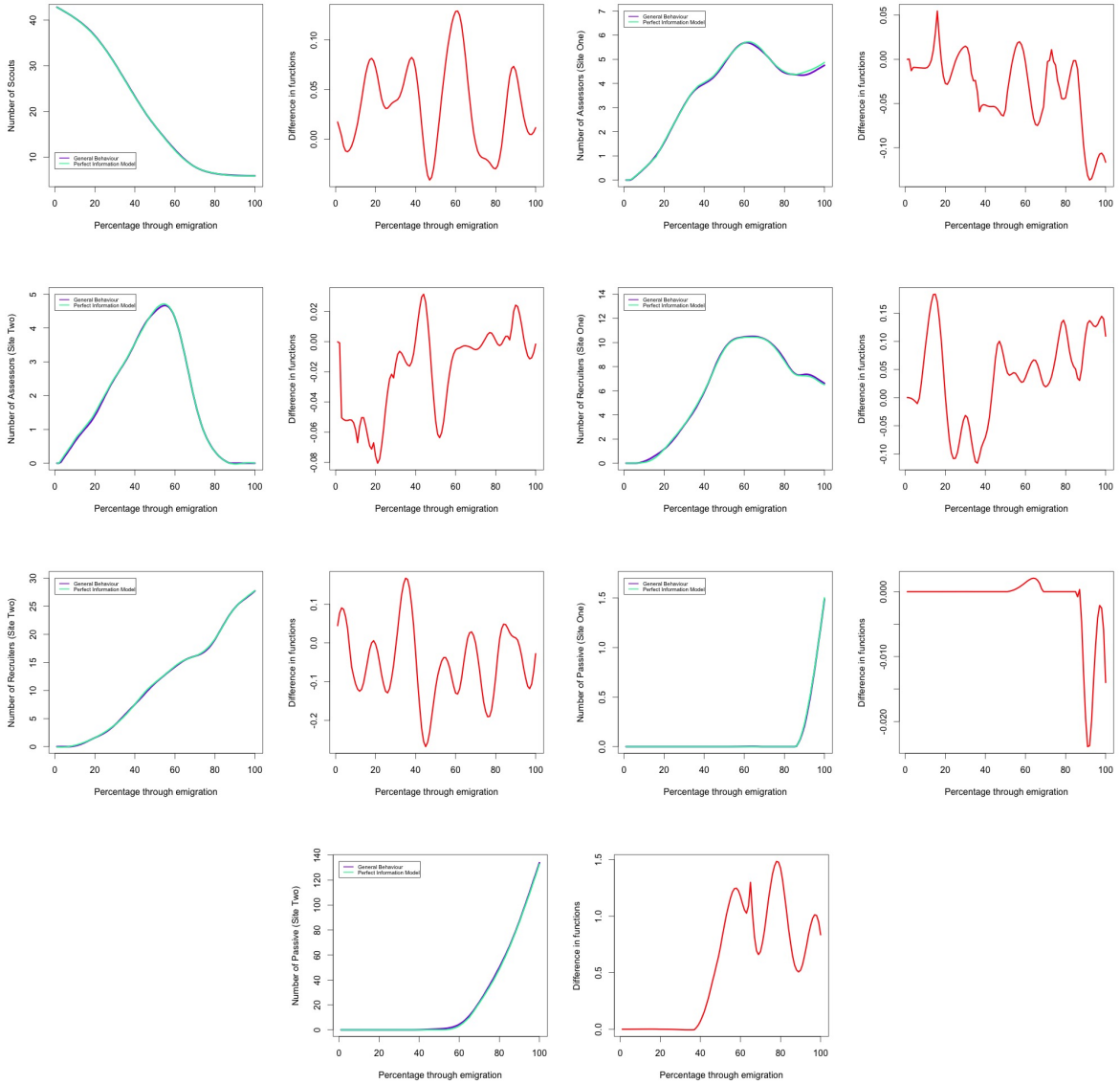


Figure 28: Results from attempting to discover the differential equation-based summaries of the SPACE model. The green line represents the role according to the discovered differential equation system, whereas the purple line is the underlying data. The plots in red represent the difference in functions (*note:* the data plots and difference plots are scaled differently). When the training data is used in building and running the model, the results become very accurate. This is as there are no errors in any of the variables used within the differential functions.

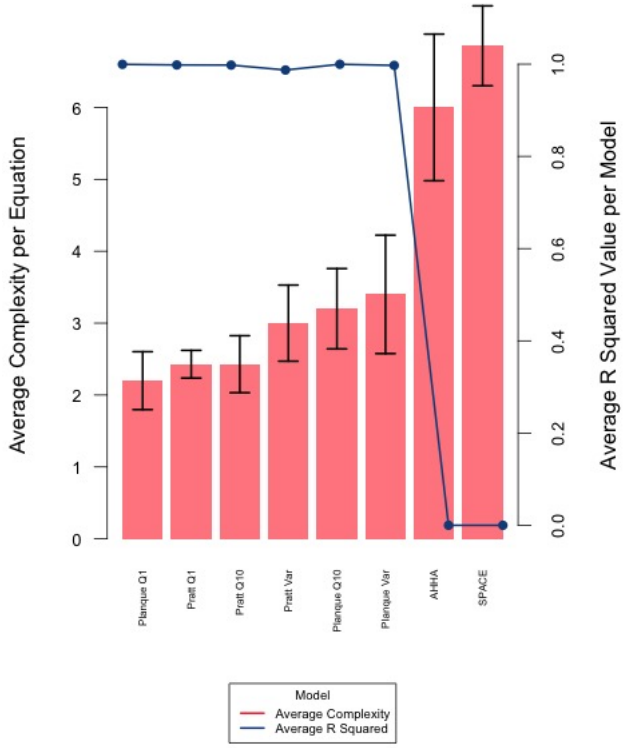


Figure 29: Plotting complexity against fit for the different model solutions derived using Eureqa. Error bars are used to represent 95% confidence intervals. We see that the fit for the Pratt and Planqué models is very high, whilst average complexity per equation is around half of that for the SPACE and AH-HA solutions. The *Pearson Product-Moment Correlation Coefficient (PPMCC)* was also taken between complexity and fit and has value -0.9636753 and $p - value$ of 0.0001166 showing a strong negative correlation (for implementation details see *Appendix* C.8, for $PPMCC$ definition see *Appendix* D.2.4).

We now identify the cause of this problem. This lies in the fact that the Eureqa system considers each of the differential equations singularly, and has no awareness of how they interact as a system. Specific to our context, Eureqa defines an equation for $\frac{dS}{dt}$, then $\frac{dA_1}{dt}$, then $\frac{dA_2}{dt}$, and so on for each variable. Individually, given the training data, the differential equations it derives are very accurate, and represent the underlying curves well. However, the Eureqa system has no awareness of the stability of the overall system. It does not consider how, when combined and allowed to interact, the differential equations will behave.

To expand further, the system contains no concept of, for example, the sensitivity of S to A_1. It uses the training data for A_1 and the other variables to define a model for S, which is accurate. It then contains no awareness of what will happen when we remove the A_1 training data, and instead used data generated from the approximating differential equation in generating S values, and the effect of any errors in this equation.

We conclude this section by summarising why the Eureqa system, when applied to complex data with no known underlying sets of equations, fails. As data becomes more complex, and the number of variables available to form expressions representing that data remains constant, the representative expressions must become more complex. As expressions become more complex, they also become more sensitive to error. As the Eureqa system cannot consider the model as a whole, only on a role by role basis, this means that it cannot form models that account for errors in other roles, and how they will affect the system of differential equations when all of the expressions are combined. Therefore it cannot find accurate systems of differential equations for complex data.

9 Experiments on Stabilising the SPACE Differential Equation-Based Summaries

As covered in *Section* 8.2.2, Eureqa fails to discover stable systems of differential equations as it does not consider the model as a whole and how the equations interact. As Eureqa only discovers a solution on an equation by equation basis this leads to small errors propagating throughout the system when equations are allowed to interact, and the differential equation model becoming unrealistic.

Therefore, we require a method of considering the model as a whole. From *Section* 7 we know that Eureqa is capable of finding accurate equations to model single quantities. From *Section* 8 we know that the reason that Eureqa fails is that it only considers the system of equations on an equation by equation basis. We now carry out three experiments in order to utilise the strengths of Eureqa, whilst addressing the weaknesses.

In each experiment we will use Eureqa to derive equations for the individual roles, and then implement other means in order to stabilise over the model as a whole. All of these experiments will be done for a single SPACE emigration, with colony size 200 and quorum size of 8. This is since these conditions represent a typical emigration, and any system that will stabilise over multiple emigrations will have to be able to stabilise over a single emigration. The choice of SPACE over AH-HA was made as finding a differential equation-based summary of SPACE is our top priority in order to add value to the wider SPACE project. However, any findings that apply to SPACE will extend to AH-HA.

A single emigration may appear a small population size to draw conclusions from. However, we make a number of arguments as to why this is an appropriate choice within this project. Initially we make a practical point. The process of running equation discovery algorithms is very slow (around one hour per equation, with seven equations per model). In addition, as the algorithm is run through an application, none of the work such as changing datasets or solution forms can be automated. This makes prototyping ideas and gathering large amounts of data within the given time frame very difficult. Additionally, we see that none of the solutions derived are exactly what we require, even for just a single emigration. The analysis of why this is so does not require multiple emigrations to be used. Hence we have restricted ourselves to analysing one emigration.

Initially, we look to stabilise the sets of equations by optimising parameters over the differential equation system (*Section* 9.1). Following this we experiment with stabilising by iteratively defining the model (*Section* 9.2). The final technique will use a brute force approach to trying to find a stable model (*Section* 9.3).

9.1 Stabilising Differential Equations through Parameter Optimisation

This marks the first experiment in stabilising the system of differential equations derived by Eureqa. Implementation details for this section can be found in *Appendix* C.9. As outlined in *Section* 8.2, the weakness with Eureqa applied to systems of differential equations lies in that it cannot consider the model as a whole, only as individual equations. By allowing Eureqa to define the proportionality of variables, and then optimising the parameters over the entire model, we look to include this global consideration and stabilise the results. We demonstrate this idea with a simplified example.

Consider the case when Eureqa derives the system of equations:

$$\frac{dA}{dt} = 2B \tag{9.1.1}$$

$$\frac{dB}{dt} = 3C \tag{9.1.2}$$

$$\frac{dC}{dt} = 4A \tag{9.1.3}$$

This model may be unstable due to the reasons outlined in *Section* 8.2. However, we know that, independently, the equations match the underlying data well. As the equations are accurate independently, it is reasonable to assume that the relationships between the variables derived are accurate. Therefore an intuitive approach is to take these relationships and use them as a basis for a model. Relating back to the example, we would use the set of equations:

$$\frac{dA}{dt} = \lambda B \tag{9.1.4}$$

$$\frac{dB}{dt} = \phi C \tag{9.1.5}$$

$$\frac{dC}{dt} = \eta A \tag{9.1.6}$$

Such that λ, ϕ and η are parameters to be optimised over. Algorithms we apply to optimise these parameters can then be chosen to minimise error over the entire system of differential equations, and so can consider the model as a whole, whilst still using the strengths of Eureqa to define the relationships between variables. Levenberg-Marquardt [42, 50, 54] was chosen as the optimisation algorithm, with absolute error as the minimisation metric. The choice of Levenberg-Marquardt was made as it is considered a standard technique for non-linear optimisation and data fitting, which has resulted in robust implementations of the algorithm. In addition, for medium sized problems it outperforms alternatives such as the gradient descent and conjugate gradient methods [77]. We implement this approach, and contain the results in *Figures* 30 and 31.

9.1.1 Results and Analysis

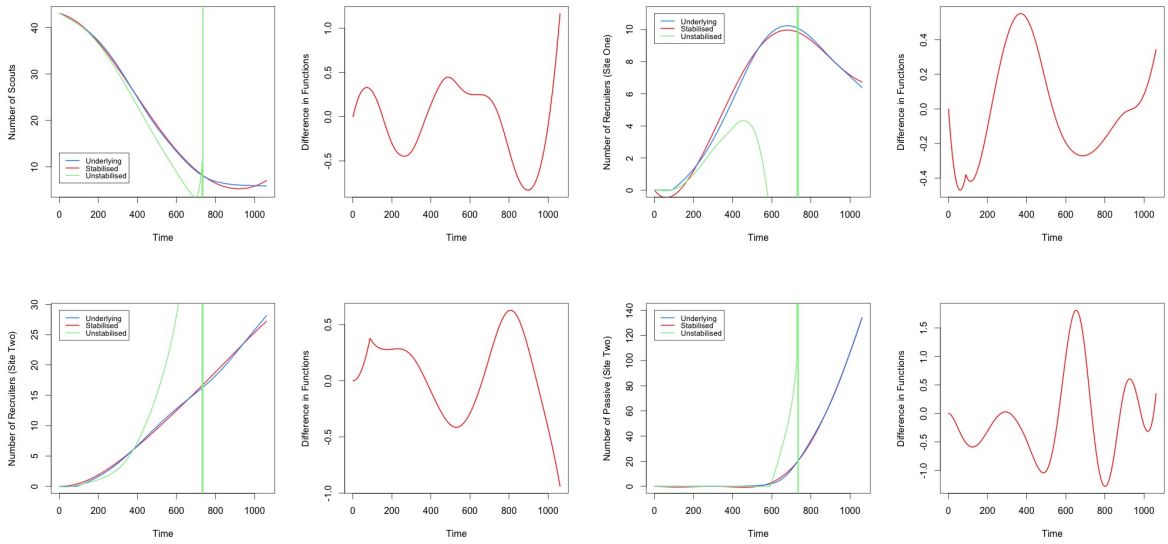


Figure 30: Results from attempting to stabilise the differential equation summaries of the SPACE model via parameter optimisation. Blue represents the derived general behaviours of the SPACE model, red the stabilised SPACE model, and green the unstabilised SPACE result. The plot on the right is then the difference between functions (*note:* the data plots and difference plots are scaled differently). For S, R_1, R_2 and P_2 the stabilisation greatly improves the model.

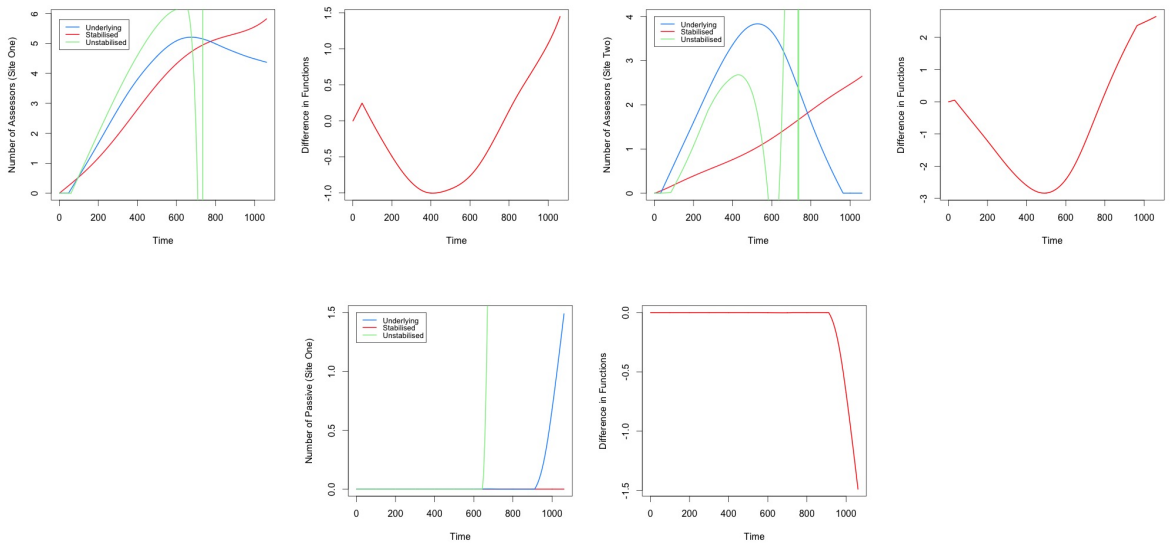


Figure 31: Results from attempting to stabilise the differential equation summaries of the SPACE model via parameter optimisation. Blue represents the derived general behaviours of the SPACE model, red the stabilised SPACE model, and green the unstabilised SPACE result. The plot on the right is then the difference between functions (*note:* the data plots and difference plots are scaled differently). For these remaining variables the equations become less volatile, but do not approximate the behaviours we hope to see.

In the majority of cases, this offers a great improvement on the original, unstabilised model. However, there are some scenarios where no improvement is made between the two. Ultimately, this is due to the limiting factor of the relationships between variables. If the relationships defined by Eureqa cannot offer a stable form to the model, then optimising over the parameter set cannot rectify this. Although it may offer an improvement in some cases, it cannot solve the problem entirely.

In addition, implementation becomes more complex if we choose to optimise over non-continuous functions. By choosing indicator functions as one of our building blocks, this leads to having to optimise for parameters in a non-continuous space. This space then needs to be broken into continuous intervals to apply Levenberg-Marquardt.

9.2 Stabilising Differential Equations Iteratively

The next tactic taken to stabilise the differential equation-based summaries of SPACE and AH-HA was to use an iteratively based method of defining the model. Implementation details for this section can be found in *Appendix* C.10. Certain quantities (the time, t, the colony size, N and the quorum, Q) are invariant to all other variables throughout an emigration. We know, without the use of differential equations, their values for each time step. This provides us a stable base for creating our model.

We now demonstrate how we would use and expand this stable base. If we define $\frac{dS}{dt}$ using only t, Q and N, then we know that errors in the other variables will not affect the result for S. Once we have defined a stable S, we can use the new S values to define an expression for A_1 in terms of S, t, Q and N, where this new expression will in turn be stable. We can then build up an expression for A_2 in terms of S, A_1, t, Q and N and iterate this for the remaining variables. To clarify we include the analytical form of the model when derived.

$$\frac{dS}{dt} = f(t, Q, N) \tag{9.2.1}$$

$$\frac{dA_1}{dt} = g(t, Q, N, S) \tag{9.2.2}$$

$$\frac{dA_2}{dt} = h(t, Q, N, S, A_1) \tag{9.2.3}$$

$$\frac{dR_1}{dt} = j(t, Q, N, S, A_1, A_2) \tag{9.2.4}$$

$$\frac{dR_2}{dt} = k(t, Q, N, S, A_1, A_2, R_1) \tag{9.2.5}$$

$$\frac{dP_1}{dt} = l(t, Q, N, S, A_1, A_2, R_1, R_2) \tag{9.2.6}$$

$$\frac{dP_2}{dt} = m(t, Q, N, S, A_1, A_2, R_1, R_2, P_1) \tag{9.2.7}$$

The order of the equations given above is the order of their derivation. Using this method it is possible to build up an accurate model, and the result is contained in *Figure* 33.

9.2.1 Results and Analysis

Despite the accuracy, this method is unsuitable for two reasons. Initially, it requires some order of precedence to the relationships between variables. By defining S in terms of t, Q and N we assume that there is no dependent relationship intrinsic to the system between S and any of the other variables. These assumptions then build up throughout the system, for example we assume that there is no dependent relationship between A_1 and any other variable except for S, A_2 and S and A_1, and so on.

The second reason relates to the flexibility of the method. When we define the function for S, we have three variables, only two of which change with emigration conditions (Q and N). Therefore any expressions for S extending over multiple emigrations must be able to capture each separate emigration in its entirety, solely through using these two values, which are fixed throughout emigrations. This does not allow enough freedom in expressions to fully capture emigrations, and so limits the extensibility of the method.

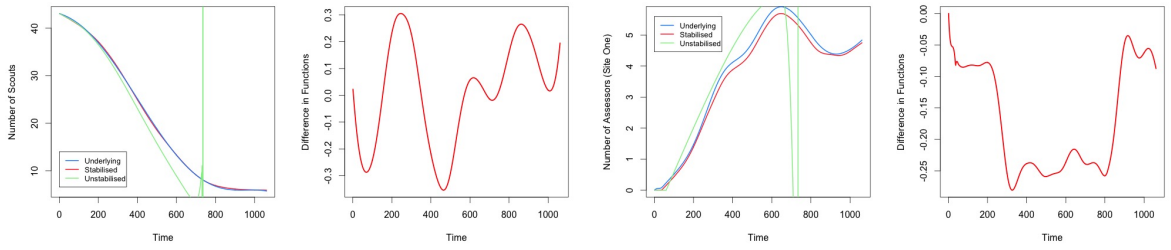


Figure 32: Results from attempting to stabilise the differential equation-based summaries of the SPACE model via iteration. Blue represents the derived general behaviours of the SPACE model, red the stabilised SPACE model, and green the unstabilised result. The plots to the right are then the differences between functions (*note:* the data plots and difference plots are scaled differently). We see that using this method, it is possible to define differential equation models which accurately represent the general behaviours we are interested in.

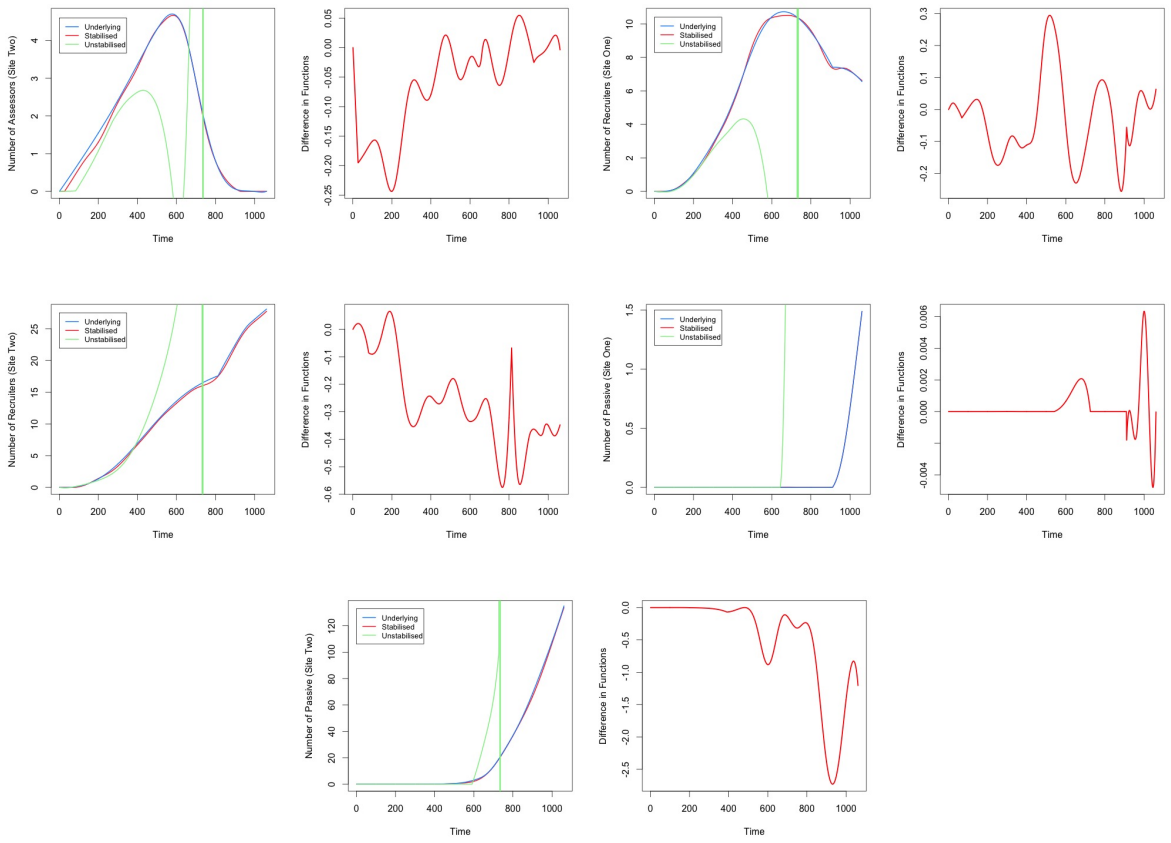


Figure 33: Results from attempting to stabilise the differential equation-based summaries of the SPACE model via iteration. Blue represents the derived general behaviours of the SPACE model, red the stabilised SPACE model, and green the unstabilised result. The plots to the right are then the differences between functions (*note:* the data plots and difference plots are scaled differently). We see that using this method, it is possible to define differential equation models which accurately represent the general behaviours we are interested in.

9.3 Stabilising Differential Equations by Brute Force

This is the final approach taken to address the weaknesses seen in Eureqa where the system does not consider the model as a whole, but only as individual equations. Implementation details for this section can be found in *Appendix* C.11. As covered in *Section* 6.1.2, Eureqa presents a number of possible solutions across an accuracy-parsimony Pareto front. Solutions along this front are then optimal in terms of their trade-off between error and complexity. Within this approach we attempt to calculate all possible models formed from the combinations of these solutions, gauging the most successful model from the combinations by averaging the R^2 goodness of fit over all roles.

Using this approach, if an accurate model exists from the combination of Eureqa presented equations, then it will be found. This technique helps combat one of the problems documented in *Section* 8.2.2. Within this section we saw that as equations become more complex, they become more sensitive to error, and errors begin building up and propagating throughout the system. Therefore there exists an argument that it is better to take simpler equations with a higher error, as the effect of inaccuracies in the other variables on that particular equation will be smaller. By building a system out of equations with simpler forms, but higher expected error, it may be possible to define an overall more accurate and stable system.

9.3.1 Results and Analysis

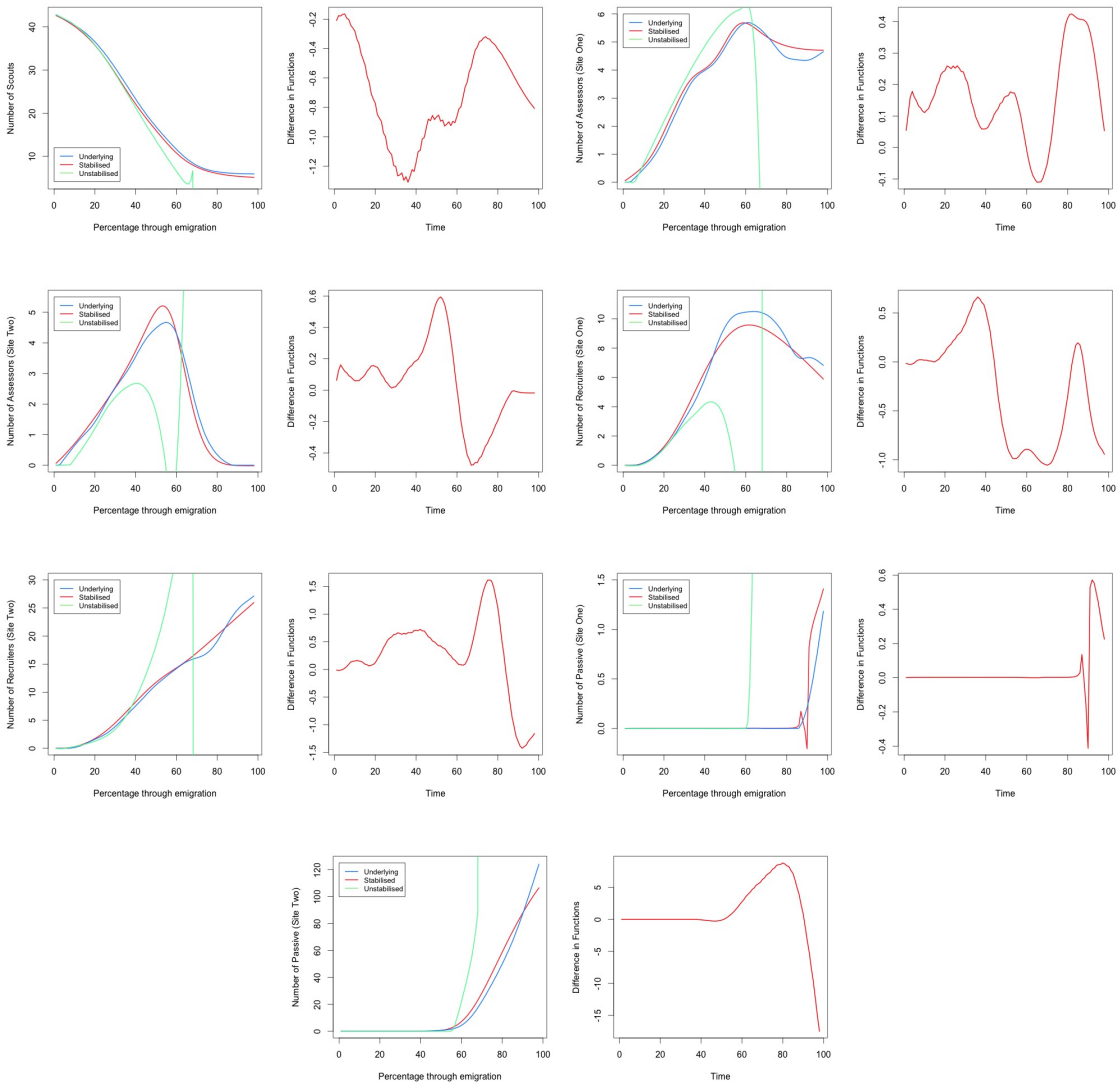


Figure 34: Results from attempting to stabilise the differential equation-based summaries of the SPACE model via brute force. Blue represents the derived general behaviours of the SPACE model, red the stabilised SPACE model, and green the unstabilised SPACE result. The plots to the right are then the differences between functions (*note:* the data plots and difference plots are scaled differently). We see that using this method, it is possible to define differential equation models which reasonably represent the general behaviours we are interested in. These results are given using percentage through the emigration due to certain limitations of the Eureqa API, as detailed in the source code.

Initially we see that the results of this approach are reasonably accurate, but without any of the compromises of the other two methods. In *Section* 9.1 we saw that by optimising over the parameters we could only increase the accuracy of a small proportion of the model. In *Section* 9.2 we saw that although it was possible to iteratively generate an accurate model, this was limited in its extensibility and in the relationships between variables it defined.

This new method offers none of these compromises. It is reasonably accurate over all of the roles, and there are no constraints on the relationships between variables. However, we do see that there are some limitations to this approach, for example the P_1 function generates a negative number of ants. Another drawback of this approach is in terms of the computational power needed for the calculations.

The model contains seven different variables: $S, A_1, A_2, P_1, P_2, R_1$ and R_2. We assume for simplicity that each variable displays n solutions across the accuracy-parsimony Pareto front according to Eureqa. This demonstration is extendible to cases where each variable displays different numbers of solutions across the Pareto front, but to save on notation we assume they have the same number, n. We then consider the case where we look to fit the model to m different emigrations appended back to back. The number of models that must be constructed, ran and the R^2 value taken becomes:

$$N_{models} = mn^7 \tag{9.3.1}$$

Where N_{models} is the number of models. As the work done for each of these models requires a large number of calculations in itself, the time and power taken for using a brute force approach increases rapidly with the number of solutions displayed on the Pareto front. For example, even when limiting each equation to only four solutions across the Pareto front and examining just a single emigration, the time taken to construct and compare the models was 8 minutes and 20 seconds on a standard laptop computer. In addition, as more sets of emigration conditions are introduced to the training set, Eureqa will require more time and computational power to find any solutions at all with a satisfactorily small error and complexity. As well as this, as we aim to model more and more emigrations with the same number of variables, the complexity of the equations needed to fully capture each of these emigrations will inevitably grow larger. There then may no longer exist simple models that can accurately capture the behaviours of each emigration.

The final point to be made aware of when taking this approach is the limitation of the solutions. Eureqa presents all of the solutions with minimal complexity and error *per equation*. As it does not consider the model as a whole, it does not consider how that compromise between error and complexity will affect the entire model. For example, there may be a solution for $\frac{dS}{dt}$ that, when the error is measured just in terms of its predictive capacity for S, does not display an optimal error-complexity balance. However, this solution, when considered as part of the entire model may reduce the error of the model, and so be optimal in this sense. Again, we are limited by the solutions that Eureqa presents.

9.4 Discussion

Within this section we have offered a number of experiments in order to address the weaknesses seen earlier with the Eureqa system, where it cannot consider the model as a whole. Each offered a different method of coping with the problems within the system, each with their relative merits and flaws.

In *Section* 9.1 we attempted to stabilise the model by optimising over the parameters, with the intention of letting Eureqa define the relationships between variables in the model, and using the parameter optimisation to induce stability over the model as a whole. Although this offered improvement in some roles, overall we still saw a large degree of instability.

This was followed by an iterative approach to enforcing accuracy across the Eureqa defined model (*Section* 9.2). Although the model defined was accurate, there were two major weaknesses in this technique. Initially, we were limited in the relationships between variables we could define. Secondly, this approach could not be extended in the case where we looked to define a model for multiple emigrations as the degrees of freedom per set of emigration conditions did not vary sufficiently.

The final proposed method for stabilising the equations was based on a brute force approach (*Section* 9.3). Within this approach we considered all possible combinations of the solutions presented by Eureqa, per equation, and the accuracy of the models they presented. We showed that this was capable of presenting reasonable approximations of the underlying dynamics we wanted to capture. However,

there were drawbacks and potential problems in extending the technique. Most notably the computational power necessary to make such an approach function on a wider scale, and the potential for multiple emigrations requiring expressions for variables so complex that no stable solution existed.

The success of the brute force approach also represents a very interesting finding relating accuracy, complexity and stability in systems of differential equations. In *Section* 8.2.2 we demonstrated how as equations become more complex, the stability of the overall model suffers as small errors propagate through the system. In *Section* 9.3 we showed that by sacrificing accuracy for complexity on an individual equation scale, we could increase the accuracy of the model as a whole.

This is of interest in that it contradicts the intuitive approach of choosing equations that minimise error on an equation by equation basis in order to define the best model. We have shown that this is not the most effective tactic. Due to our earlier findings relating complexity and stability, we realise that when defining a system of differential equations, it is more beneficial to choose equations with a lower complexity, even if they have a relatively high error, as this will enforce a higher level of accuracy over the model as a whole.

This concludes the section on experimentation done in order to rectify the weaknesses of the Eureqa system when applied to deriving the differential equation-based summaries of SPACE. Each approach had relative merits and flaws that have been presented in order for further work in the field to take place. To summarise the findings of the section, contained below is a table of the R^2 values for each of the roles and each of the approaches.

Stabilisation Technique	$\frac{dS}{dt}$	$\frac{dA_1}{dt}$	$\frac{dA_2}{dt}$	$\frac{dR_1}{dt}$	$\frac{dR_2}{dt}$	$\frac{dP_1}{dt}$	$\frac{dP_2}{dt}$
Unstabilised	0	0	0	0	0	0	0
Parameter Optimisation	0.999114	0.829001	0	0.000115	0.999590	0.993718	0.998377
Iteration	0.999797	0.988737	0.996330	0.998879	0.998750	0.999976	0.999394
Brute Force	0.996531	0.985323	0.975534	0.973888	0.992310	0.600222	0.981224

Table 11: Table of R^2 values for different roles when stabilised using different techniques. The MAE and MSE were omitted as they do not offer as much insight as the R^2 values due to being proportional to the range of values taken by the different roles. They are, however, included within the source code, and can be viewed there.

We now look to conclude the thesis by evaluating the work that has been carried out and discussing how what we have learnt of the strengths and weaknesses of Eureqa can be turned into strategies for future research.

Part IV
Project Evaluation

This section will cover an evaluation of the work done within the project, and will be split into several parts. The first will be an evaluation of the contributions to various fields made by the project (*Section* 10). In *Section* 2 we laid out a number of expected contributions: here we re-examine these expectations in light of what was achieved, evaluating the project work in its relevance to current research. As the contributions to the project are so tightly linked to the project aims and objectives, the evaluation of the contributions will also serve as the evaluation of having achieved what was laid out in *Section* 1. In the conclusion we will also explicitly relate the findings of the thesis to each of the objectives in turn.

Following this, we will evaluate the current SPACE project as a whole (*Section* 11). As seen in *Section* 5.2.4, SPACE is an ongoing area of work at the University of Bristol, of which this project is only a small part. The work done towards the thesis has an effect on the larger project and here we contextualise what has been achieved within this scope.

The next sections will focus on evaluating Eureqa as a tool for the contextual aim of the project: deriving differential equation-based summaries of the agent-based models (*Section* 12). Following this we evaluate the realism of aims of the project (*Section* 13), discussing how what we have learnt from the work done within the thesis impacts upon this.

10 An Evaluation of Contributions and Achievement of Objectives

This part of the evaluations will be broken into three. The first section will cover the contributions of the project to the field of computer science (*Section* 10.1). The second will cover the contributions to biology (*Section* 10.2). We will finish with a summary of the resources generated by the project, and how they are relevant to future research (*Section* 10.3).

10.1 Contributions to the Field of Computer Science

In *Section* 2.1 three main contributions to the field of computer science were outlined.

- **An evaluation of the strengths and weaknesses of equation discovery applied to discovering systems of differential equations.**
- **A set of results from experimentation done in an attempt to address the current weaknesses in the application of equation discovery to systems of differential equations.**
- **Findings relevant to future attempts to derive differential equation-based summaries of the agent-based models of ant population dynamics.**

We review the work done towards each of these contributions in light of the contents of the thesis.

An evaluation of the strengths and weaknesses of equation discovery applied to discovering systems of differential equations: In *Section* 7.2.1 we verified the findings of the most recent research in the field by showing that Eureqa was capable of discovering modelling equations for single

quantities [20, 55, 79]. We then extended away from modelling single quantities and into our area of interest, modelling systems of differential equations. In *Section* 7.2.2 we showed that, when there are well-defined underlying systems of differentials to find, Eureqa can discover accurate sets of equations approximating these systems, concurring with the existing evidence [80]. These can be seen as the strengths of the system. This also marks having achieved our first objective from *Section* 1, where we looked to carry out a preliminary analysis of Nutonian Eureqa.

In *Section* 8 we showed that, as data becomes more complex and there exists no simple underlying formulae, that equations must become more complex to capture the behaviours of the model in full. This in turn leads to a high sensitivity of the model to errors in the equations and small errors propagate through the system and it becomes unstable. As Eureqa only considers the model on an equation by equation basis, and has no concept of the model in its entirety it does not account for this. This can be seen as a weakness of the Eureqa system. The work done within this section also marked having completed our second objective of demonstrating and rationalising the weaknesses of Nutonian Eureqa applied to deriving systems of differential equations from complex data.

A set of results from experimentation done in an attempt to address the current weaknesses in the application of equation discovery to systems of differential equations: The strengths and weaknesses of the Eureqa system as documented above were turned into a set of experiments attempting to use the strongest points of the system to address the weakest and to derive a single, stable case of the SPACE model (*Section* 9). Each of the approaches employed had their own merits and flaws, but the findings can be used to drive forward future work in the area. The experimentation within this section also achieved the third objective of attempting to address the weaknesses seen in Eureqa within our given application.

Findings relevant to future attempts to derive differential equation-based summaries of the agent-based models of ant population dynamics: The entire project was considered in the context of deriving differential equation-based summaries from existing agent-based models of *Temnothorax albipennis* population dynamics. The lessons learnt in stabilising models retrieved from Eureqa can then be applied in the process of achieving our contextual aim, and marked the completion of our fourth and final objective, discovering findings relevant to future work on the subject.

In light of the work done in the thesis, it is reasonable to say the proposed contributions to the field of computer science have been made, and our objectives achieved.

10.2 Contributions to the Field of Biology

One main contribution to the field of biology was suggested.

- **A set of results that can inform future efforts associated with deriving differential equation-based summaries of the agent-based models, SPACE and AH-HA.**

In *Section* 9 we outlined how the strengths and weaknesses of Eureqa exposed within the project informed a set of experiments addressing the flaws in the system. The results of these experiments can then be used to inspire future efforts on deriving the differential equation-based summaries of SPACE and AH-HA. This leads us to conclude that our proposed contribution to the field of Biology has been achieved.

10.3 Contributions in the Form of Project Resources

A number of useful resources have been generated throughout the duration of this project, including:

The Pratt and Planqué models: These models have been written with both fixed and variable parameters and can be used in future research.

The AH-HA model: The AH-HA model was originally written in 2004, being augmented until 2006. The model was written in *Java* [64], with dependencies on a number of libraries, mainly based around the agent-based *Repast* [62] framework. Taken from previous work [89], these have been gathered together, and the model rewritten in order to generate the correct form of data for this project. This will be vital in future research in the field as the AH-HA model will need to be studied in the same context as SPACE, Pratt and Planqué.

SPACE and AH-HA training sets: Comprehensive training sets were also generated from the SPACE and AH-HA models for the project. In the case of AH-HA, such training sets can be generated rapidly, however the SPACE model runs in real time, and so gathering data from the model is a time consuming task. Having access to a range of data already generated from SPACE will greatly aid prospective further research.

A nonparametric regression-based method of smoothing and deriving general behaviours from the agent-based models: Deriving simple curves from the complex SPACE and AH-HA model data without using restrictive parametric techniques is an important step surrounding the aim of deriving differential equation-based summaries of the agent-based models. Using the raw data, no results can be found in the equation discovery process, therefore smoothing was a requisite of the project. A mathematically tenable system for doing so will aid future research as a way of turning data with complex differentials into the core trends seen within the data, and simplifying the problem of deriving the differentials.

The results of the experimentation done within the thesis: The results of the thesis will serve as a warning of the type of problems that can occur using similar systems to Eureqa, and as a demonstration of the relationship between complexity and stability when deriving systems of differential equations. These problems will not be localised to the field of equation discovery, but will occur in any method that does not consider the model as whole when attempting to derive representative equations.

11 An Evaluation of the Wider SPACE Project

SPACE is an ongoing, growing project at the University of Bristol of which the work done within this thesis is only a small part. Therefore we evaluate our findings within the context of the wider project.

Initially it is important to recognise that this is a completely new direction for the work done surrounding SPACE. Previously efforts have been concentrated on building or augmenting the existing SPACE model [30, 78, 84] or on developing a robotic ant arena [7]. This is the first attempt to derive modelling differential equations from the data generated by SPACE. This new area of work has highlighted some points regarding the wider SPACE project.

11.1 Problems with Time Scaling in Unity

Each simulation within the SPACE model takes place in real time, the impact of which is that the process of gathering data from multiple simulations takes many hours to complete. This cannot be rectified as the existing scripts for the SPACE model are not connected to the Unity3D timekeeping functionality, which means that the existing time scaling functions cannot be run. In projects such as this, which are significantly data reliant, it would be advantageous to be able to generate data more rapidly.

11.2 Existing Computational Errors

The existing SPACE model also intermittently produces an error: *'NullReferenceException: Object reference not set to an instance of an object AntManager.NearerOld () (at Assets/Scripts/AntManager.cs:731)*

AntSenses.OnTriggerEnter (UnityEngine.Collider other) (at Assets/Scripts/AntSenses.cs:54)'. This appears to be due to ants losing each other during tandem runs, therefore the pointers that originally led from the follower to the leader become *null*. Projects in following years can aim to address this error.

12 An Evaluation of Nutonian Eureqa as a Tool for Deriving Differential Equation-Based Summaries of the Agent-Based Models

A lot of the work within this project has gone into discerning the strengths and weaknesses of the Nutonian Eureqa system for equation discovery. Therefore we evaluate it as a suitable tool for our aim of deriving differential equation-based summaries of the agent-based models.

We have seen that Eureqa is powerful when applied to finding equations for modelling single quantities, and that it is capable of discovering differential equation-based models where there are underlying models to find. However, we also demonstrated the instabilities of more complex models found using Eureqa, and that it was not immediately capable of discovering the differential equation-based summaries of SPACE and AH-HA. These two factors bring us to evaluate Eureqa's suitability as a tool for achieving our contextual aims.

12.1 Alternative Equation Discovery Systems to Nutonian Eureqa

As we have demonstrated that Nutonian Eureqa is not directly capable of achieving our desired results, one alternative would be to employ a different system, potentially LAGRAMGE. Although in *Section* 6 we ruled out alternative systems due to their limitations in requiring a strong declarative bias, from the results of the thesis we argue that it may be worth experimenting with defining a grammar for LAGRAMGE (*note: Section* 13 will offer a supplementary argument for experimenting with LAGRAMGE).

However, defining a grammar for the problem seems almost synonymous with the work done by Pratt and Planqué in defining their original models. In *Sections* 5.2.1 and 5.2.2 we saw that both models used observations of colony emigration behaviours to define model forms, fitting only the parameters to the empirical data. In defining a grammar for the problem we go through a similar process where theory is used for defining the relationships between variables, and empirical data to fit the model. It could be that we achieve very similar results.

12.2 Alternatives to Equation Discovery

The fact that we cannot immediately use contemporary equation discovery techniques to derive the differential equation-based summaries of the agent-based models may suggest that this is not a reasonable aim. Instead another route should be chosen that, similar to the process of using differential equations, can take a set of parameters and from them generate data modelling an emigration. For example, it would be possible to train a neural network [35] to do this using the existing SPACE data. However, other methods do not address two of the main motivations for attempting to derive differential equation models specifically.

Initially, we aim to derive differential equation-based summaries of the agent-based models as this will standardise the available models in the field to the same form. In addition, having a differential equation model means that the model can be analysed analytically. These two factors prevent us from employing alternative strategies to attempting to find a differential equation model for SPACE and AH-HA.

13 An Evaluation of the Aims of the Project

The final evaluation of the work done within the project is in terms of what we set out to achieve. For the most part, the main objective of the project was to derive differential-equation based summaries of the agent-based models. We now consider how plausible these aims are, both in terms of execution and on a theoretical basis.

13.1 Project Execution Feasibility

First of all, we have shown that deriving systems of differential equations from complex data is a non-trivial task. We have outlined one of the implementation based challenges as the time taken to generate the data necessary from the SPACE model (*Section* 12). This, coupled with the time taken to run the equation discovery algorithm (greater than one hour per equation, with seven equations per model) makes progress and prototyping ideas an extremely slow process, especially within the given time frame. Minor errors, or ideas that require experimentation, require days of work to fix or try with little to show in the way of results. In light of this, large amounts of time has gone into experimentation with ineffective stabilisation techniques that have not been reported within the thesis, and the time taken for implementing new ideas, rectifying errors and generating evidence to support hypotheses should be considered in future work.

In addition, as equation discovery is such a recent field, and Eureqa such a new tool, there is little in way of support or related research. Many other fields have large amounts of online resources and previous work to be built upon, which is not the case for Eureqa or equation discovery applied to deriving systems of differential equations from complex data.

In terms of project execution, this means a deep understanding of both the mathematical and computer science elements of the project is needed. As there is not a wide knowledge base to draw from, a large amount of wholly original thought and work has gone into the thesis. Choosing and implementing the nonparametric system for deriving general behaviours from the agent-based models and simplifying differentials; recognising that errors are caused by the interaction of variables; demonstrating the relationship between complexity and accuracy; and the three approaches to including a model-wide consideration of stability into our method for deriving differential equation-based summaries of the agent-based models are all novel concepts and implementations.

13.2 Theoretical Justification of the Aims of the Project

We also consider the aims of the project in their theoretical sense. There are two main points relating to this consideration. Initially, we discuss whether or not there exists a set of differential equations that can accurately replicate the SPACE and AH-HA dynamics, and secondly whether it make sense to look for them in the manner in which we have approached the subject.

In answer to the first point, there necessarily exists a set of equations that can, using only the variables presented to it, capture the behaviours of the SPACE and AH-HA models. However, the level of complexity of these solutions is not bounded. To capture these behaviours it may require the modelling equations to become arbitrarily complex. When we review the examples in previous research with Eureqa: the single and double pendula and the double-mass air track [79], although Eureqa discovered simple relationships governing their motion, this was as it was known that there were such relationships to discover. In the case of the AH-HA and SPACE models, there is no such guarantee.

This then raises the question of, once found, whether or not the differential equation-based summaries of the agent-based models will be of any use. Even in the case of them perfectly representing the training data, this may be an effect of interpolation, rather than them representing some hidden, underlying truth.

In relation to whether it makes sense to look for systems of differential equations in the manner we have attempted, we review, at the most fundamental level, why we construct differential equation models. Such models are used to represent, and more importantly understand, systems of quantities and how the values of quantities relate to each other. With this in mind, in some senses it is justifiable to take the attitude of defining models using the process of giving an algorithm access to data from variables x and y, reviewing the output and drawing conclusions of the form *'There exists a relationship $z(x, y)$, as shown by the data'*. However, within this approach, there is no distinction between correlation and causation, and we fall into the trap of interpolating models to the training data: solutions derived using this method do not necessarily tell us anything about the mechanics underpinning the system, only that the given equations model the training data accurately.

A more palatable approach is to define the expected relationship between variables x and y, and fit this to the data, drawing a conclusion of the form *'We expected the relationship $z(x, y)$ between variables x and y for this reason, this fits the observed data well, giving support for our reasoning.'* In contrast to the former approach, using this methodology we derive models with sound logic underpinning them and empirical data supporting them.. Historically the second approach has been shown to be both popular and successful [45, 51, 66, 71, 94].

Within the thesis, we have purely employed the first methodology. The aim of the thesis, as much as it was to derive differential equation-based summaries of agent-based models, was to do so using equation discovery. However, we now reflect on whether or not this is the best approach.

Although results can be found that will be able to accurately replicate any training data provided to the algorithm, this is not equivalent to having found equations that capture the mechanics of the system. A potentially better approach would be to define an expected form of a model that would capture the mechanics of the system and fit that to the data, as per the second methodology. This may well involve sacrificing some of the accuracy of the solutions (they will not one be able to one hundred percent accurately reproduce each training emigration in its entirety), but will more accurately reflect what is going on within an emigration.

This approach has already been taken in the traditional sense by the Pratt and Planqué models. However, a compromise between the two tactics is offered by LAGRAMGE. By defining a grammar for the system, we give some logic to any of the solutions derived, but without the severity of the limitations imposed by defining the complete form of the model except the parameters. Implementing LAGRAMGE does come with the possibility that no reasonable solutions exist following the defined grammar, but the motivation for experimenting is still present.

Part V
Further Work and Conclusion

The final part of the thesis will be split into two sections. The first will cover the suggested further work following the findings of the project (*Section* 14). The second section will be the conclusion of the project where we will tie together everything we have learnt throughout the thesis and summarise the work's findings (*Section* 15).

14 Recommended Further Work

This section takes the prior evaluations of the project and turns it into proposed further work in the field. Two different directions are suggested. The first is an entirely new system based on the same logic as Eureqa, but developed specifically for systems of differential equations. The second direction is replacing Eureqa with LAGRAMGE as a system of equation discovery.

14.1 A New System for Equation Discovery in Systems of Differential Equations

In the results and evaluation of *Section* 9.3 we saw that, although the brute force approach to model stabilisation is successful in one sense, it is limited by the results Eureqa displays on the accuracy-parsimony Pareto front for each equation. Even if a result would increase the stability and lower the error of the overall model, if it does not offer an optimal solution in terms of the isolated equation for the single variable, it will not appear on the Pareto front. This means that it cannot form a part of the model formed by the combination of solutions from the front and the brute force approach would not be able to find the optimal model.

One way to address this is to build a new system for equation discovery specifically for systems of differential equations. This would be an ambitious undertaking, but it would rectify a number of problems seen previously.

Data for an emigration could be provided, and expressions formed randomly for each variable using the same techniques as in Eureqa (see *Section* 6.1.2). These expressions would then be combined to create an entire model. The provided data would be split into training and validation sets. Finally the model would be run from initial conditions and a measurement of error taken between the results and the validation set, identically to the brute force approach. Models minimising the error metric would be retained, analogously to the original Eureqa system, but in terms of an entire model rather than single equations.

This rectifies the problem of equations only being available to the brute force approach if they are optimal in terms of the sole variable they represent. It also deals with initial condition problems such as those documented in *Section* 7.3. In addition, as the method has been implemented from scratch, it will be possible to include conditions such that the total number of ants within the model remains constant and the number of ants per role does not fall below zero.

Defining our own system for deriving sets of differential equations would also help us deal with one of the other challenges of this project: the inclusion of probabilistic elements in the processes we wish to model. In *Section* 8.1.2 we attempted to reduce the effect of the random model components using nonparametric regression. However, this does not address the problem entirely.

Due to the probabilistic components of the models, it could be argued that systems of equations representative of AH-HA and SPACE do not necessarily have to replicate exactly the emigrations given to them in the training data, they must only generate data that reflects a reasonable emigration pattern under the probabilities dictated by the models. This opens up an interesting question regarding the definition of fitness functions for determining the accuracy of any differential equations suggested by our new system.

If we no longer look to model training data exactly, and instead want to capture more general behaviours, it makes less sense to measure model fit in terms of quantities such as the MAE, MSE and R^2 values. Instead, fit could be measured using emergent properties of the generated system such as emigration time, the proportion of ants at different nests at the end of an emigration, the speed-accuracy trade off (*Section* 4.3.1) or the speed-cohesion trade off (*Section* 4.3.2). By generating models, running them multiple times and comparing these factors between the training data and the generated model, this will allow us to see if the generated models behave similarly to the training data in a more general sense, rather than if they can simply recreate the same data under the same conditions.

Using these methods would represent a novel and interesting new approach to equation discovery, not only in deriving systems of differential equations, but also in capturing probabilistic elements of data.

14.2 LAGRAMGE as an Alternate Tool For Equation Discovery

Finally we cover the potential to use LAGRAMGE as a method of equation discover. In *Section* 13 we discussed the problem of interpolating data to retrieve a solution in lieu of capturing the underlying mechanics of the system. Although LAGRAMGE requires the definition of a grammar, this guarantees any solutions will follow the given form. If this is well-defined then this limits the ability of solutions to interpolate to the data, and increases the probability of finding a solution that reflects the underlying dynamics of the system.

15 Conclusion

We conclude the thesis with a summary of the main findings of the project, relating them to the aims and objectives laid out in *Section* 1. The central aim of the project was to investigate the application of equation discovery to modelling the behaviour of social insects. We chose to implement this investigation within the domain of the population dynamics of the ant species *Temnothorax albipennis*, and the problem of deriving differential equation-based summaries of the agent-based models, taking the following steps to achieve our goal.

We initially built up our experience of the equation discovery system Eureqa, reinforcing the latest research within the field by showing it capable of discovering equations for modelling single quantities (*Section* 7.2.1) and simple, well-defined systems of differential equations (*Section* 7.2.2). This achieved our first objective of carrying out a preliminary analysis of Nutonian Eureqa.

The following section expanded past the current work done within the field and attempted to derive a differential equation model for a more complex problem (*Section* 8). In doing this we demonstrated that as data becomes more complex, the equations necessary to model the data become more complex also. As the number of variables remains the same, this in turn means that the equations become more sensitive to errors within the values of the variables. The overall effect of this is that small errors rapidly propagate throughout the system and the system of differential equations becomes unstable.

We attributed this property to the methodology Eureqa employs when deriving the system of differential equations (*Section* 8.2.2), where Eureqa considers the model on an equation by equation basis and does not account for interactions between equations. This in turn means that the system gives no consideration to the stability of the model as a whole. We had then achieved our second objective of demonstrating and rationalising the weaknesses of equation discovery applied to deriving systems of differential equations from complex data.

To address this problem, we took the weakness we had demonstrated within the system and attempted a number of different methods (*Section* 9). Each of these methods revolved around the fact that Eureqa did not consider the model as a whole, and tried to compensate for this, enforcing stability. In the first method we let Eureqa define the proportionalities within the model, and optimised over the parameters to minimise model-wide error (*Section* 9.1). We showed that although this made the model more accurate over some terms, the limiting factor of the Eureqa defined relationships between variables meant that the whole model did not reflect the underlying behaviours.

Following this we attempted an iterative method to stabilise the model, where we built up stable terms from certain initial known quantities (*Section* 9.2). We showed that although this would define an accurate model, this was not a sufficient solution as it did not extend to the multiple emigration case and required some order of precedence to the relationships between variables.

Finally we implemented a brute force approach to stabilising the equations (*Section* 9.3). We built all possible models using the solutions presented by Eureqa on the accuracy-parsimony Pareto front, taking the most accurate model from the results. We showed that this offered an improvement over previous techniques, without any of the compromises. These experiments marked the achievement of our third objective, carrying out experimentation attempting to address the weaknesses in the application of equation discovery to deriving systems of differential equations from complex data. All of the above were done in the context of *Temnothorax albipennis* emigrations, and so it is also fair to say we have achieved our final objective of discovering findings relevant to future attempts to derive differential equation-based models of the agent-based models of ant population dynamics.

The thesis culminated in a critical evaluation of the steps taken in the project, and their relevance to future research within the area (*Part* IV). The final work done within the project saw these evaluations being refined into an actionable set of directions for further work.

This ends the work and findings of the project, and so the thesis.

References

[1] Unity3D Game Engine. http://unity3d.com/. Accessed: 2015-04-23.

[2] University of Bristol Ant Lab colony image. http://www.bristol.ac.uk/biology/research/behaviour/antlab/. Accessed: 2014-23-04.

[3] University of Bristol Predictive Life Sciences image. http://www.bristol.ac.uk/media-library/sites/research/images/predictive-life.jpg. Accessed: 2014-23-04.

[4] M. J. Best and M. Chakravarti. Active set algorithms for isotonic regression: A unifying framework. *Mathematical Programming, Volume 47, pp. 425 - 439.*, 1990.

[5] E. Bonabeau, M. Dorigo, and G. Theraulaz. Swarm intelligence: from natural to artificial systems. *New York: Oxford University Press*, 1999.

[6] L. Boneva, D. Kenfall, and I. Stefanov. Spline transformations: Three new diagnostic aids for the statistical data analyst (with discussion). *J.R. Statistics Society B-33, pp. 1 - 70.*, 1971.

[7] C. Calder. Applying 3d printing to ant colony emigration experiments. *Master's Thesis, University of Bristol*, 2015.

[8] S. Camazine, P. K. Visscher, and R. S. Vetter. House-hunting by honey bee swarms: collective decisions and individual behaviour. *Insectes Soc 46:348-360*, 1999.

[9] M. Collett and T. S. Collet. Memory use in insect visual navigation. *Nature Reviews Neuroscience 3, 542-552, July*, 2002.

[10] Microsoft Corporation. *c#* language specification. https://www.microsoft.com/en-us/download/details.aspx?id=7029. Accessed: 2015-08-08.

[11] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Information Theory Society. Volume IT-13, No. 1, January, pp. 21-27*, 1967.

[12] L. Davies and A. Kovac. R package ftnonpar. https://cran.r-project.org/web/packages/ftnonpar/index.html. Accessed: 2015-08-08.

[13] T. L. Dean and M. Boddy. An analysis of time-dependent planning. *Proceedings of Seventh National Conference on Artificial Intelligence, pp. 49-54*, 1988.

[14] UC Berkeley EECS Department. Ptplot java library. http://ptolemy.eecs.berkeley.edu/java/ptplot/. Accessed: 2015-08-08.

[15] A. Dornhaus, N. R. Franks, R. M. Hawkins, and H. N. S. Shere. Ants move to improve: colonies of leptothorax albipennis emigrate whenever they find a superior nest site. *Animal Behaviour, Volume 67, Issue 5, May, Pages 959âĂŞ963*, 2004.

[16] L. Dumbgen and A. Kovac. Extensions of smoothing via taut strings. *Electronic Journal of Statistics, Volume 3, pp. 41 - 75.*, 2008.

[17] L. Duran, M. Fournier, N. Massei, and J.-P. Dupont. Assessing the nonlinearity of karst response function under variable boundary conditions. *Journal of Computer and System Sciences*, 2014.

[18] S. Dzeroski and L. Todorovski. Discovering dynamics: from inductive logic programming to machine discovery. *Journal of Intelligent Information Systems, 4: 89-108*, 1977.

[19] S. Dzeroski and L. Todorovski. Discovering dynamics. *Proceedings of the Tenth International Conference on Machine Learning, pp. 97 - 103*, 1993.

[20] E. Finn. Predicting the perceived quality of a first person shooter game: the team fortress 2 t-model. *Worcester Polytechnic Institute. Bachelor's Degree Dissertation.*, 2013.

[21] E. Fix and J. L. Hodges. N/a. *Never Published*, 1951.

[22] CERN European Organization for Nuclear Research. Colt project java library. *Journal of Statistical Software*, 8(11):1–20, 2003.

[23] Apache Software Foundation. Apache commons csv. https://commons.apache.org/proper/commons-csv/. Accessed: 2015-08-08.

[24] Boost Software Foundation. Boost c++ libraries. http://www.boost.org/. Accessed: 2015-08-08.

[25] Python Software Foundation. *The Python Language Reference - Python 2.7.10*, 2015.

[26] N. R. Franks, A. Dornhaus, J. Fitzsimmons, and M. Stevens. Speed versus accuracy in collective decision making. *Proceedings of the Royal Society, London, B 270:2457-2463*, 2003.

[27] N. R. Franks, E. B. Mallon, H. E. Bray, M. J. Hamilton, and T. M. Mischler. Strategies for choosing between alternatives with different attributes: exemplified by house-hunting ants. *Animal Behaviour, Volume 65, Issue 1, pp. 215 - 223, January*, 2003.

[28] N. R. Franks and T. Richardson. Teaching in tandem-running ants. *Nature, 439(7073):153, January*, 2006.

[29] N. R. Franks, T. O. Richardson, N. Stroeymeyt, R. W. Kirby, W. M. D. Amos, P. M. Hogan, J. A. R. Marshall, and R. Schlegel. Speed-cohesion trade-offs in collective decision making in ants and the concept of precision in animal behaviour. *Animal Behaviour, 85(6): 1233-1244*, 2013.

[30] M. S. Garrad. Decentralised decision making in temnothorax albipennis colonies. *Master's Thesis, University of Bristol*, 2013.

[31] Switzerland: International Organization for Standardization (ISO). Geneva. Iso international standard iso/iec 14882:2014(e) - programming language c++. https://isocpp.org/std/the-standard. Accessed: 2015-08-08.

[32] D. M. Gordon and A. E. Hirsh. Distributed problem solving in social insects. *Annals of Mathematics and Artificial Intelligence 31:199-221*, 2001.

[33] V. Grimm and S. F. Railsback. *Individual-based Modeling and Ecology.* Princeton University Press, 2005.

[34] Walt Disney Internet Group. Teatrove java library. http://teatrove.sourceforge.net/. Accessed: 2015-08-08.

[35] Simon Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998.

[36] E. Hunt. PhD Thesis. *University of Bristol*, Ongoing.

[37] Brian W. Kernighan. *The C Programming Language.* Prentice Hall Professional Technical Reference, 2nd edition, 1988.

[38] J. R. Koza. *Genetic Programming: On the programming of computers by the means of natural selection.* MIT Press, Cambridge, MA, 1992.

[39] V. Krizman, S. Dzeroski, and B. Kompare. Discovering dynamics from measured data. *Electrotechnical Review, 62: 191-198*, 1995.

[40] P. W. Langley. BACON: A production system that discovers empirical laws. 1977.

[41] P. W. Langley. *Scientific Discovery: Computational Explorations of the Creative Processes.* MIT Press, 1977.

[42] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics 2: 164-168.*, 1944.

[43] U. Ligges and M. Mächler. Scatterplot3d - an r package for visualizing multivariate data. *Journal of Statistical Software*, 8(11):1–20, 2003.

[44] Object Refinery Limited. Jfreechart java library. http://www.jfree.org/jfreechart/. Accessed: 2015-08-08.

[45] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences 20 (2): 130-141*, 1963.

[46] E. B. Mallon, S. C. Pratt, and N. R. Franks. Individual and collective decisionmaking during nest site selection by the ant leptothorax albipennis. *Behavioral Ecology and Sociobiology, 50(4):352âĂŞ359, September*, 2001.

[47] E.B. Mallon and N. R. Franks. Ants estimate area using buffon's needle. *Proceedings of the Royal Society Interface, 3:243-254*, 2006.

[48] E.B. Mallon, S.T. Mugford, and N. R. Franks. The accuracy of buffon's needle: a rule of thumb used by ants to estimate area. *Behavioural Ecology, 12:655-658*, 2001.

[49] J. A. R. Marhsall, A. Dornhaus, N. R. Franks, and T. Kovacs. Noise, cost and speed-accuracy trade-offs: decision-making in a decentralized system. *Journal of the Royal Society Interface, 3(7):243-254*, 2006.

[50] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics 11 (2): 431-441.*, 1963.

[51] J. C. Maxwell. Dynamical theory of the electromagnetic field. *Proceedings of the Royal Society of London, Volume 13 pp. 531 - 537*, 1864.

[52] P. E. Merlotti. Simulation of artificial ant's behaviour in a digital environment. *Graduate Seminar in Artificial Intelligence Evolutionary and Adaptive Computation*, 200.

[53] M. Möglich, U. Maschwitz, and B. Hölldobler. Tandem calling: a new kind of signal in ant communication. *Science, 13:186(4168):1046-7*, 1974.

[54] J. J. Moré. The levenberg-marquardt algorithm: Implementation and theory. *Lecture Notes in Mathematics Volume 630, 1978, pp 105-116*, 2006.

[55] D. Moreno-Sanchez, J. Tijerina-Aguilera, and A. Y. Aguilar-Villarreal. Use of symbolic regression for lean six sigma projects. *Proceedings of the 2015 Industrial and Systems Engineering Research Conference*, 2015.

[56] R. M. Mottola. Visual comparison of ensemble averaged transverse arching profiles of golden age cremonese violins and curtate cycloid curves. *Savart Journal 1, no. 2*, 2012.

[57] K. M. Mullen. R package minpack.lm. https://cran.r-project.org/web/packages/minpack.lm/index.html. Accessed: 2015-08-08.

[58] E. A Nadaraya. On estimating regression. *Theory of Probability and its Applications, Volume 9, pp. 141 - 142*, 1963.

[59] A. Noble. Ant of the genus *Temnothorax albipennis*: Image. Taken from AntWeb.org.

[60] A. Noble. Dorsal view of an ant of the genus *Temnothorax albipennis*: Image. Taken from AntWeb.org.

[61] A. Noble. Top view of an ant of the species *Temnothorax albipennis*: Image. Taken from AntWeb.org.

[62] M.J. North, N. T. Collier, and J. R. Vos. Implementations of the repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation, Vol. 16, Issue 1, pp. 1-25*, 2006.

[63] The Mathematical Theory of Communication. *Shannon, C. E. and Weaver, W.* University of Illinois Press, Champaign, 1949.

[64] Oracle. *Java Language and Virtual Machine Specifications*, 2015.

[65] L. W. Partridge, K. A. Partridge, and N. R. Franks. Field survey of a monogynous leptothoracine ant (hymenoptera, formicidae): evidence of seasonal polydomy? *Insectes socieaux. 44 75 âĂŞ 83*, 1997.

[66] R. Planqué, F.X. Dechaume-Moncharmont, N. R. Franks, T. Kovacs, and J. A. R. Marshall. Why do house-hunting ants recruit in both directions? *Die Naturwissenschaften, 94(11):911âĂŞ8, November*, 2007.

[67] R. Planqueé, A. Dornhaus, N. R. Franks, T. Kovacs, and J. A. R. Marshall. Weighting waiting in collective decision-making. *Behavioual Ecology and Sociobiology, Volume 61, Issue 3, pp. 347 - 356, January*, 2007.

[68] S. Pratt. Quorum sensing by encounter rates in the ant *Temnothorax albipennis*. *Behavioural Ecology 16(2): pp 488-496*, 2005.

[69] S. C. Pratt, S. E. Brooks, and N. R. Franks. The use of edges in visual navigation by the ant *Leptothorax albipennis*. *Ethology, 107(12):1125-1136*, 2001.

[70] S. C. Pratt, N. R. Franks, and M. A. McLeman. Navigation using visual landmarks by the ant *Leptothorax albipennis*. *Insectes Sociaux, 49:203-208*, 2002.

[71] S. C. Pratt, E. B. Mallon, D. J. Sumpter, and N. R. Franks. Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant leptothorax albipennis. *Behavioral Ecology and Sociobiology July 2002, Volume 52, Issue 2, pp 117-127*, 2002.

[72] S. C. Pratt, D. J. Sumpter, B. E. Mallon, and N. R. Franks. An agent-based model of collective nest choice by the ant *Temnothorax albipennis*. *Animal Behaviour, Volume 70, Issue 5, pp. 1023 - 1036, November*, 2005.

[73] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

[74] O. Ray. Learning population dynamics in ant colonies. University of Bristol Project Proposals. 2015.

[75] E. J. H. Robinson, N. R. Franks, S. Ellis, S. Okuda, and J. A. R. Marshall. A simple threshold rule is sufficient to explain sophisticated collective decision-making. *PloS one, 6(5):e19981, January*, 2011.

[76] E. J. H. Robinson, F. D. Smith, K. M. E. Sullivan, and N. R. Franks. Do ants make direct comparisons? *Proceedings. Biological Sciences / The Royal Society, 276(1667):2635-41, July*, 2009.

[77] S. Roweis. Levenberg-marquardt optimization. https://www.cs.nyu.edu/ roweis/notes/lm.pdf. Accessed on 9/09/2015.

[78] N. Sampson. Developing a spatially realistic simulation of ant colony emigration. *Master's Thesis, University of Bristol*, 2013.

[79] M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science, Vol. 324, no. 5923, pp. 81-85*, 2009.

[80] M. D. Schmidt, R. V. Ravishankar, J. W. Jenkins, J. E. Hood, S. Abhishek, J. P. Soni, J. P. Wikswo, and H. Lipson. Automated refinement and inference of analytical models for metabolic networks. *Proceedings of the ACM SIGART International Symposium on Methodologies for Intelligent Systems, pp. 296 - 307*, 1986.

[81] T.D. Seeley and S. C. Buhrman. Group decision making in swarms of honey bees. *Behavioural Ecology and Sociobiology 31: 375-383*, 1999.

[82] A. B. Sendova-Franks and N. R. Franks. Division of labour in a crisis: task allocation during colony emigration in the ant leptothorax unifasciatus (latr.). *Behavioral ecology and sociobiology, 36(4):269-282*, 1995.

[83] Karline Soetaert, Thomas Petzoldt, and R. Woodrow Setzer. Solving differential equations in r: Package desolve. *Journal of Statistical Software*, 33(9):1–25, 2010.

[84] G. Southgate. Simulating spatial trajectories in ant colony emigrations. *Master's Thesis, University of Bristol*, 2015.

[85] C. M. Tigaret, K. Tsaneva-Atanasova, G. L. Collingridge, , and J. R. Mellor. Wavelet transform-based de-noising for two-photon imaging of synaptic ca2+ transients. *Biophysical Journal, Volume 104, Issue 5, pp. 1006 - 1017.*, 2013.

[86] L. Todorovski. Declarative bias in equation discovery. *MSc Thesis, Faculty of Computer and Information Science, Ljubljana, Slovenia*, 1998.

[87] L. Todorovski. *Enyclopedia of Machine Learning.* Springer Press, pp. 327 - 330, 2010.

[88] L. Todorovski and S. Džeroski. Declarative bias in equation discovery. *Proceedings of the Fourteenth International Conference on Machine Learning, Morgan Kaufmann, San Mateo, CA, pp. 376-384*, 1997.

[89] A. Venn. An improved agent-based model of the speed-accuracy trade-offs in house-hunting ants. *Master's Thesis, University of Bristol*, 2015.

[90] T. Washio and H. Motoda. Discovering admissible models of complex systems based on scale-types and identity constraints. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, pp. 810-817*, 1997.

[91] G.S Watson. Smooth regression analysis. *Sankhya, The Indian Journal of Statistics, Volume A, pp. 359 - 372*, 1964.

[92] Hadley Wickham. Reshaping data with the reshape package. *Journal of Statistical Software*, 21(12):1–20, 2007.

[93] L. H. Yang, Y. Zhao, Y. Fan, Y. Zhu, and Yu J. Peak power modeling for join algorithms in dbms. *Journal of Computer and System Sciences*, 2014.

[94] J. A. Yorke and W. N. Anderson. Predator-prey patterns (volterra-lotka equations). *PNAS, vol. 70, issue 7, pp. 2069-2071*, 1973.

[95] S. Zilberstein. Using anytime algorithms in intelligent systems. *AI Magazine, Volume 17, Number 3*, 1996.

Appendix

The four sections of the appendix are as follows.

- **A. Experimenting with Methods for Capturing Periodicity:** In *Section* 7.2.1 we claimed that using trigonometric functions as a method for capturing periodicity defined simpler solutions than using modulus building blocks. Here we offer a small amount of experimentation to support that claim.
- **B. Planqué Variable Parameters:** The table of uniform distribution parameters used in *Section* 7.2.2.
- **C. Source Code and Technologies:** Details of all code, languages and resources used in the implementation of the project.
- **D. Notation and Technical Definitions used in the Project:** Details of the mathematical notation and definitions employed within the thesis.

A Experimenting with Methods for Capturing Periodicity

In *Section* 7.2.1 we claimed that trigonometric functions were a better method for capturing periodicity in equations than modulus building blocks. To support this we offer experiments with two functions containing well-defined periodic components, but with complex analytical forms. Data with added Gaussian noise of mean zero and variance one from each function was provided to Eureqa for ten minutes, and the R^2 goodness of fit, MAE and MSE were taken between the solution and the underlying data. These are reported in the below table to offer weight to our decision to use trigonometric functions as a method for capturing periodicity.

$$y = \begin{cases} 0 & 0 \leq x < 10 \text{ or } 20 \leq x < 30 \text{ or } ... \\ 20 & otherwise \end{cases} \tag{A.0.1}$$

$$y = \frac{1}{500}x^3 + \frac{1}{4000}x^2 + \frac{1}{1000}x + \begin{cases} 0 & 0 \leq x < 5 \text{ or } 10 \leq x < 15 \text{ or } ... \\ 200 & otherwise \end{cases} \tag{A.0.2}$$

Function	R^2 Goodness of Fit		MAE		MSE		Solution Size	
	$Trig$	Mod	$Trig$	Mod	$Trig$	Mod	$Trig$	Mod
A.0.1	0.9936451	0.99364384	0.5990240	0.5988150	0.642432	0.6425587	17	17
A.0.2	0.9999733	0.99997252	2.3782154	2.3026901	9.278472	9.5474793	23	31

Table 12: Results for the experimentation with methods of capturing periodicity in functions. $Trig$ is shorthand for the approach using trigonometric functions. Mod is shorthand for the approach using modulus functions.

In both cases the difference in fit and error is minimal, but the solutions using modulus building blocks has either the same or greater solution size (where solution size is proportional to complexity). Combining this with the fact that a modulus building block is assigned a complexity rating of 4 under the Eureqa system, and trigonometric functions a complexity of 3, this justifies choosing trigonometric functions over modulus as a method of capturing periodicity.

B Planqué (General Form) Parameter Ranges

In *Section* 7.2.2 a number of simulations were generated using the Planqué model with uniformly distributed parameters. The ranges of the uniform distribution for each parameter are given in *Table* 13.

Parameter	Definition	Range
N	Colony size	[240, 260]
F	Fraction of active ants	[0.05, 0.5]
Q	Quorum Threshold.	$[1, 12]$
f	Fraction of post-quorum reverse tandem running time.	$[0.05, 0.15]$
μ	Rate at which active ants at old nest become scouts ($ants^{-1}min^{-1}$).	$[0.01, 0.2]$
λ	Rate at which ants following tandem runs become recruiters ($ants^{-1}min^{-1}$).	$[0.05, 0.15]$
ϕ	Rate at which passive ants are carried to a new nest ($ants^{-1}min^{-1}$).	$[0.1, 0.3]$
k	Rate at which scouts independently become recruiters ($ants^{-1}min^{-1}$).	$[0.0001, 0.001]$

Table 13: Ranges of parameters used for testing Eureqa with the Planqué model.

C Source Code and Technologies

The implementation details of the project are contained within this section. The information has been split to mirror the sections of the project as much as possible. Included in each part are the hardware, software, language and package requirements involved in that specific part of the project implementation.

C.1 Equation Discovery with Nutonian Eureqa

Unless stated otherwise all equation discovery was done using the Nutonian Eureqa desktop application [79] under an academic license. The majority of experiments were run in the University of Bristol High Performance Computing Suite.

C.2 Basic Testing Functions

Code Line Count: 516. All basic testing functions were written in *R* [73], with support from the *3Dscatterplot* [43] library. Equation discovery and the analysis of the results was done natively in Eureqa.

C.3 The Pratt and Planqué Models

Code Line Count: 2,295. The Pratt and Planqué models were written with both fixed and varied parameters in *Python* [25]. Equation discovery was then carried out in Eureqa, with the analysis of results done in *R*.

C.4 The AH-HA Model

Additional Code Line Count: 150. The AH-HA model was written in *Java* [64] with support from the *Colt* [22], *Commons CSV* [23], *JFreeChart* [44], *Ptplot* [14], *Repast* [62] and *TeaTrove* [34] libraries. As part of the project the model has been rewritten to provide the correct output. Details of how to build the model can be found in the supporting documentation enclosed with the source code.

C.5 The SPACE Model

Additional Code Line Count: 224. The SPACE model was originally written in *C#* [10] in the *Unity3D* [1] game engine. Alterations to the model were made in order for it to output the correct form of data, all of which were done within the *C#* scripts.

C.6 Smoothing SPACE and AH-HA

Code Line Count: 621. The smoothing of the SPACE and AH-HA data was done in the language *R* using the *ftnonpar* [12] package implementation of spline smoothing.

C.7 SPACE and AH-HA Model Evaluation

Code Line Count: 375. The SPACE and AH-HA models were generated using the Eureqa desktop application, with the results built and run in *R*.

C.8 Analysing the Relationship Between Complexity and Fit

Code Line Count: 155. The visual and statistical analysis of the relationship between complexity and fit in generating models was done in language *R*.

C.9 Stabilising Differential Equations through Parameter Optimisation

Code Line Count: 313. The experimentation involving the stabilisation of the differential equations using parameter optimisation was done in language *R*. The package *reshape2* [92] was used for reshaping data, *deSolve* [83] for generating systems of differential equations to optimise over, and *minpack.lm* [57] to implement the Levenberg-Marquardt algorithm [42, 50, 54]

C.10 Stabilising Differential Equations Iteratively

Code Line Count: 410. The process of stabilising the systems of differential equations iteratively was done in language *R* and using the Eureqa desktop application.

C.11 Stabilising Differential Equations by Brute Force

Code Line Count: 1,518. The work done on stabilising the differential equations by brute force was done in a mixture of technologies. The Eureqa API was used with a *C++* [31] binding and support from the *Boost* [24] libraries to generate all of the possible solutions across the accuracy-parsimony front. A parser was then written in *Python* to turn the solutions into systems of differential equations and to measure the error.

D Notation and Definitions

This section of the appendix is split into two. The first outlines the notation used throughout the project. The second focusses on defining certain mathematical measures used within the thesis.

D.1 Notation

D.1.1 Functions and Variables

Lower case letters are used to represent functions and variable constants.

Upper case letters are used to represent randomly distributed quantities.

D.1.2 Probability Distributions

$$Q \sim N(\mu, \sigma^2) \tag{D.1.1}$$

The random variable Q is distributed normally with mean μ and variance σ^2.

$$Q \sim U(a, b) \tag{D.1.2}$$

The random variable Q has a uniform distribution between lower bound a and upper bound b.

D.1.3 Logarithms

$$ln(x) \tag{D.1.3}$$

The natural logarithm of x.

$$log_n(x) \tag{D.1.4}$$

The logarithm base n of x.

D.1.4 Vectors

Bold font is used to represent vector quantities, *eg.* $\mathbf{x}$.

$(x_1, x_2, ..., x_n)$ is used to represent a vector of size n, where x_i is the i^{th} element.

D.2 Technical Definitions

D.2.1 Mean Squared Error (MSE)

The mean squared error can be considered as the average squared error between a model and the data it attempts to model. This corresponds to the expected squared loss. If $\mathbf{y}$ is the vector of n true values and $\hat{\mathbf{y}}$ the vector of n predicted values, then it can be defined:

$$MSE = \frac{\sum_{i=1}^{n}(\hat{y_i} - y_i)^2}{n} \tag{D.2.1}$$

D.2.2 Mean Absolute Error (MAE)

The mean absolute error can be considered as the average absolute error between a model and the data it attempts to model. Using the $\mathbf{y}$ and $\hat{\mathbf{y}}$ from D.2.1, the MAE can be defined:

$$MAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n} \tag{D.2.2}$$

D.2.3 R^2 Goodness of Fit

The R^2 goodness of fit can be considered a measurement of how well a model replicates the data it attempts to model. Using the same $\mathbf{y}$ and $\hat{\mathbf{y}}$ as previous and defining $\bar{\mathbf{y}}$ to be the mean of the data:

$$\bar{\mathbf{y}} = \frac{\sum_{i=1}^{n} y_i}{n} \tag{D.2.3}$$

then defining the *total sum of squares* as:

$$SS_{tot} = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{D.2.4}$$

Defining the *sum of squares of residuals* as:

$$SS_{res} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{D.2.5}$$

Then we can define the R^2 Goodness of Fit as:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{D.2.6}$$

D.2.4 Pearson Product-Moment Correlation Coefficient

The Pearson Product-Moment Correlation Coefficient ($PPMCC$) is a measure of the linear correlation between two variables. It takes values in the range -1 to 1 inclusive, where -1 indicates a total negative correlation, 1 a total positive correlation, and 0 represents no correlation whatsoever. If we define $\mathbf{x}$ to be a vector of n values, and $\mathbf{y}$ to be a vector of n values, then the formula for the the $PPMCC$ between $\mathbf{x}$ and $\mathbf{y}$ is as follows:

$$r = r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{D.2.7}$$