Individual-Based Modelling of Ecological Systems and Social Aggregations

by

Moshi Arthur Charnell

BSc, University of Victoria, 1998
MSc, University of Northern British Columbia, 2001

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

Masters of Science

in the Department of Mathematics and Statistics

University of Victoria

Individual-Based Modelling of Ecological Systems and Social Aggregations

by

Moshi Arthur Charnell

BSc, University of Victoria, 1998
MSc, University of Northern British Columbia, 2001

Supervisory Committee

Dr. R. Edwards, Co-Supervisor (Department of Mathematics and Statistics)

Dr. W. Reed, Co-Supervisor (Department of Mathematics and Statistics)

Dr. L. Cowen, Member (Department of Mathematics and Statistics)

Supervisory Committee

Dr. R. Edwards, Co-Supervisor (Department of Mathematics and Statistics)

Dr. W. Reed, Co-Supervisor (Department of Mathematics and Statistics)

Dr. L. Cowen, Member (Department of Mathematics and Statistics)

Abstract

This thesis gives a general model formulation of discrete time individual-based models and two specific individual-based models for gregarious behaviour. A spatially explicit individual-based model for a predator, prey and plant ecosystem is the first specific model considered. The movement of the individuals is solely based on nearest-neighbour attraction/repulsion and reproduction is asexual. The second specific model considered is a spatially explicit individual-based model for schooling behaviour. The movement of the individuals is based on the physics of fluids and their reactions (repel/comfort/attract) are solely based on directional light intensities.

In the ecosystem model emergent spatial organization of the individuals into clusters or groups is present even though all the individuals (predators and prey) are intra-specifically repelled by each other. The model ecosystem was used to determine whether an intra-specific attraction among the prey could increase their individual fitness. The attraction mechanism considered is such that when a prey is not affected by a predator or a plant then this prey is attracted to its nearest-neighbour prey. Under the assumptions of the parametrized model ecosystem, this mechanism seems highly selective on the individual level.

The schooling model assumes that the individuals have absolutely no spatial reasoning and cannot discern other individuals as individuals. In particular, the individuals school without the capacity to match the velocities or orientations of other individuals. Individuals have the ability to perceive their environment using directional photo-sensors and adjust the direction of their acceleration in response to the values of these photo-sensors. This result should fundamentally change the way social aggregations are modeled.

The main objective of this thesis is to formalize mathematically individual-based models with the intention that they become more prevalent in the scientific inquiry into assessing evolutionary aspects of social behaviour.

Table of Contents

List of Tables

List of Figures

List of Symbols

$\mathscr{M}(t)$	model ecosystem at time t.
K	number of populations in the model ecosystem.
$\mathbb{M}^k(t)$	population k at time t.
$\mathrm{I}^k(t)$	size of population k at time t.
$\mathrm{M}_i^k(t)$	individual i in population k at time t.
J^k	number of attributes for individuals in population k.
$\mathrm{M}_{ij}^k(t)$	attribute j of individual i in population k at time t.
$\mathrm{N}_i^{kl}(t)$	nearest-neighbour in population l of individual i in population k at time t.
$\mathbf{p}_i^k(t)$	position of individual i in population k at time t.
$\mathbf{v}_i^k(t)$	velocity of individual i in population k at time t.
$\mathbf{o}_i^k(t)$	orientation vector of individual i in population k at time t.
$\theta_i^k(t)$	orientation angle of individual i in population k at time t.
$\mathbf{g}_i^k(t)$	acceleration of individual i in population k at time t.
$\mathbf{h}_i^k(t)$	acceleration direction of individual i in population k at time t.
$\mathbf{e}_i^k(t)$	visual perception of individual i in population k at time t.
$\mathrm{a}_i^k(t)$	age of individual i in population k at time t.
$\mathrm{m}_i^k(t)$	mass of individual i in population k at time t.
$\mathrm{s}_i^k(t)$	stage of individual i in population k at time t.
$\mathrm{r}_i^k(t)$	radius of individual i in population k at time t.

$\mathrm{d}_{ef}^{kl}(t)$	distance from individual e in population k to individual f in population l at time t.
$\mathrm{s}_i^{kl}(t)$	distance from individual i in population k to its nearest-neighbour in population l at time t.
$\mathbf{b}_i^{kl}(t)$	bearing from individual i in population k to its nearest-neighbour in population l at time t.
$\mathrm{q}_i^k(t)$	number of individuals feeding on individual i in population k at time t.
$\mathscr{R}$	repulsion region of an individual.
$\mathscr{C}$	comfort region of an individual.
$\mathscr{A}$	attraction region of an individual.
δ	maximum radius of a plant.
ς	radial number of plants.
σ^{kl}	distance within which an individual in population k reacts to its nearest-neighbour in population l.
τ^k	maximum age of an individual in population k.
α^k	speed of an individual in population k.
β^k	minimum weight at which an individual in population k can reproduce.
ω^k	maximum weight of an individual in population k.
υ^k	minimum weight of an individual in population k.
ρ^k	density of an individual in population k.
γ^k	consumption rate of an individual in population k.
ϵ	radial growth rate of a plant.
μ^k	acceleration of an individual per unit surface area in population k.
η	viscosity of the fluid.
λ	attenuation of light in the fluid.

Acknowledgements

Throughout the development of my thesis there have been many persons whose participation were vital. These persons include my fellow students, University of Victoria support staff, my colleagues, and my family. With support from these people the conception, development, and completion of this thesis were possible.

I would like to thank my supervising committee for treating me like a colleague. Rod Edwards, Bill Reed and Laura Cowen are good people and deserve my thanks. I would also like to thank my external examiner, John Dower, for his valuable comments on my thesis. His interest in my thesis topic made the research more worthy.

Finally, I would like to thank my dear friends Laura Shepherd, Nick Henderson, Steve Lowdon, Gary Miller and Angus Argyle for their personal support. Sometimes the only thing that can clear your mind is a casual conversation with a close personal friend.

Chapter 1

Introduction

The main objective of this thesis is to detail a mathematical framework that can assess the evolution of spatial organization and social behaviour of a mobile species within an ecological system. A general model framework as a nonlinear map (chapter 2) is presented along with two specific individual-based models. The first model describes a tritrophic ecosystem (chapter 3) and the second model describes the behaviour of schooling (chapter 4). The general model framework and the tritrophic ecosystem model have been published in Charnell (2008a). Each of the specific models is unambiguous and mathematically formulated, implemented as computer code, and visually inspected using animations. Computer code and animations for the specific models can be viewed through the website: Charnell (2008b). The tritrophic model is parameter-optimized for model persistence and plausibility, and used to elicit an evolutionary mechanism that leads to gregarious behaviour in a prey species. The schooling model describes a possible set of behavioral characteristics of an individual that leads to schooling. These models are strictly deterministic and discrete in time where relationships between individuals are based on nearest neighbours for the tritrophic model and visual perception for the schooling model. The tritrophic model has the novel aspect that the primary resource is confined to an 'oasis' and the schooling model has the novel aspect that it does not include the alignment or velocity matching behaviour.

An individual-based model is a system composed of interacting individuals and the system evolves through changes in the properties of the individuals and the relationships between them. The individual-based modelling approach is a reductionist (Łomnicki 1992) or bottom-up (Grimm 1999) approach which starts with the parts (individuals) of the system (population, ecosystem) and then tries to understand how the system's properties emerge from the interactions between the parts (Grimm 1999). Individual-based models are considered more intuitive to an ecologist or biologist, conceptually simpler, and require fewer unrealistic assumptions than population-based models (Huston et al. 1988, Grimm et al. 1999, Grimm 1999, Schmitz 2000, Booth 1997, Kostova et al. 2004, Beecham and Farnsworth 1998, Breckling et al. 2006, Railsback 2001).

Many reviews of the individual-based modelling approach have appeared (Huston et al. 1988, Breckling et al. 2005, 2006, Dumont and Hill 2004, Grimm 1999, Berec 2002, Łomnicki 1992, Uchmański and Grimm 1996). A critique of ecological modelling gave the conclusion that individual-based models will induce a paradigm shift in ecology towards an 'ecology of individuals' (Uchmański and Grimm 1996). Individual-based models are considered to be within the void between real biological systems and population-based models (Wilson 1998). It has been proposed that this transition is continuous (Bolker et al. 1997) and individual-based models fit within this continuum, but also, individual-based modelling is fundamentally different (Uchmański and Grimm 1996) or incompatible (Fahse et al. 1998) with population-based models. Regardless of how individual-based models sit within the scope of ecological modelling, they are generally more realistic than population-based models.

To comprehend the mechanisms that change a population, an ecologist will study mechanisms that operate on individuals (Krebs 1985) and, moreover, to ignore knowledge of individuals and their interrelations is unreasonable (Łomnicki 1999). Modelling interacting populations as collections of individuals, rather than continuous densities, alters specific and general theoretical predictions of population dynamics (Wilson 1998). Ecological models that combine many individuals and assume that they can be described by a single variable violate the biological principle that individuals are different. As well, these models may also assume that the effect of one individual on other individuals is equal and this violates the biological principle that interactions are inherently local (Huston et al. 1988). Incorporating the principles and tenets of biology and ecology in a modelling approach gives credibility to the resulting model. Furthermore, neglecting the individual's perspective in a modelling exercise that describes population level phenomena counters the scientific process of inquiry into animal behaviour.

Similarly, the basic unit to assess a prey species should be a tritrophic food chain. Most organisms have evolved and function within a multi-trophic environment and many aspects of their ecology and behaviour are governed by interaction with individuals from other trophic levels (Poppy 1997). Other researchers have considered the tritrophic food chain where the producers are plants; the primary consumers are prey (herbivores); the secondary consumers are predators (carnivores); and each of the three trophic-levels are composed of individuals that age, consume, grow, reproduce and die (Booth 1997, Schmitz 2000). Prey compete with one another for food while avoiding predation and thus assessing the social behaviour of a prey species can only be accomplished within at least a tritrophic food chain.

The theory of evolution is based on the struggle for life and the survival of the fittest and this theory is inseparable from the discipline of ecology. The problem of animal aggregation is central to both ecological and evolutionary theory (Flierl et al. 1999) and social organization is one of the fundamental aspects of animal behaviour (Mogilner et al. 2003). Gregarious behaviour occurs in animals across a wide variety of taxa and this implies that this behaviour has evolved independently several times

(Reluga and Vicido 2005). This further implies that the evolutionary mechanisms that select for this behaviour are generic. The two main explanations for the evolution of gregarious behaviour in a prey species are to increase resource acquisition and to reduce predation risk (Morton et al. 1994, Flierl et al. 1999, Gueron et al. 1996) with the drawback of increased competition and disease transmission (Gueron et al. 1996). Individual-based modelling has the ability to model competition processes (Beecham and Farnsworth 1998) and select for movement and dispersal behaviour (Schofield et al. 2005). Individual-based models can assess evolutionary processes that led to gregarious behaviour within an ecological system.

Model ecosystems should be as simple or primitive as possible (Conrad and Pattee 1970) in order to capture the essence of ecosystems and removing unnecessary complexity would be extremely beneficial for the scientific value of individual-based models (Grimm et al. 1999). Simplistic models attempt to capture the barest essentials of a process (Flierl et al. 1999) and can be used as baseline models where increasing complexity can be incorporated (Ovaskainen and Hanski 2004). Individual-based models should be ‘strategic’ as apposed to ‘tactical’ (Grimm 1999). A strategic model is abstract and accounts for features that are generic; whereas, tactical models address specific real world situations (Murdoch et al. 1992). The model systems presented in this thesis are simple and strategic, yet comprehensive, and as such can be used to explore fundamental questions in theoretical biology.

Individual-based models have many desirable characteristics, yet there is no standard protocol for describing them (Grimm et al. 2006). Individual-based models have lacked a mathematical formalism that has been well integrated into population-based models (Grimm 1999, Ginot et al. 2002, Kostova et al. 2004, Souissi et al. 2005, Grimm et al. 2006). The relationships that individual-based models have with population-based models will become clear only when individual-based models are properly defined (Uchmański and Grimm 1996) and using the language of mathematics to describe individual-based models will allow them to be easily reproduced and increase the success or utility of these models (Grimm et al. 2006).

There is the claim that mathematical formalization of an individual-based model is not possible or at least too difficult and the only complete description of any individual-based model is its computer code (Berec 2002, Lorek and Sonnenschein 1999, Łomnicki 1999, Dumont and Hill 2004). Furthermore, 'incomplete' descriptions can be expressed using a systemic language or pseudo-code (Breckling et al. 2006, Ferreira 1995). It has been stated that the development of individual-based models should start with computer modelling (Łomnicki 1999, Minar et al. 1996) and that these computer programs must be commercially supported (Grimm et al. 1999) or use dedicated and open-source platforms (Souissi et al. 2005). Simulation software packages for individual-based models have been developed for modelling components of an ecological system with the intention that these programs would be used by non-programmers (Beecham and Farnsworth 1998, Booth 1997, Ginot et al. 2002, Kreft et al. 1998, Lorek and Sonnenschein 1999, Minar et al. 1996). Individual-based models are generally coded for computer execution and the code should be open-source with the programs that execute the code freely available, but computer code is not as concise as mathematical formalization.

Kostova et al. (2004) presented a mathematical description of an individual-based ecological model in terms of a discrete-time multidimensional nonlinear map that maps matrices to matrices, yet the presentation did not have the elegance normally attributed to mathematical formalism. The models presented in this paper are mathematically formalized as discrete-time nonlinear maps and use set theory, analytic geometry, and difference equations to describe the system's evolution through time. The set notation describes the changes in populations as a result of births and deaths, the analytic geometry organizes spatial relationships, and the difference equations describe the changes in individuals as a result of movement, growth and aging. These mathematical languages give a concise framework in which to express unambiguous spatially explicit individual-based models of multi-trophic ecosystems.

Although a mathematical description of an individual-based model is important for communication and understanding, these models are generally not amenable to analysis and computer simulations are used provide evidence to support hypotheses. Writing computer code to execute an individual-based model is not a simple process and in itself has a major part in the development of these models. The models presented in this thesis were coded using three different computer languages and if all components for the presentation of this thesis were included, then coding would account for more than half of the development time. The computer code presented in Appendices A and B are working programs that execute the models that are mathematically formalized in this thesis and as such are examples that show that these models can be coded for computers.

In an individual-based model, persistence is an emergent property that can be used to measure the fitness of a population (Grimm et al. 1999, Booth 1997, Kostova et al. 2004). The persistence of the model tritrophic ecosystem and its generation of phenomena that appear natural were the measures of success for the model parameters. Nature would tend to select the parameter configurations that produced ecosystems that persisted longer. Parameter values were adjusted by viewing the model output and experimenting with parameter values with the intention to increase in the system's persistence and that the behaviour of the individuals were visually realistic. "Natural" group formation was obtained within a system that persisted for many thousands of prey generations and this system was built from assumptions that are well established. The parameterized model tritrophic ecosystem presented is a plausible representation of general characteristics of a real ecosystem.

In the tritrophic ecological model presented in this thesis, the characteristics of the individuals (plants, prey, and predators) change in relation to their current states and the states of their nearest-neighbours. All individuals age and die as a result of aging or consumption. Plants consume an unlimited resource and grow radially at a constant rate. Prey acquire resources from feeding on plants and predators feed on prey. When individuals acquire sufficient resources they asexually reproduce by seeding or division (Booth 1997, Schmitz 2000).

The predators and prey within this model ecosystem have limited vision, are memoryless, and react only to their nearest neighbours. Reacting to a nearest neighbour is a primitive response that is simple and plausible, in that it could probably evolve (Morton et al. 1994). Predators chase and consume the nearest prey, and a prey will flee from the nearest predator or move towards and consume the nearest plant. Predators and prey are both repelled by their nearest neighbours within their own species, thereby attempting to maintain a minimum distance between themselves and other individuals (Couzin et al. 2002, Beecham and Farnsworth 1998, Flierl et al. 1999, Gueron et al. 1996, James et al. 2004, Kreft et al. 1998, Romey 1996, Viscido et al. 2005). This intra-specific repulsion is essential since individuals should not intra-specifically occupy the same space, as well as accounting for intra-specific competition.

Plants are confined to a bounded circular region, the oasis. Within the oasis, the plants are arranged hexagonally because of the superior properties of rotational symmetry to a rectangular pattern while maintaining a highly structured environment (Beecham and Farnsworth 1998). Boundary conditions for most ecological models are normally and explicitly either reflecting, periodic or absorbing (Wilson 1996). A novel aspect to this model is that it is unbounded. Prey feed and reproduce within the oasis and prey can be pushed off the oasis by predators. Predators chase and feed upon the prey within the oasis and can be lead out of the oasis by prey. Since individuals have limited reaction distances, it is possible that these individuals become 'lost', not reacting to other individuals, and eventually die of old age. The implicit absorbing boundaries depend on the positions of all individuals within the system, but are generally the reaction distances away from the boundary of the oasis for the predators and prey. Outside these circles, implicit absorbing boundaries, the prey and predators are lost.

The modelling assumptions allow the situation that a prey is lost even though there may be another prey within its limited vision. This situation can be used to assess the fitness of gregarious behaviour in a prey species. One of the main emergent properties of the model is that individuals form groups, even though they are not intra-specifically attracted to each other. That is, group formation is not dependent on intra-specific attraction within the null model. The benefit of attraction is that prey which are lost can be led back to a plant by prey which are not lost. It is obvious that there is an increase in the reproductive success of a prey having an intra-specific attraction that engages when a prey is not reacting to a predator and cannot see a plant as opposed to having no intra-specific attraction. The ecosystem model presented in this thesis seems to be as simple as possible and still able to provide evidence for this evolutionary spatial mechanism.

Intra-specific attraction precludes complex group formations that exist without external stimulus. A school is such a group formation that is defined by the high degree of polarization within the group members. The tritrophic model is reduced to a single species and the behaviour of this species is extended to include an intra-specific comfort. The resulting schooling model also includes more complex visual and physical components that seem important for the resulting school formations. Schooling is a visually complex social aggregation that has been studied using a number of different methods and the model presented in this thesis is fundamentally different.

Schooling models have been developed using many approaches, but the most successful models have been individual-based where local interactions lead to emergent global schooling structures. Many reviews of modelling schooling behaviour have appeared (Viscido et al. 2007, 2004, Couzin and Krause 2003, Mirabet et al. 2007, Adioui et al. 2003) and it seems that all individual-based models have individuals performing complex spatial calculations. These calculations include determining the direction to and distance between individuals, estimating the velocity and orientation of individuals, and counting the number of individuals that satisfy specific spatial conditions. In particular, it has been concluded that without an individual's ability to match the orientations of other individuals, schooling behaviour will not result from the interaction of the individuals (Couzin and Krause 2003, Viscido et al. 2007). The schooling model assumes that the individuals are memoryless, have no external spatial reasoning and cannot discern other individuals as individuals.

It would seem that other aggregation models are developed from the human perspective. That is, as humans how would we create aggregations and the complex spatial patterns that are apparent in other species. This human approach seems to start with the pattern and then the model is developed with a predisposition to create that pattern. For example, if the pattern is parallel orientations, then an orientation matching component in a model will produce this pattern. This approach would be acceptable if the modelling exercise was the production of the pattern. However, an aggregation should emerge from a model that is developed from the perspective of the individual and this individual's interactions should be local and physically plausible. Furthermore, the individuals within a model should not have the intention to produce patterns.

Individuals in the schooling model have the ability to perceive their environment using directional photo-sensors and adjust the direction of their acceleration in response to the values (shades of gray) of these photo-sensors. The responses are attraction, repulsion, and comfort, and depend on the direction of the photo-sensor. Most of the model formulation is developing the functions that determine the values of the photo-sensors and the functions that move individuals through a fluid under changing acceleration. The local environment is presented to the individuals and their reactions lead to schooling.

The photo-sensors are the individual's surrogate for distance and direction calculations and account for an individual obstructing the view of another individual. A higher value of an individuals photo-sensor indicates closer neighbours and repulsion occurs against a direction of a photo-sensor with a high value. Similarly, attraction occurs towards a direction of a photo-sensor with a low value. Between high and low values, medium values of the photo-sensors the reaction is a positive enforcement of current behaviour, comfort. The conditions for high, medium and low values are directionally dependent and oriented with the movement of an individual. Under certain conditions, it was found that these movement rules produced the group dynamic of schooling.

A mathematical framework is presented that clearly and unambiguously describes a possible circumstance that leads to primitive gregarious behaviour and a possible gregarious behaviour that leads to schooling. This framework along with the functional relationships described in the particular individual-based models should provide the structure to assess a wide variety of sociobiological phenomena.

Chapter 2

General Model

2.1 Introduction

The objective for this chapter is to provide a mathematical framework for discrete time individual-based models. A protocol for describing individual-based models should start with a general mathematical framework and then descriptions of the particular functional relationships between model variables should be expressed unambiguously and concisely. Mathematical formalism has been well integrated into population-based models and using the language of mathematics to describe individual-based models will certainly increase their utility and provide the structure for comparative analysis between differing model assumptions.

The specific models presented in this thesis are mathematically formalized as discrete-time nonlinear maps and use set theory, analytic geometry, and difference equations to progress the models through time. The set notation describes the changes in populations as a result of births and deaths, the analytic geometry describes spatial relationships between individuals, and the difference equations describe the changes in individuals as a result of biological and geographical situations. These three mathematical languages give a concise framework in which to express an unambiguous spatially explicit individual-based model.

2.2 The General Model

The general model framework is defined as a nonlinear map which progresses a collection of sets of vectors discretely through time. The general model is defined as a nonlinear map, Φ, which transforms a collection $\mathscr{M}(t)$ of K sets $\mathbb{M}^k(t)$, into the collection $\mathscr{M}(t+\Delta t)$ of K sets $\mathbb{M}^k(t+\Delta t)$ where $k \in \{1, 2, \ldots, \mathrm{K}\}$ and $t \in \{n\Delta t \mid n \in \mathbb{Z}^+\}$. Thus,

$$\Phi : \mathscr{M}(t) \mapsto \mathscr{M}(t+\Delta t);\ t \in \{n\Delta t \mid n \in \mathbb{Z}^+\}$$

where

$$\mathscr{M}(t) = \{\mathbb{M}^1(t), \mathbb{M}^2(t), \ldots, \mathbb{M}^{\mathrm{K}}(t)\}$$

is the model ecosystem at time t composed of K populations.

Each of the sets $\mathbb{M}^k(t)$ has $\mathrm{I}^k(t)$ vector elements of dimension J^k. The population $\mathbb{M}^k(t)$ at time t has $\mathrm{I}^k(t)$ individuals $\mathrm{M}_i^k(t), i \in \{1, 2, \ldots, \mathrm{I}^k(t)\}$ that have J^k attributes values $\mathrm{M}_{ij}^k(t), j \in \{1, \ldots, \mathrm{J}^k\}$. Thus,

$$\mathbb{M}^k(t) = \{\mathrm{M}_1^k(t), \mathrm{M}_2^k(t), \ldots, \mathrm{M}_{\mathrm{I}^k(t)}^k(t)\}$$

is population k composed of $\mathrm{I}^k(t)$ individuals at time t, and

$$\mathrm{M}_i^k(t) = (\mathrm{M}_{i1}^k(t), \mathrm{M}_{i2}^k(t), \ldots, \mathrm{M}_{i\mathrm{J}^k}^k(t))$$

is individual i with J^k attributes from population k at time t. $\mathrm{M}_{ij}^k(t)$ is the value of attribute j for individual i in population k at time t.

Chapter 3

Tritrophic Ecosystem

3.1 Introduction

The main objective of this chapter is to detail a mathematical framework that can assess the evolution of spatial organization and social behaviour of a mobile prey species within an ecological system. An unambiguous mathematically formulated model is presented, implemented as computer code, parameterized for model persistence and plausibility, and used to elicit an evolutionary mechanism that leads to gregarious behaviour in a prey species. The model ecosystem is a strictly deterministic, discrete time, individual-based model of a tritrophic food chain where relationships between individuals are solely based on nearest neighbours.

The characteristics of the individuals (plants, prey, and predators) change in relation to their current states and the states of their nearest-neighbours. When individuals acquire sufficient resources they asexually reproduce by seeding or division. The predators and prey within this model ecosystem have limited vision and are memoryless. Predators chase and consume the nearest prey, and a prey will flee from the nearest predator or move towards and consume the nearest plant. Predators and prey are both repelled by their nearest neighbours within their own species.

Plants are confined to a bounded circular region, the oasis, and arranged hexagonally within an unbounded two dimensional plane. Prey feed and reproduce within the oasis and prey can be pushed off the oasis by predators. Predators chase and feed upon the prey within the oasis and can be lead out of the oasis by prey. Since individuals have limited reaction distances, it is possible that these individuals become 'lost', not reacting to other individuals, and eventually die of old age.

The persistence of the model ecosystem and its generation of phenomena that appear 'natural' were the measures of success for the model parameters. Appearing 'natural' is that a system does not appear to contradict physical relationships. Parameter values were adjusted by viewing the model output and experimenting with parameter values with the intention to increase in the system's persistence and that the behaviour of the individuals were visually realistic. 'Natural' group formation was obtained within a system that persisted for many thousands of prey generations.

The modelling assumptions allow for the situation that a prey is lost even though there may be another prey within its limited vision. This situation can be used to assess the fitness of gregarious behaviour in a prey species. The benefit of attraction is that prey which are lost can be led back to a plant by prey which are not lost. The ecosystem model seems to be as simple as possible and still is able to provide evidence for this evolutionary spatial mechanism.

3.2 Tritrophic Model

The general model framework is defined as a nonlinear map (chapter 2) which progresses a collection of sets of vectors discretely through time. The specific model ecosystem is an individual-based tritrophic food chain (plants, prey, and predators) that is spatially-explicit in two continuous dimensions and strictly deterministic. Individuals are represented as dense circular disks where the centres of the plants and their seeds are at discrete locations in a bounded circular area, the oasis, but the predators and prey move in response to nearest-neighbours within a continuous unbounded environment.

The interactions and interrelationships between individuals are: plants produce seeds and cannot seed locations that have already been seeded or occupied by plants, seeds produce plants, prey are attracted to and consume plants, prey reproduce and are repelled by other prey, prey are repelled and consumed by predators, predators are attracted to and consume prey, and predators reproduce and are repelled by other predators. The model is initialized and all individuals will age, die, and have the potential to acquire resources and reproduce.

Plants and seeds have very different roles in an ecosystem and as such are different stages in a plant species' life history. In the model, plants grow radially from spatially discrete iso-points and are bounded by non-overlapping equivalent hexagonal regions. Their radial growth rate is constant and there are no environmental limitations, besides area, that restrict growth. Growth also adds mass to a plant and a plant has a maximum mass. Reproduction of plants is considered asexually as the seeding of unoccupied adjacent iso-points (or hexagonal regions). A plant will seed other regions when it has reached a certain mass threshold and does not lose resources as a result of reproduction. When seeds occupy a region, they expand until they have reached a certain mass and then sprout into plants. The vegetative mass of a plant is the potential resource for the prey.

Predators and prey have non-directional, limited reaction distances. Prey react to the vegetative stage of the plant and are unaware of seeds and predators do not react directly to plants. During consumption, mass is transferred from plant to prey or from prey to predator. When predators or prey have sufficient mass they reproduce by division. Each offspring receives half of the mass of its parent. Mass is the commodity of the system and this commodity is lost when an individual dies of old age.

3.2.1 Attribute and Species Definitions

In the particular model, plants, prey and predators are in populations $k = 1$, $k = 2$ and $k = 3$, respectively. Attributes that all the individuals possess are: position in space, mass and age. Predators and prey have an orientation and plants have an indicator of life stage (seed or vegetative). The position of an individual is a continuous two-dimensional vector, mass is a positive number, orientation is a continuous two-dimensional unit vector, age is a positive number, and the stage is a one-dimensional natural number.

Let $\mathbf{p}_i^k(t) \in \mathbb{R}^2$, $\mathbf{o}_i^k(t) \in \mathbb{R}^2$ with $\left\|\mathbf{o}_i^k(t)\right\| = 1$, $\mathrm{m}_i^k(t) \in \mathbb{R}^+$, $\mathrm{a}_i^k(t) \in \mathbb{R}^+$ and $\mathrm{s}_i^1(t) \in \mathbb{N}$ be the position, orientation, mass, age, and stage of individual i in population k at time t, respectively. Then, plant i at time t is

$$\mathrm{M}_i^1(t) = \left(\mathbf{p}_i^1(t), \mathrm{m}_i^1(t), \mathrm{a}_i^1(t), \mathrm{s}_i^1(t)\right),$$

prey i at time t is

$$\mathrm{M}_i^2(t) = \left(\mathbf{p}_i^2(t), \mathbf{o}_i^2(t), \mathrm{m}_i^2(t), \mathrm{a}_i^2(t)\right),$$

and predator i at time t is

$$\mathrm{M}_i^3(t) = \left(\mathbf{p}_i^3(t), \mathbf{o}_i^3(t), \mathrm{m}_i^3(t), \mathrm{a}_i^3(t)\right).$$

3.2.2 Oasis, Spatial Occupancy, Distance Between Individuals, and Nearest-Neighbours

The oasis is a composition of hexagonal regions whose centres (iso-points) are the potential positions of the plant centres. The oasis of plants is centred at the origin, $(0, 0)$, and defined by half the distance between plant centres (maximum radius of a plant), $\delta > 0$, and the radial number of plants, $\varsigma \geq 0$. The radius of the circle that bounds the potential locations of the plant centres is $2\varsigma\delta$ and the set of locations is

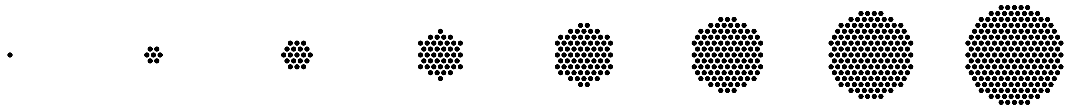

Figure 3.1: Point configurations for the potential locations of plants using radial number of plants, ς, having values 0 through 7 from left to right, respectively.

defined as

$$\mathbb{I} \;=\; \left\{p \mid p = \delta\left(a \pm 2b, \pm\sqrt{3}a\right); a, b \in \{0, 1, 2, \ldots\};\; \|p\| \leq 2\varsigma\delta\right\}. \tag{3.1}$$

The magnitude of the set $\mathbb{I}$ is at least 1, when $\varsigma = 0$, and is bounded above by limitations of a computer's computational abilities. Figure 3.1 displays the point configurations for values of $\varsigma = 0, 1, \ldots, 7$. These positions can be vacant or occupied by a plant.

Individuals are dense circular disks within the planar surface. Plants, prey and predators have differing densities, yet are constant within each species. Spatial extent of the individual i in population k at time t is a function of an individual's mass, $\mathrm{m}_i^k(t)$, and the population density parameter ρ^k. The radius of individual i in population k at time t, $\mathrm{r}_i^k(t)$, is defined through the relationship

$$\mathrm{m}_i^k(t) = \rho^k \pi \left(\mathrm{r}_i^k(t)\right)^2, \tag{3.2}$$

where π is the ratio of the diameter of a circle to its circumference.

The distance between two individuals is a function of the centres and radii of those individuals. The distance between individual e from population k and individual f from population l, $\mathrm{M}_e^k(t)$ and $\mathrm{M}_f^l(t)$, with respective radii, $\mathrm{r}_e^k(t)$ and $\mathrm{r}_f^l(t)$, at time t is:

$$\mathrm{d}_{ef}^{kl}(t) = \left\|\mathbf{p}_e^k(t) - \mathbf{p}_f^l(t)\right\| - \left(\mathrm{r}_e^k(t) + \mathrm{r}_f^l(t)\right).$$

Note that this distance function can be negative. A positive value indicates that the circles do not intersect (do not overlap), a value of zero indicates that circles intersect at a point (touch) and a negative value indicates that the circles intersect at many points (overlap).

The model uses nearest neighbours to define movement, either towards or away from a nearest-neighbour; and/or a change in an individual's attribute (example: feeding on a nearest-neighbour). The nearest neighbour from population l, its distance to, and bearing from individual e in population k, respectively, are:

$$\mathrm{N}_e^{kl}(t) = \mathrm{M}_f^l(t);$$

$$\mathrm{s}_e^{kl}(t) = \mathrm{d}_{ef}^{kl}(t); \text{ and}$$

$$\mathbf{b}_e^{kl}(t) = \frac{\mathbf{p}_f^l(t) - \mathbf{p}_e^k(t)}{\left\|\mathbf{p}_f^l(t) - \mathbf{p}_e^k(t)\right\|}, \tag{3.3}$$

where f is the index of an individual in population l defined through the following function:

$$\mathrm{d}_{ef}^{kl}(t) = \begin{cases} \min_i \left(\mathrm{d}_{ei}^{kl}(t)\right) & if \quad k \neq l \\ \\ \min_{i \neq e} \left(\mathrm{d}_{ei}^{kl}(t)\right) & if \quad k = l \end{cases}.$$

3.2.3 Reaction Distances

Predators and prey only react to their nearest neighbours. The behaviour of a prey is determined by the prey's nearest plant, nearest predator and nearest prey, and the behaviour of a predator is determined by the predator's nearest prey and nearest predator.

The reaction distances are constant within a population. Let σ^{kk} be the distance within which an individual in population k reacts to another individual in population k. This intra-specific reaction is repelling and individual i in population k will be repelled by the nearest neighbour in population k at time t when the distance to that nearest neighbour is less than or equal to the intra-specific reaction distance. Also, let σ^{kl} be the maximum distance within which an individual in population k can react to an individual in population l. These inter-specific distances define when individual i in population k can be attracted to or repelled by the nearest neighbour in population l at time t.

3.2.4 Aging and Life Stages

All individuals age and their age can equal their maximum age, τ^k, but these individuals will die at the next time step (section 3.2.6). The age of individual i in population k at time $t + \Delta t$ is

$$\mathrm{a}_i^k(t + \Delta t) = \mathrm{a}_i^k(t) + \Delta t.$$

Only plants have different developmental stages in their lives. Plants start as seeds (stage 1), these seed grow (accumulate mass), and when these seeds have sufficient mass, υ^1, they become vegetative plants (stage 2). That is, the stage of plant i at time $t + \Delta t$ is

$$\mathrm{s}_i^1(t + \Delta t) = \begin{cases} 1 & \text{if} \quad (\mathrm{m}_i^1(t) < \upsilon^1) \wedge (\mathrm{s}_i^1(t) = 1) \\ \\ 2 & \text{if} \quad (\mathrm{m}_i^1(t) \geq \upsilon^1) \vee (\mathrm{s}_i^1(t) = 2) \end{cases} .$$

3.2.5 Resource Acquisition and Loss

Resource acquisition is the accumulation of mass by growing or feeding. Predators consume prey, prey consume vegetative plants, and plants grow from an unlimited resource but are bounded by the hexagonal regions of the oasis. A predator feeds on the nearest prey when the predator's disk intersects the prey's disk. A prey feeds on the nearest vegetative plant when the prey's disk intersects the plant's disk and the prey is not reacting to a predator. Prey and predators gain γ^2 and γ^3, units of mass for each unit of time that they are feeding upon prey and plants, respectively. Each plant or prey that is being fed upon loses γ^2 and γ^3 units of mass for each prey or predator, respectively, that is feeding upon it during each unit of time. Plants can grow and be consumed at the same time, but prey cannot consume plants when they are reacting to predators.

All individuals have a population specific maximum mass ω^k or maximum radius defined using equation 3.2. Thus, the following equation defines the mass of individual i in population k at time $t + \Delta t$:

$$\mathrm{m}_i^k(t+\Delta t) = \min\left(\left(\mathrm{m}_i^k(t) + \Delta\mathrm{m}_i^k(t)\right), \omega^k\right), \tag{3.4}$$

where $\Delta\mathrm{m}_i^k(t)$ is the change in mass of individual i in population k that occurs at time t and is defined for each population.

Plants grow radially at a constant rate, ϵ (unit distance per unit time), and are consumed by prey during their vegetative stage. The vegetative mass loss for a plant is $\gamma^2 \Delta t$ units of mass for every prey that is consuming it. The number of prey feeding on a vegetative plant is the number of prey whose circular disks intersect with the disk of the plant and are not being influenced by a predator. Thus, the number of prey feeding on plant i at time t is

$$\mathrm{q}_i^1(t) = \begin{cases} 0 & \text{if} \quad \mathrm{s}_i^1(t) = 1 \\ \\ n & \text{if} \quad \mathrm{s}_i^1(t) = 2 \end{cases},$$

where

$$n = \left| \left\{ \mathrm{M}_j^2(t) \mid \mathrm{N}_j^{21}(t) = \mathrm{M}_i^1(t) \wedge \mathrm{s}_j^{21}(t) \leq 0 \wedge \mathrm{s}_j^{23}(t) > \sigma^{23} \right\} \right|.$$

The growth of plant i at time t is the addition of an annulus of density ρ^1 where the thickness is $\epsilon\Delta t$ and inner radius is the radius of the plant at time t, $\mathrm{r}_i^1(t)$. Thus, the change in mass for plant i at time t is

$$\Delta \mathrm{m}_i^1(t) = \rho^1 \pi \left(\left(\mathrm{r}_i^1(t) + \epsilon\Delta t\right)^2 - \left(\mathrm{r}_i^1(t)\right)^2 \right) - \mathrm{q}_i^1(t)\gamma^2\Delta t.$$

When a plant is in its vegetative stage its mass can fall below υ^1, but it will die at time $t + \Delta t$ (section 3.2.6). Also, plants are bounded by the hexagonal regions of the oasis and the mass of a plant cannot exceed ω^1. Thus, the mass of plant i at time $t + \Delta t$ is calculated using equation 3.4.

A prey will gain $\gamma^2\Delta t$ units of mass (feeding) when its disk intersects the disk of a plant and it is not responding to a predator, not change mass when its disk does not intersect the disk of a plant or is fleeing from a predator, or lose $\gamma^3\Delta t$ units of mass for every predator that is consuming it. The number of predators feeding on a particular prey is the number of predators that have this prey as their nearest neighbour and whose circular disks intersect the disk of this prey. A predator's disk may intersect more than one prey's disk, but a predator will only feed on the nearest prey. Thus, the number of predators feeding on prey i at time t is

$$\mathrm{q}_i^2(t) \quad = \quad \left| \left\{ \mathrm{M}_j^3(t) \mid \mathrm{N}_j^{32}(t) = \mathrm{M}_i^2(t) \wedge \mathrm{s}_j^{32}(t) \leq 0 \right\} \right|.$$

Then the change in mass for prey i at time t is

$$\Delta \mathrm{m}_i^2(t) = \begin{cases} \gamma^2\Delta t & \text{if } (\mathrm{s}_i^{21}(t) \leq 0) \wedge (\mathrm{s}_i^{23}(t) > \sigma^{23}) \\ -\mathrm{q}_i^2(t)\gamma^3\Delta t & \text{if } \mathrm{s}_i^{23}(t) \leq 0 \\ 0 & \text{otherwise} \end{cases}.$$

The mass of a prey at time t can fall below υ^2 but that prey will die at time $t+\Delta t$

(section 3.2.6). The mass of a prey at time t is at most ω^2 and a prey will reproduce at time $t + \Delta t$ when its mass at time t is ω^2 (section 3.2.6). Therefore, the mass of prey i at time $t + \Delta t$ is calculated using equation 3.4.

A predator will gain $\gamma^3 \Delta t$ units of mass (feeding) when its disk intersects the disk of a prey and otherwise will not change mass. Then the change in mass for predator i at time t is

$$\Delta \mathrm{m}_i^3(t) = \begin{cases} \gamma^3 \Delta t & \text{if} \quad \mathrm{s}_i^{32}(t) \leq 0 \\ \\ 0 & \text{if} \quad \mathrm{s}_i^{32}(t) > 0 \end{cases} .$$

The mass of a predator is never decreasing and does not exceed ω^2, and a predator will reproduce at time $t + \Delta t$ when its mass at time t is ω^2 (section 3.2.6). Therefore, the mass of predator i at time $t + \Delta t$ is calculated using equation 3.4.

3.2.6 Reproduction and Death

An individual is removed from the model ecosystem when it dies and an individual is added when a birth occurs. All individuals die when their ages exceed a population specific maximum age. Predators only die of old age; whereas, plants and prey can also die when their mass is less than a minimum mass. Reproduction occurs when an individual has accumulated sufficient mass. Reproduction for predators and prey occurs when they have reached their maximum mass; whereas, plants can reproduce at a mass less than their maximum mass. Reproduction for prey and predators is by division and plants reproduce by seeding.

The vegetative stage of a plant dies at time $t + \Delta t$ when its mass is below the minimum mass, υ^1, or its age is at least the maximum age, τ^1, at time t. A plant can seed unoccupied adjacent iso-points contained within the set of oasis points, $\mathbb{I}$ (equation 3.1), at time $t + \Delta t$, when its mass is at least the minimum mass for reproduction, β^1, at time t.

Let $\mathbb{V}(t)$ and $\mathbb{U}(t)$ be the sets of points contained in $\mathbb{I}$ that have and do not have, respectively, plants centred upon them at time t. An element of $\mathbb{V}(t)$ is removed and added to $\mathbb{U}(t)$ when a plant dies at time $t + \Delta t$, and an element of $\mathbb{U}(t)$ is removed and added to $\mathbb{V}(t)$ when a seed is sown at time $t + \Delta t$.

Let $\mathbb{F}(t)$ be the sets of positions that have seed producing plants (fertile) at time t. Then the set of positions for fertile plants at time t is

$$\mathbb{F}(t) = \left\{ \mathbf{p}_i^1(t) \;\middle|\; \left(\mathrm{m}_i^1(t) \geq \beta^1\right) \wedge \left(\mathrm{a}_i^1(t) \leq \tau^1\right) \right\}.$$

Also, let $\mathbb{D}(t)$ and $\mathbb{S}(t)$ be the set of positions that have plants that died at time t and positions that are seeded at time t, respectively. Then the set of positions for plants that die at time $t + \Delta t$ is

$$\mathbb{D}(t + \Delta t) = \left\{ \mathbf{p}_i^1(t) \;\middle|\; \left(\mathrm{a}_i^1(t) \geq \tau^1\right) \vee \left(\mathrm{s}_i^1(t) = 2 \wedge \mathrm{m}_i^1(t) < \upsilon^1\right) \right\}.$$

The positions of plants that are seeded at time $t + \Delta t$ are elements of the unoccupied positions at time t and these positions are adjacent (distance between positions is 2δ) to fertile plants at time t. Thus, the set of positions that are seeded at time $t + \Delta t$ is

$$\mathbb{S}(t + \Delta t) = \left\{ \mathrm{U} \;\middle|\; (\mathrm{U} \in \mathbb{U}(t)) \wedge \left(\exists \mathrm{F} \in \mathbb{F}(t) \ni \left\|\mathrm{F} - \mathrm{U}\right\| = 2\delta\right) \right\}.$$

The set of unoccupied positions at time $t + \Delta t$ is the union of the unoccupied locations at time t and the positions of the plants that die at time $t + \Delta t$ with the positions that are seeded at time $t + \Delta t$ removed. Thus,

$$\mathbb{U}(t + \Delta t) = (\mathbb{U}(t) \cup \mathbb{D}(t + \Delta t)) \setminus \mathbb{S}(t + \Delta t).$$

The positions of the plants at time $t + \Delta t$ is the union of the positions of the plants at time t and the positions that are seeded at time $t + \Delta t$ with the positions of the plants that die at time $t + \Delta t$ removed. Thus,

$$\mathbb{V}(t + \Delta t) \quad = \quad (\mathbb{V}(t) \cup \mathbb{S}(t + \Delta t)) \setminus \mathbb{D}(t + \Delta t).$$

When plant i is removed at time $t + \Delta t$ then the element $\mathrm{M}_i^1(t)$ is removed from $\mathbb{M}^1(t)$. When an unoccupied position is seeded at time $t + \Delta t$ then an element is added to $\mathbb{M}^1(t)$. A plant that is seeded at time $t + \Delta t$ has a position from $\mathbb{S}(t + \Delta t)$, is one time step, Δt, old, has one unit of mass, and is in stage 1. Thus, the population of plants at time $t + \Delta t$ is

$$\mathbb{M}^1(t + \Delta t) = \left\{\mathbb{M}_i^1(t) \mid \mathbf{p}_i^1(t) \notin \mathbb{D}(t + \Delta t)\right\} \cup \left\{(\mathbf{p}, \Delta t, 1, 1) \mid \mathbf{p} \in \mathbb{S}(t + \Delta t)\right\}$$

Prey and predators reproduce and die in a similar manner. A prey dies at time $t + \Delta t$ when its mass is less than the minimum mass, v^2, or its age is at least the maximum age, τ^2, at time t. A predator dies at time $t + \Delta t$ only when its age is at least the maximum age, τ^3, at time t. Then the set of dead prey at time $t + \Delta t$ is

$$\mathbb{D}^2(t + \Delta t) = \left\{\mathrm{M}_i^2(t) \mid \left(\mathrm{a}_i^2(t) \geq \tau^2\right) \vee \left(\mathrm{m}_i^2(t) < v^2\right)\right\};$$

and the set of dead predators at time $t + \Delta t$ is

$$\mathbb{D}^3(t + \Delta t) = \left\{\mathrm{M}_i^3(t) \mid \mathrm{a}_i^3(t) \geq \tau^3\right\}.$$

Prey and predators reproduce at time $t + \Delta t$ when their masses are at least their reproductive mass or their maximum masses, $\beta^k = \omega^k$, at time t. Then the prey and predators that will reproduce (divide) at time $t + \Delta t$ are

$$\mathbb{F}^k(t + \Delta t) = \left\{\mathrm{M}_i^k(t) \mid \mathrm{m}_i^k(t) \geq \beta^k = \omega^k\right\}.$$

When a prey or predator reproduces that individual divides into two individuals that have equal mass $\beta^k/2$. One of the two offspring maintains the position, orientation, and age of the parent, and the other offspring is positioned one step behind, has the opposite orientation and an age of one time step, Δt. The offspring that maintain the ages of the parents at time $t + \Delta t$ are

$$\mathbb{C}^k(t+\Delta t) = \left\{ \left(\mathbf{p}_i^k(t), \mathbf{o}_i^k(t), \mathrm{a}_i^k(t), \beta^k/2 \right) \;\middle|\; \mathrm{M}_i^k(t) \in \mathbb{F}^k(t+\Delta t) \right\}.$$

Whereas, the offspring that are at age Δt after the division of the parents at time $t + \Delta t$ are

$$\mathbb{B}^k(t+\Delta t) = \left\{ \left(\mathbf{p}_i^k(t) - \alpha^k \mathbf{o}_i^k(t), -\mathbf{o}_i^k(t), \Delta t, \beta^k/2 \right) \;\middle|\; \mathrm{M}_i^k(t) \in \mathbb{F}^k(t+\Delta t) \right\}.$$

Then the prey and predators that are not affected by reproduction or death at time $t + \Delta t$ are

$$\mathbb{U}^k(t+\Delta t) = \mathbb{M}^k(t) \backslash \left(\mathbb{D}^k(t+\Delta t) \cup \mathbb{F}^k(t+\Delta t) \right),$$

and thus, the populations of prey and predators at time $t + \Delta t$ are

$$\mathbb{M}^k(t+\Delta t) = \mathbb{U}^k(t+\Delta t) \cup \mathbb{B}^k(t+\Delta t) \cup \mathbb{C}^k(t+\Delta t).$$

3.2.7 Movement

Movement is solely based on attraction towards or repulsion from nearest-neighbours and prey and predators move at the constant speeds α^2 and α^3, respectively. Predators and prey have reaction distances, within which they react to their nearest neighbours (section 3.2.3). Prey are stationary only if they are being consumed by at least one predator or they are consuming a plant and not being influence by other individuals. Predators are stationary only if they are consuming a prey and not influenced by other predators.

Movement of an individual is a two-step process where an individual changes orientation and then moves in the direction of the adjusted orientation. The adjusted orientation of an individual is a function of the current orientation and the bearings to the nearest neighbours (section 3.2.2). The effect of the nearest neighbours on the orientation is a linear combination of the bearings to the nearest neighbours depending on the values of the nearest-neighbour distances, $\mathrm{s}_i^{kl}(t)$ (section 3.2.2), and the reaction distances, σ^{kl} (section 3.2.3). For simplicity, the possible values for the coefficients in the linear combinations are 1, -1, and 0, for an attraction, repulsion and no effect, respectively.

The effect of the nearest neighbours on orientation $\Delta\mathbf{o}_i^k(t)$ is added to the previous orientation and normalized, so that the orientation of individual i in population k at time $t+\Delta t$ is

$$\mathbf{o}_i^k(t+\Delta t) = \frac{\mathbf{o}_i^k(t) + \Delta\mathbf{o}_i^k(t)}{\left\|\mathbf{o}_i^k(t) + \Delta\mathbf{o}_i^k(t)\right\|}. \tag{3.5}$$

A predator or prey can move or not move at time t. When an individual moves at time t, it will move in the direction, $\mathbf{o}_i^k(t+\Delta t)$, at the speed α^k. Thus, the position of predator or prey i at time t is

$$\mathbf{p}_i^k(t+\Delta t) = \mathbf{p}_i^k(t) + \Delta\mathbf{p}_i^k(t), \tag{3.6}$$

where $\Delta\mathbf{p}_i^k(t)$ is $\alpha^k\mathbf{o}_i^k(t+\Delta t)\Delta t$ when the individual is moving and 0 when it is not moving. When an individual is not moving there is not a change in position or in orientation.

A prey can feed and move in response to another prey, but does not feed when it is responding to a predator. A prey will move away from the nearest predator and nearest prey if these nearest-neighbours are within the reaction distances, and if a prey is not reacting to a predator, then the prey will move towards the nearest plant when the plant is within the plant reaction distance. If a prey is not reacting to another individual, then it moves along its previous trajectory. Thus, for prey i the change in orientation at time t, $\Delta \mathbf{o}_i^2(t)$, is computed using equation 3.7, the orientation at time $t + \Delta t$, $\mathbf{o}_i^2(t + \Delta t)$, is computed using equation 3.5, the change in position at time t, $\Delta \mathbf{p}_i^2(t)$, is computed using equation 3.8, and the position at time $t + \Delta t$, $\mathbf{p}_i^2(t + \Delta t)$, is computed using equation 3.6.

$$\Delta \mathbf{o}_i^2(t) = \begin{cases} -\mathbf{b}_i^{23}(t) - \mathbf{b}_i^{22}(t) & \text{if } (\mathrm{s}_i^{22}(t) \leq \sigma^{22}) \wedge (0 < \mathrm{s}_i^{23}(t) \leq \sigma^{23}) \\ \mathbf{b}_i^{21}(t) - \mathbf{b}_i^{22}(t) & \text{if } (\mathrm{s}_i^{22}(t) \leq \sigma^{22}) \wedge (0 < \mathrm{s}_i^{21}(t) \leq \sigma^{21}) \wedge (\mathrm{s}_i^{23}(t) > \sigma^{23}) \\ -\mathbf{b}_i^{23}(t) & \text{if } (\mathrm{s}_i^{22}(t) > \sigma^{22}) \wedge (0 < \mathrm{s}_i^{23}(t) \leq \sigma^{23}) \\ \mathbf{b}_i^{21}(t) & \text{if } (\mathrm{s}_i^{22}(t) > \sigma^{22}) \wedge (0 < \mathrm{s}_i^{21}(t) \leq \sigma^{21}) \wedge (\mathrm{s}_i^{23}(t) > \sigma^{23}) \\ -\mathbf{b}_i^{22}(t) & \text{if } (\mathrm{s}_i^{22}(t) \leq \sigma^{22}) \wedge (\mathrm{s}_i^{21}(t) > \sigma^{21}) \wedge (\mathrm{s}_i^{23}(t) > \sigma^{23}) \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

$$\Delta \mathbf{p}_i^2(t) = \begin{cases} 0 & \text{if } ((\mathrm{s}_i^{21}(t) \leq 0) \wedge (\mathrm{s}_i^{23}(t) > \sigma^{23}) \wedge (\mathrm{s}_i^{22}(t) > \sigma^{22})) \vee (\mathrm{s}_i^{23}(t) \leq 0) \\ \alpha^2 \mathbf{o}_i^2(t + \Delta t)\Delta t & \text{otherwise} \end{cases} \tag{3.8}$$

A predator is stationary only if it is feeding on a prey and is not reacting to other predators. Moving predators will chase the nearest prey and move away from the nearest predator if these nearest-neighbours are within the reaction distances; and otherwise, they continue along their previous trajectories. Thus, for predator i the change in orientation at time t, $\Delta \mathbf{o}_i^3(t)$, is computed using equation 3.9, the orientation at time $t + \Delta t$, $\mathbf{o}_i^3(t + \Delta t)$, is computed using equation 3.5, the change in position at time t, $\Delta \mathbf{p}_i^3(t)$, is computed using equation 3.10, and the position at time $t + \Delta t$, $\mathbf{p}_i^3(t + \Delta t)$, is computed using equation 3.6.

$$\Delta \mathbf{o}_i^3(t) = \begin{cases} \mathbf{b}_i^{32}(t) - \mathbf{b}_i^{33}(t) & \text{if } (\mathrm{s}_i^{33}(t) \leq \sigma^{33}) \wedge (0 < \mathrm{s}_i^{32}(t) \leq \sigma^{32}) \\ \mathbf{b}_i^{32}(t) & \text{if } (\mathrm{s}_i^{33}(t) > \sigma^{33}) \wedge (0 < \mathrm{s}_i^{32}(t) \leq \sigma^{32}) \\ -\mathbf{b}_i^{33}(t) & \text{if } (\mathrm{s}_i^{33}(t) \leq \sigma^{33}) \wedge (\mathrm{s}_i^{32}(t) > \sigma^{32}) \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

$$\Delta \mathbf{p}_i^3(t) = \begin{cases} 0 & \text{if } (\mathrm{s}_i^{33}(t) > \sigma^{33}) \wedge (\mathrm{s}_i^{32}(t) \leq 0) \\ \alpha^3 \mathbf{o}_i^3(t + \Delta t)\Delta t & \text{otherwise} \end{cases} \tag{3.10}$$

3.3 Model Implementation, Parameterization and Experimentation

The experimental objective was to create a model that could assess the intra-specific attraction among the prey that only occurs when a prey is not reacting to predators or plants. In order to accomplish this, a persistent, plausible model that did not have intra-specific attraction among the prey, a null model, had to be constructed and implemented. The model described in section 3.2 is the null model, since it contains no intra-specific attraction and the situation that a prey is not reacting to either a predator or a plant is present. The model was implemented and parameterized with a set of values that lead to persistence, and was a plausible representation of an ecosystem. The prey behaviour was then modified to incorporate the desired intra-specific attraction and the two types of prey (unmodified and modified) were allowed to compete within the ecological simulation.

The model was implemented using computer languages R and FORTRAN. R (R 2008) is a language and environment for statistical computing and graphics. FORTRAN (GCC 2008b) is a compiled imperative language used for scientific computing. Spatial organization of the individuals was assessed through animations of model output produced using the image editing software: The GIMP (GIMP 2008). R's availability, graphics capabilities, and its integration with FORTRAN make it ideal for the development of individual-based models. The computer code for this model is displayed in Appendix A and this code along with animations can be viewed at the website: Charnell (2008b).

The model depends on a set of parameters and through model development a base set was determined (table 3.1). The time step, Δt, used was 1 unit time and for most of the model development a value of 1 was used for the radial number of plants, ς (7 plants). This value was large enough to assess the global spatial patterns of the individuals and the persistence of the model, yet small enough that the simulation of the model ran quickly ($\sim$20 time steps/sec).

Initially, the distance between plants, 2δ, and the speed of the prey, α^2, were adjusted so that the prey moved about the space within a reasonable amount of time but their motion still appeared smooth. While keeping the values ς, 2δ, and α^2 constant, the initial spatial locations of the prey and predators were randomly distributed within the oasis and the other parameter values were adjusted by trial and error and then systematically using a grid-based search. Parameter values were chosen with the intention that the model ecosystem persists for many prey generations and that the population sizes were small, for computational efficiency.

The parameter space for this model is vast and as such conditions were imposed to restrict this space to a manageable set. Considerations for determining parameter values were: predators are more dense than prey which are more dense than plants, predators can accumulate more mass than plants which can accumulate more mass than prey, predators are faster than prey, predators and prey have equal radii when they are at their respective maximum masses, plants live longer than predators which live longer than prey, plants reproduce faster than prey which reproduce faster than predators, predators can see further than prey and prey react to predators and plants at an equal distance, and group formations did not contradict physical relationships.

The density and mass of individuals were chosen to account for energy stored at the trophic levels, the speeds were chosen so that predators could catch prey, reproduction and maximum age were chosen so that the lower trophic levels could recover faster from low population sizes, and predators see further than prey so that a prey can be feeding even though a predator is pursuing it. These considerations lead to a working model and certainly do not fully explore the parameter space.

The main experiment was to determine whether a simple attraction mechanism could lead to an increase in the fitness of the prey at the individual level. The base set of parameter values and initial population sizes (displayed in table 3.1, with $\varsigma = 2$ and $\mathrm{I}^1(0) = 19$) were used for assessing the fitness of this attraction mechanism for the prey. Each potential location of a plant, $\mathbb{I}$, starts with a plant at its maximum mass and its age is drawn uniformly from the interval $[1, \tau^1]$. Predators and prey are randomly distributed within the oasis with random orientations and their masses and ages are uniformly drawn from $[\omega^k/2, \omega^k]$ and $[1, \tau^k]$, respectively.

The unmodified prey were repelled by predators and other prey, and attracted to plants. The modified prey can be repelled and attracted, not at the same time, by other prey. The reaction distance that unmodified prey had with other prey, σ^{22}, was such that if the distance between a prey and its nearest-neighbour prey is within this reaction distance, then the prey is repelled by its nearest-neighbour. The modified prey can be attracted to another prey when it is within a distance σ_a^{22} and repelled when a prey is within σ_r^{22}. For a modified prey, σ^{21} and σ^{23} are the same as for the unmodified prey, the attraction distance σ_a^{22} is equal to the repulsion distance for predators, σ^{23}, and the repulsion distance σ_r^{22} is equal to the repulsion distance σ^{22} for unmodified prey.

The modified prey behave in exactly the same way as the unmodified prey when they are reacting either to plants or predators. It is when a modified prey is not reacting to a plant or a predator that it behaves differently from the unmodified prey. Thus, the change in orientation for modified prey i at time t is defined by equation 3.11.

$$\Delta\mathbf{o}_i^2(t) = \begin{cases} -\mathbf{b}_i^{23}(t) - \mathbf{b}_i^{22}(t) & \text{if } (\mathrm{s}_i^{22}(t) \leq \sigma_r^{22}) \wedge (0 < \mathrm{s}_i^{23}(t) \leq \sigma^{23}) \\ \mathbf{b}_i^{21}(t) - \mathbf{b}_i^{22}(t) & \text{if } (\mathrm{s}_i^{22}(t) \leq \sigma_r^{22}) \wedge (0 < \mathrm{s}_i^{21}(t) \leq \sigma^{21}) \wedge (\mathrm{s}_i^{23}(t) > \sigma^{23}) \\ -\mathbf{b}_i^{23}(t) & \text{if } (\mathrm{s}_i^{22}(t) > \sigma_r^{22}) \wedge (0 < \mathrm{s}_i^{23}(t) \leq \sigma^{23}) \\ \mathbf{b}_i^{21}(t) & \text{if } (\mathrm{s}_i^{22}(t) > \sigma_r^{22}) \wedge (0 < \mathrm{s}_i^{21}(t) \leq \sigma^{21}) \wedge (\mathrm{s}_i^{23}(t) > \sigma^{23}) \\ -\mathbf{b}_i^{22}(t) & \text{if } (\mathrm{s}_i^{22}(t) \leq \sigma_r^{22}) \wedge (\mathrm{s}_i^{21}(t) > \sigma^{21}) \wedge (\mathrm{s}_i^{23}(t) > \sigma^{23}) \\ \mathbf{b}_i^{22}(t) & \text{if } (\sigma_r^{22} < \mathrm{s}_i^{22}(t) \leq \sigma_a^{22}) \wedge (\mathrm{s}_i^{21}(t) > \sigma^{21}) \wedge (\mathrm{s}_i^{23}(t) > \sigma^{23}) \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

In order to determine whether the modified prey are individually more fit than the unmodified prey, ten simulations of the ecosystem starting with half of the prey modified and half unmodified were performed. The number of simulations where the modified prey completely dominate the prey population (success) out of the ten simulations performed will provide evidence as to whether the modified prey success rate is greater than 50% and therefore whether the modified prey are individually more fit than the unmodified prey.

3.4 Results

The parameterized model ecosystem described in this paper is a plausible representation of a tritrophic ecosystem that can assess attraction between individuals of a prey species. The model described in section 3.2 is fully implemented except for the condition where centres of individuals are so close to one another that a computer cannot discern the distance between them. This distance is less than the machine tolerance, and gives this distance a value of zero. If the distance between an indi-

Table 3.1: Parameter values and initial population sizes used for experimentation and visualization.

$\mathrm{I}^1(0)$	$=$	7 or 19	$\mathrm{I}^2(0)$	$=$	$10 \cdot \mathrm{I}^1(0)$	$\mathrm{I}^3(0)$	$=$	$4 \cdot \mathrm{I}^1(0)$
ρ^1	$=$	10	ρ^2	$=$	300	ρ^3	$=$	3000
ω^1	$=$	$\rho^1 \pi \delta^2$	ω^2	$=$	100	ω^3	$=$	1000
β^1	$=$	$\omega^1/3$	β^2	$=$	ω^2	β^3	$=$	ω^3
σ^{21}	$=$	5	σ^{22}	$=$	1.25	σ^{23}	$=$	5
τ^1	$=$	10000	τ^2	$=$	600	τ^3	$=$	3000
δ	$=$	5	ς	$=$	1 or 2	α^2	$=$	0.25
α^3	$=$	$1.375\alpha^2$	υ^1	$=$	$\omega^1/15$	υ^2	$=$	$\omega^2/4$
γ^2	$=$	1	γ^3	$=$	1	σ^{32}	$=$	40
σ^{33}	$=$	0	ϵ	$=$	$\delta/75$			

vidual and its nearest neighbour is zero then the bearing from the individual to this neighbour is undefined as a result of dividing by zero (equation 3.3). This situation occurs when a group of prey are pushed together by surrounding predators. The computer code was modified and when the distance between nearest-neighbours is less than the machine tolerance the individuals do not respond to their respective nearest-neighbour and continue along their previous trajectory.

The parametrization is displayed in table 3.1. All parameter values were fixed based on the conditions set in section 3.3 except for the speed of the predators (α^3) and the growth rate of the plants (ϵ). These two parameters seemed to have the most affect on the model persistence and an experiment was conducted to determine which values of these parameters lead to the highest persistence. This optimization experiment used the initialization conditions referred to in section 3.3 and varied the time until the maximum radial growth of a plant (δ/ϵ) and the relative speed of predators to the prey (α^3/α^2). A grid-based search was considered with each grid point replicated ten times and the results are displayed in figure 3.2. Although the open circle in figure 3.2 indicates the most persistent parameter values, the largest closed circle was chosen since the predator, prey and plant population sizes were much

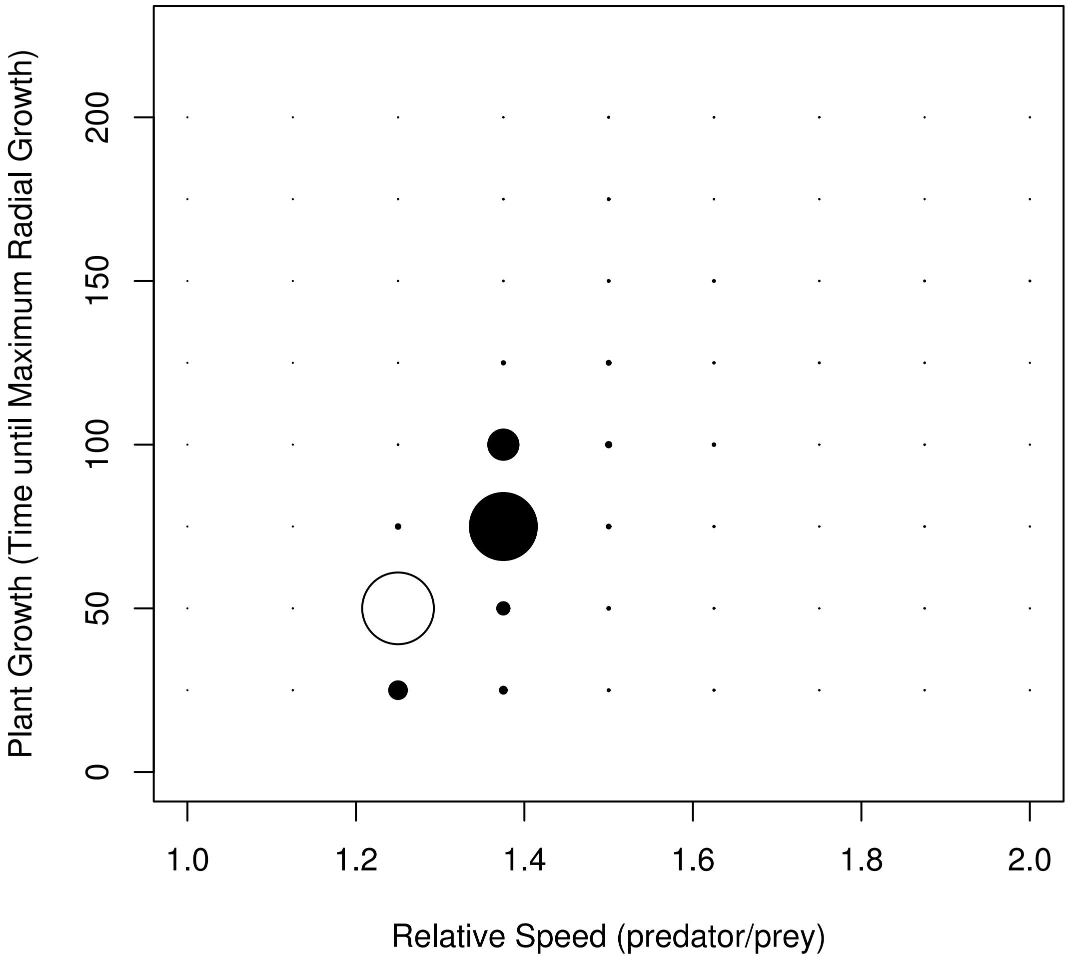

Figure 3.2: Mean persistence in relation to the time until the maximum radial growth of a plant (δ/ϵ) and the relative speed of predators to the prey (α^3/α^2). The persistence's are right censored at 50,000 iterations and the means were calculated assuming an Exponentially distributed persistence. The size of the circles are linearly proportional to the mean persistence and the open circle indicates that all persistence values exceeded 50,000 iterations.

smaller. The resulting set of parameter values were sufficient to acquire results for the experimental question concerning attraction between prey and provide a working model for future research.

The formation of prey and predators into groups is an emergent property of the

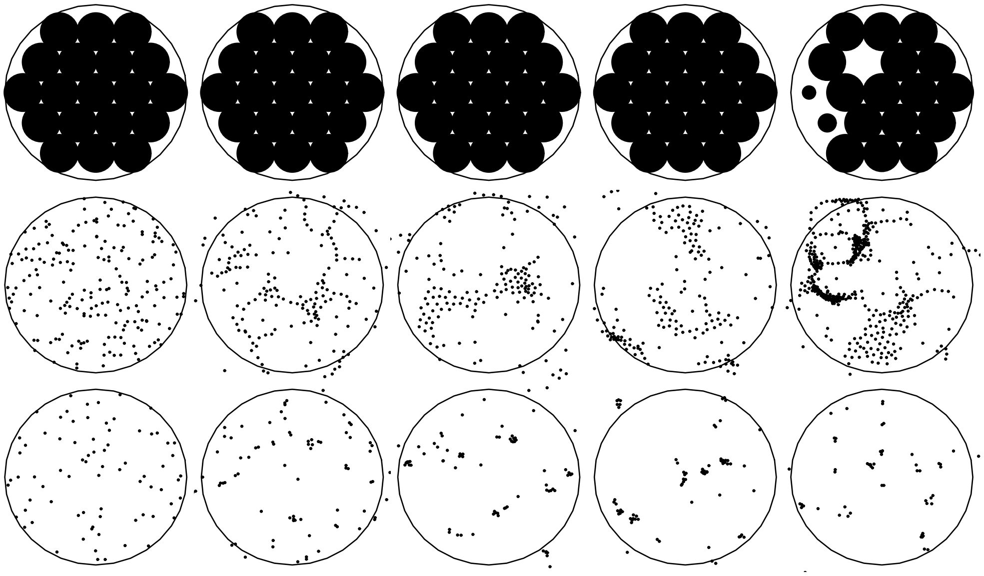

Figure 3.3: Spatial population distributions for the plants, prey and predators, across rows 1,2, and 3, respectively. The distributions are at times $t = 1, 50, 100, 250, 500$, down columns 1 through 5, respectively.

tritrophic model and is an interesting aspect of the model that requires further study. Figure 3.3 displays the formation of groups from the initial random distributions ($t = 1$) to the clustered distributions ($t = 500$). The two types of prey considered had exactly the same behaviour when they were responding to either predators or plants, but their group dynamics were distinctly different when predators and plants were not in their limited vision. Within the influence of the oasis, the prey congregate on plants and reproduce, and when a plant was fully consumed the prey disperse radially to neighbouring plants. This process correlated the individuals through births (Martin 2004) and the common attraction to plants. Large loose groups or herds of prey formed, moving from plant to plant. The predators also formed groups or packs and chased the prey into tight formations that had the appearance of swarming (Couzin et al. 2002). Swarming behaviour was mostly confined to the influence of the oasis, but the pack formations were present within the entire space.

The result of the simulation experiment is that the modified prey appear individually more fit than the unmodified prey. For the experimental result, time is expressed in terms of prey generations. If a prey reproduces, it does so at an average of 175 time units. Using the random initialization procedure outlined in Section 3.3, when the two subspecies of prey were introduced into the simulated ecosystem in equal numbers and allowed to compete, the mean number of prey generations for one subspecies to dominate the population was 149, and 10 times out of 10 the modified prey were the winners in these competitions. Assessing the statistic 10 times out of 10 using a Binomial distribution there is sufficient evidence (P-value $\sim$ 0.001) that the modified prey are more individually fit than the unmodified prey.

Outside of the influence of the oasis, the prey that were not attracted to other prey (unmodified) dispersed radially from the oasis and the prey that were attracted to other prey (modified) formed temporary shoals (Romey 1996) just outside the influential boundary of the oasis. If these shoals of prey were not dispersed in response to predators and one of the shoaling prey perceived a plant, then this prey moved towards the plant and drew other prey to within the influence of the oasis. As well, if the prey within these shoals are not influenced by either a predator or a plant, then these prey eventually formed into pairs or triplets. A pair of prey will either rotate in a circle or follow a trajectory; whereas, a triplet of prey will follow a seemingly random walk. This random behaviour of prey triplets together with shoaling seem to be the emergent social behaviours that were selected as more fit, within the competition experiment.

Figure 3.4 displays the dispersion of the unmodified and modified prey in response to a predator with three different positions of a plant. Each row starts with identical spatial distributions of the prey and a predator moves horizontally through the group of prey. When the prey are initially on top of the plant (Figure 3.4, rows 1 and 2), the unmodified and modified prey behave nearly identically in response to a predator, except that the unmodified prey have a slightly greater chance of being dispersed

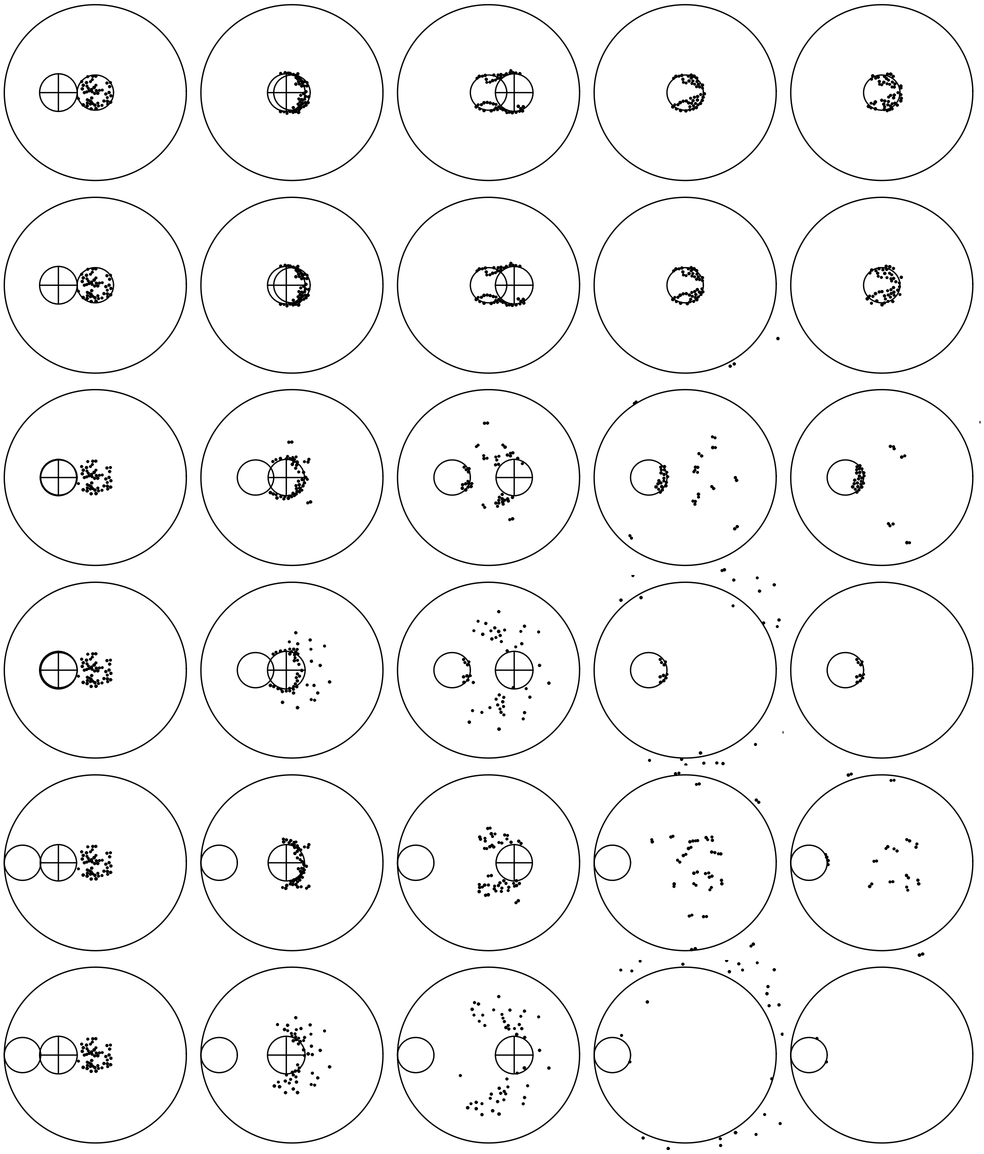

Figure 3.4: Spatial dispersion, in response to a predator (cross-haired circle represents the influence of the predator) with and without the presence of a plant (small open circle), for prey (closed dots) that are not attracted to each other (unmodified, rows 2, 4 and 6) and those that are attracted to each (modified, rows 1, 3 and 5). The oasis (large open circle) is at the same scale as Figure 3.3. The spatial distributions of the prey are at times $t = 1, 50, 100, 250, 500$, down columns 1 through 5, respectively.

outside the influence of the plant and thus become lost. When the prey are not initially on the plant (Figure 3.4, rows 3 through 6), the unmodified and modified prey have distinctly different behaviour, where the unmodified prey disperse and the modified prey coalesce into groups. At time $t = 100$ the formation of shoals is present for the modified prey (Figure 3.4, rows 3 and 5), but by time $t = 250$ these prey have either been attracted to the plant or have formed into pairs and triplets. The unmodified prey disperse from the predator and have a much greater chance of leaving the oasis than the modified prey (Figure 3.4, rows 3 through 6, $t = 500$).

The random behaviour of prey triplets was particularly interesting, but seems to be an artifact of the movement portion of the model. Simulations of three prey without other individuals were performed to assess their behaviour. The dispersal of the centroid of the three prey was assess in relation to the time step Δt. For 1000 time units or $1000/\Delta t$ iterations the dispersal of the centroid dramatically decreased as Δt decreased. If the random behaviour was not an artifact then this behaviour would be evident for all values of Δt. Based on observations of the three prey behaviour, it would seem that the erratic movement of the centroid is a result of the prey overshooting the attraction/repulsion boundaries by a larger amount for larger values of Δt causing an instability in the position of the centroid. It seems reasonable to assume based on simulations that as Δt goes to zero a stationary equilateral triangle would become increasingly more stable for the configuration of the prey triplet. Statistically, the modified prey have an advantage over the unmodified prey as the simulation results indicate, but it is unclear whether this advantage is a property of real ecosystems.

Chapter 4

Schooling

4.1 Introduction

The objective of this chapter is to mathematically describe an individual-based model of schooling where individuals cannot discern other individuals as individuals. The model ecosystem is a strictly deterministic, discrete time, individual-based model of schooling where relationships between individuals are solely based on light intensities.

Movement of the individuals is a based on a fourth-order approximation of a differential equation that describes the forces acting on the individuals. An individual changes its acceleration direction in response to the presence of other individuals and then accelerates in that direction under the influence of frictional drag. Although the individuals are confined to two dimensions, they are modeled as spheres accelerating through a viscous medium.

Individuals have the ability to perceive their environment using directional photo-sensors and adjust the direction of their acceleration in response to the values (shades of gray) of these photo-sensors. The responses are attraction, repulsion, and comfort, and depend on the direction of the photo-sensor. The photo-sensors are the individual's surrogate for distance and direction calculations, and account for an individual obstructing the view of another individual. A higher value of an individual's photo-sensor indicates closer neighbours and repulsion occurs against a direction of a photo-sensor with a high value. Similarly, attraction occurs towards a direction of a photo-sensor with a low value. Comfort is a positive enforcement of current behaviour and occurs at medium values of the photo-sensors. The conditions for high, medium and low values are directionally dependent and oriented with the movement of an individual. Under certain conditions, it was found that these movement rules produced the group dynamic of schooling.

4.2 Schooling Model

The general model framework is defined as a nonlinear map (chapter 2) which progresses a collection of sets of vectors discretely through time. In this schooling model, there is only one population and the superscript k will be omitted and the model system is an individual-based model that is spatially-explicit in two continuous dimensions and strictly deterministic. Individuals are represented as dense spheres whose centres are bounded in a two dimensional plane.

4.2.1 Attribute and Species Definitions

Attributes that the individuals possess are: position in space, radius, velocity, orientation, acceleration direction, and visual perception. The position and velocity of an individual are continuous two-dimensional vectors, the radius is a positive number, the orientation is an angle, the acceleration direction is a continuous two-dimensional unit vector, and the visual perception is an n dimensional vector whose components are contained in a unit interval.

Let $\mathbf{p}_i(t) \in \mathbb{R}^2$, $\mathrm{r}_i(t) \in \mathbb{R}^+$, $\mathbf{v}_i(t) \in \mathbb{R}^2$, $\theta_i(t) \in [0, 2\pi)$, $\mathbf{h}_i(t) \in \mathbb{R}^2$ with $\left\|\mathbf{h}_i(t)\right\| = 1$, and $\mathbf{e}_i(t) \in [0,1]^n$ be the position, radius, velocity, orientation, acceleration direction and oriented visual perception of individual i at time t, respectively. Then

$$\mathrm{M}_i(t) = (\mathbf{p}_i(t), \mathrm{r}_i(t), \mathbf{v}_i(t), \theta_i(t), \mathbf{h}_i(t), \mathbf{e}_i(t))$$

is individual i at time t.

4.2.2 Spatial Occupancy, Directional Distance and Vision

Individuals are oriented dense spheres whose centres are confined in a two-dimensional plane. The radii for all individuals are constant and equal, thus $\mathrm{r}_i(t) = \mathrm{r}$ for all i and t. An individual's orientation is in the direction of its velocity. That is, for individual i at time t the following relationship holds:

$$\big(\cos(\theta_i(t)), \sin(\theta_i(t))\big) = \frac{\mathbf{v}_i(t)}{\|\mathbf{v}_i(t)\|}.$$

The directional distance between two individuals is a function of the centres and radii of those individuals and considered only within the two dimensional plane. The direction distance is derived using analytic geometry by intersecting rays that start from the centre of one individual with the circle defined by another individual. The expression of the directional distance uses the dot product between two vectors, $\langle\cdot,\cdot\rangle$, the vector from the centre of individual i to the centre of individual j, $\mathbf{w}_{ij}(t) = \mathbf{p}_j(t) - \mathbf{p}_i(t)$, and the unit vector in direction φ, $\mathbf{u}(\varphi) = (\cos(\varphi), \sin(\varphi))$. The directional distance from individual i to individual j, $j \neq i$, in direction φ at time t is

$$\mathrm{d}_{ij}(t,\varphi) = \langle \mathbf{w}_{ij}(t), \mathbf{u}(\varphi)\rangle - \big(\mathrm{r}^2 - \langle \mathbf{w}_{ij}(t), \mathbf{u}(\varphi+\pi)\rangle^2\big)^{\frac{1}{2}} - \mathrm{r}$$

if $\langle \mathbf{w}_{ij}(t), \mathbf{u}(\varphi)\rangle \geq 0$ and $\mathrm{r} \geq \langle \mathbf{w}_{ij}(t), \mathbf{u}(\varphi+\pi)\rangle$. Otherwise, $\mathrm{d}_{ij}(t,\varphi) = \infty$ (note: ∞ is considered a number that is larger than any possible distance). A positive value for this distance function indicates that the circles do not intersect (do not overlap), a value of zero indicates that circles intersect at a point (touch) and a negative value indicates that the circles intersect at many points (overlap).

Vision is modeled as an n dimensional array of directional photo-sensors. Each element in this array is contained in the unit interval, $[0, 1]$, and the array for individual

i at time t is

$$\mathbf{e}_i(t) = \left(\mathrm{e}_i^1(t), \mathrm{e}_i^2(t), \mathrm{e}_i^3(t), ..., \mathrm{e}_i^n(t)\right)$$

The direction of photo-sensor $\mathrm{e}_i^j(t)$ is $\theta_i(t) + \frac{j-1}{n}2\pi$ and its value at time t is:

$$\mathrm{e}_i^j(t) = \min\left\{1, \exp\Big(-\lambda \mathrm{d}_i\big(t, \theta_i(t) + \frac{j-1}{n}2\pi\big)\Big)\right\},$$

where λ is the attenuation rate of light in the fluid (Luria and Kinney 1970) and $\mathrm{d}_i(t,\varphi) = \min_{j\neq i}\mathrm{d}_{ij}(t,\varphi)$ is the distance to the closest individual in direction φ from individual i at time t. Figure 4.1 gives an example of five individuals and their perception of each other.

4.2.3 Acceleration Direction

The affect of the vision on the acceleration orientation is a linear combination of the directions for the photo-sensors that depends on the values of the photo-sensors and their directions. The effects are attraction, repulsion, and comfort. Attraction, repulsion, and comfort zones are defined relative to the individual in terms of light intensities and directions. If a value of an individual's photo-sensor is within the attraction zone then the individual will tend towards the direction of that photo-sensor. Similarly, a value within the repulsion zone will result in the individual tending away from the photo-sensor. Within the comfort zone, values of the photo-sensors will have a positive influence on the current orientation of the individual. Finally, if the values of all the photo-sensors are zero then the individual will attempt to maintain its current orientation.

Around an individual, there are four sectors that are defined angularly, laterally symmetric, and relative to the direction of the individual. These sectors are front, side, back, and obscured. The obscured sector is based on the physical limitations of an individual to perceive behind itself and the other sectors are approximations to

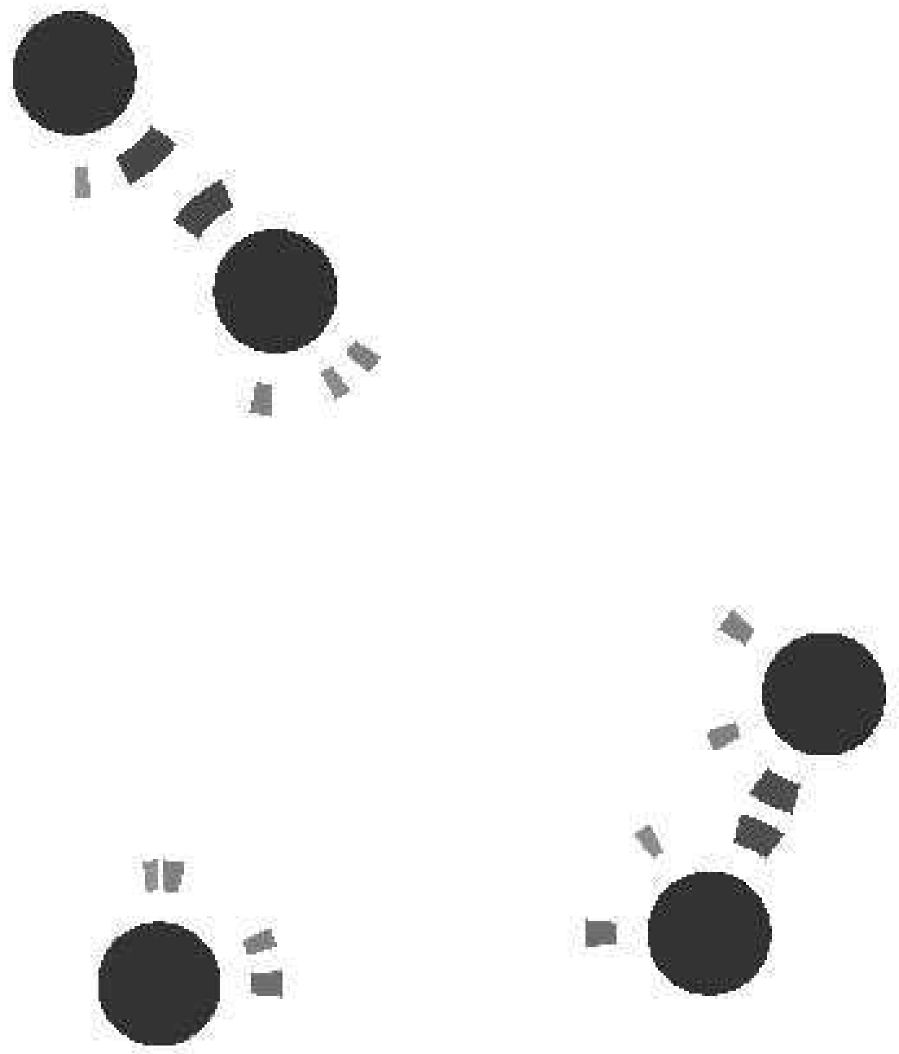

Figure 4.1: Vision of five individuals. Individuals are the black circles and the gray regions surrounding them represents the perception that they have for one another.

continuously varying behaviour. The left front sector is $[0, \beta_f]$, the left side sector is $(\beta_f, \beta_s]$, the left back sector is $(\beta_s, \beta_b]$, and the left obscure sector is $(\beta_b, \pi]$. The right sectors are defined as reflections of the left sectors about the polar origin. The front, side and back sectors are reaction sectors; whereas, the obscure sector has no bearing on the behaviour of the individual. Within these reaction sectors there are repulsion, comfort, and attraction zones. From the perspective of the observer, these zones are based on the distance from the individual where $[0, \mathrm{d}_f^r]$, $(\mathrm{d}_f^r, \mathrm{d}_f^c]$, and $(\mathrm{d}_f^c, \mathrm{d}_f^a]$ are the intervals for repulsion, comfort, and attraction within the front sector. Similarly, $[0, \mathrm{d}_s^r]$, $(\mathrm{d}_s^r, \mathrm{d}_s^c]$, and $(\mathrm{d}_s^c, \mathrm{d}_s^a]$, and $[0, \mathrm{d}_b^r]$, $(\mathrm{d}_b^r, \mathrm{d}_b^c]$, and $(\mathrm{d}_b^c, \mathrm{d}_b^a]$ are the intervals for the side and back sectors, respectively.

In order to develop the perspective of the individuals, a transformation of distance to light intensity is required. The exponential decay transformation of a set S with decay parameter λ is defined to be $E_\lambda(S) = \{exp(-\lambda s) \big| s \in S\}$ (Luria and Kinney 1970). Also, the reflection of a region $G = I \times P$, $I \subset [0, \infty)$ and $P \subset [0, 2\pi)$, described in polar coordinates is defined as $\overset{\leftrightarrow}{G} = \{(i, -p \bmod 2\pi) \big| (i,p) \in G\}$. Then the repulsion, comfort, and attraction regions are defined, respectively as:

$$\mathscr{R} = \mathscr{R}_l \cup \overset{\leftrightarrow}{\mathscr{R}_l}; \quad \mathscr{C} = \mathscr{C}_l \cup \overset{\leftrightarrow}{\mathscr{C}_l}; \quad \text{and} \quad \mathscr{A} = \mathscr{A}_l \cup \overset{\leftrightarrow}{\mathscr{A}_l}\,. \tag{4.1}$$

Where

$$\mathscr{R}_l = E_\lambda\big([0, \mathrm{d}_f^r]\big) \times [0, \beta_f] \cup E_\lambda\big([0, \mathrm{d}_s^r]\big) \times (\beta_f, \beta_s] \cup E_\lambda\big([0, \mathrm{d}_b^r]\big) \times (\beta_s, \beta_b],$$

$$\mathscr{C}_l = E_\lambda\big((\mathrm{d}_f^r, \mathrm{d}_f^c]\big) \times [0, \beta_f] \cup E_\lambda\big((\mathrm{d}_s^r, \mathrm{d}_s^c]\big) \times (\beta_f, \beta_s] \cup E_\lambda\big((\mathrm{d}_b^r, \mathrm{d}_b^c]\big) \times (\beta_s, \beta_b],$$

and

$$\mathscr{A}_l = E_\lambda\big((\mathrm{d}_f^c, \mathrm{d}_f^a]\big) \times [0, \beta_f] \cup E_\lambda\big((\mathrm{d}_s^c, \mathrm{d}_s^a]\big) \times (\beta_f, \beta_s] \cup E_\lambda\big((\mathrm{d}_b^c, \mathrm{d}_b^a]\big) \times (\beta_s, \beta_b].$$

Figure 4.2 displays an example of an observer's perspective of an individual's perception zones.

The acceleration direction of an individual is a cumulative response to the values of its directional photo-sensors and its current orientation. Individual i at time t is oriented in the direction $\theta_i(t)$ and the direction of its photo-sensor j is $\theta_i(t) + 2\pi(j-1)/n$. At time t for individual i, if the value of its photo-sensor j, $\mathrm{e}_i^j(t)$, in the relative direction $2\pi(j-1)/n$ is within its attraction zone, $\mathscr{A}$, then the individual will have a response towards the direction of that photo-sensor equal to the value of the photo-sensor. Similarly, a value within the repulsion zone, $\mathscr{R}$, will result in an individual responding away from the direction of the photo-sensor. Within the comfort zone, $\mathscr{C}$, values of the photo-sensors will have a positive influence on the current direction of the individual. Finally, if the values of all the photo-sensors are zero then the individual will maintain its current orientation. The acceleration direction of individual i at time $t + \Delta t$ is

$$\mathbf{h}_i(t+\Delta t) = \frac{(cos(\theta_i(t)), sin(\theta_i(t))) + \sum_{j=1}^{n} \mathbf{k}\left(\mathrm{e}_i^j(t), \theta_i(t), \frac{j-1}{n}2\pi\right)}{\|(cos(\theta_i(t)), sin(\theta_i(t))) + \sum_{j=1}^{n} \mathbf{k}\left(\mathrm{e}_i^j(t), \theta_i(t), \frac{j-1}{n}2\pi\right)\|},$$

where $\mathbf{k}(\mathrm{e}, \theta, \phi)$ is the directional influence of the value of e on individual i at time t in direction ϕ relative to θ and defined as:

$$\mathbf{k}(\mathrm{e}, \theta, \phi) = \begin{cases} -\mathrm{e}\,(cos(\theta+\phi), sin(\theta+\phi)) & \text{if } (\mathrm{e}, \phi) \in \mathscr{R} \\ \mathrm{e}\,(cos(\theta), sin(\theta)) & \text{if } (\mathrm{e}, \phi) \in \mathscr{C} \\ \mathrm{e}\,(cos(\theta+\phi), sin(\theta+\phi)) & \text{if } (\mathrm{e}, \phi) \in \mathscr{A} \\ (0,0) & \text{otherwise} \end{cases}.$$

4.2.4 Movement

Movement of an individual is based on a fourth-order approximation of a differential equation that describes the forces acting on the individual. An individual changes its acceleration direction in response to other individuals and then accelerates in that direction under the influence of frictional drag.

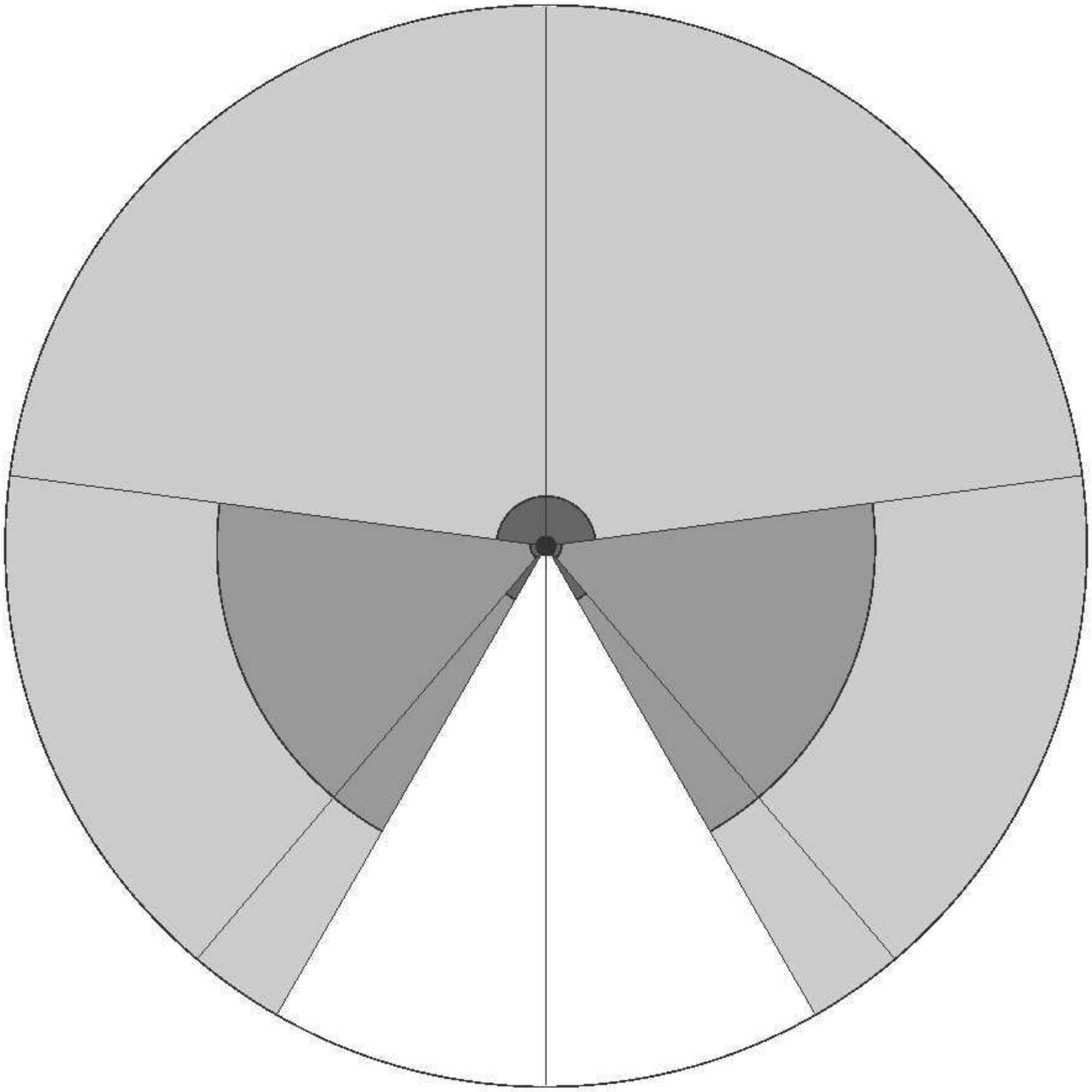

Figure 4.2: Reaction zones of an individual as seen from the perspective of the observer. The black circle is the individual, the dark gray region is the repulsion zone, the medium gray region is the comfort zone, the light gray region is the attraction zone, and the white region has no effect on the individuals behaviour. This representation of an individuals reaction zones is similar to Kunz and Hemelrijk (2003)

Let $\mathbf{F}_i(t)$ be the total force acting on individual i at time t and

$$\mathbf{F}_i(t) = m_i(t)\frac{\mathrm{d}\mathbf{v}_i(t)}{\mathrm{d}t} = m_i(t)\frac{\mathrm{d}^2\mathbf{p}_i(t)}{\mathrm{d}t^2}, \tag{4.2}$$

where $\mathrm{m}_i(t)$ is the mass of individual i at time t, respectively. All individuals are neutrally buoyant and the mass densities for both the individuals and the surrounding fluid are taken as unity. Thus,

$$\mathrm{m}_i(t) = \mathrm{m} = \frac{4}{3}\pi \mathrm{r}^3.$$

Individuals apply an internal force in a desired direction under the influence of the force of fluid drag. For the individuals modeled the force of fluid drag is assumed proportional to an individuals velocity (Emlet et al. 1985, Long and Weiss 1999, Vogel 1994) and the total force on individual i at time t is

$$\mathbf{F}_i(t) = \mathrm{m}\mathbf{g}_i(t) - \mathrm{c}_i(t)\mathbf{v}_i(t), \tag{4.3}$$

where $\mathbf{g}_i(t)$ is the internal acceleration generated by individual i at time t and $\mathrm{c}_i(t)$ is the frictional drag on individual i at time t. $\mathbf{g}_i(t)$ is assumed proportional to the individual's surface area in the direction $\mathbf{h}_i(t)$. Let μ be the population parameter that defines the acceleration per unit surface area and

$$\mathbf{g}_i(t) = \mu 4\pi \mathrm{r}^2\mathbf{h}_i(t). \tag{4.4}$$

The frictional drag $\mathrm{c}_i(t)$ is assumed proportional to the individual's radius through the following relationship (Emlet et al. 1985, Long and Weiss 1999):

$$\mathrm{c}_i(t) = \mathrm{c} = 6\pi\eta \mathrm{r}, \tag{4.5}$$

where η is the viscosity of the fluid. Then, using equations 4.2, 4.3, 4.4, and 4.5 the velocity of individual i at time t is the solution to the following differential equation:

$$\frac{\mathrm{d}\mathbf{v}_i(t)}{\mathrm{d}t} = \mu 4\pi \mathrm{r}^2\mathbf{h}_i(t) - \frac{9\eta}{2\mathrm{r}^2}\mathbf{v}_i(t) \tag{4.6}$$

A fourth order Runge-Kutta approximation of an ordinary differential equation (Golub and Ortega 1992) that describes an object moving through a fluid will provide discrete time difference equations for the movement and velocity of the individuals. In order to develop the conditions on the time step Δt for the numerical approximation of equation 4.6, consider the one dimensional situation of an individual starting at rest on the origin and moving in a positive direction, $\mathbf{v}_i(t) = \mathrm{v}(t)$, $\mathbf{h}_i(t) = 1$ and $\mathbf{g}_i(t) = \mathrm{g}$. Then, the solution to the differential equation 4.6 is

$$\mathrm{v}(t) = \frac{\mathrm{mg}}{\mathrm{c}}\left(1 - e^{-\frac{\mathrm{c}}{\mathrm{m}}t}\right) = \frac{8\pi\mu\mathrm{r}^4}{9\eta}\left(1 - e^{-\frac{9\eta}{2\mathrm{r}^2}t}\right) \tag{4.7}$$

Using equation 4.7, an individual has a maximum velocity of mg/c and can attain this velocity at an exponential rate of c/m. For a numerical approximation of equation 4.6 to be precise, an individual will move at most one quarter of its body length during a time step and Δt will satisfy the following:

$$\frac{\mathrm{mg}}{\mathrm{c}}\Delta t \leq \frac{\mathrm{r}}{2} \quad \text{or} \quad \Delta t \leq \frac{9\eta}{16\pi\mu\mathrm{r}^3} \tag{4.8}$$

For numerical stability of a forth-order Runge-Kutta approximation to equation 4.6, Δt must satisfy the following (Abbas 1966):

$$\frac{\mathrm{c}}{\mathrm{m}}\Delta t \leq \zeta \quad \text{or} \quad \Delta t \leq \zeta\frac{2\mathrm{r}^2}{9\eta}, \tag{4.9}$$

where

$$\zeta = \frac{1}{3}\left(\left(4\left(2349\right)^{\frac{1}{2}} + 172\right)^{\frac{1}{3}} - \left(4\left(2349\right)^{\frac{1}{2}} - 172\right)^{\frac{1}{3}} + 4\right).$$

If Δt satisfies the conditions imposed by equations 4.8 and 4.9 then accurate and precise stable solutions for the position and velocity of individual i at time $t + \Delta t$

are, respectively:

$$\mathbf{p}_i(t+\Delta t) = \mathbf{p}_i(t) + \mathbf{v}_i(t)\Delta t + \frac{(\Delta t)^2}{24}\left(12 - 4\frac{\mathrm{c}\Delta t}{\mathrm{m}} + \left(\frac{\mathrm{c}\Delta t}{\mathrm{m}}\right)^2\right)\left(\mathbf{g}_i(t) - \frac{\mathrm{c}}{\mathrm{m}}\mathbf{v}_i(t)\right),$$

and

$$\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t) + \frac{\Delta t}{24}\left(24 - 12\frac{\mathrm{c}\Delta t}{\mathrm{m}} + 4\left(\frac{\mathrm{c}\Delta t}{\mathrm{m}}\right)^2 - \left(\frac{\mathrm{c}\Delta t}{\mathrm{m}}\right)^3\right)\left(\mathbf{g}_i(t) - \frac{\mathrm{c}}{\mathrm{m}}\mathbf{v}_i(t)\right).$$

4.2.5 Collisions

Collisions between two individuals are inelastic (plastic) and result when the distance between them is less than or equal to zero. A collision is an instantaneous reaction and individual i will collide with individual j when $\mathrm{d}_{ij}(t,\alpha) \leq 0$ for some angle α and the velocity of individual i at time t is

$$\mathbf{v}_i(t) = \frac{\mathrm{m}_i(t)\mathbf{v}_i(t) + \mathrm{m}_j(t)\mathbf{v}_j(t)}{\mathrm{m}_i(t) + \mathrm{m}_j(t)} = \frac{\mathbf{v}_i(t) + \mathbf{v}_j(t)}{2}.$$

4.3 Results

The experimental objective was to create a model that could assess schooling behaviour where individuals cannot discern other individuals as individuals. The model was implemented using the compiled computer language C (GCC 2008a). Spatial organization of the individuals was assessed through animations of model output produced using OpenGL (OpenGL 2008) directives in C. The computer code is displayed in Appendix B and this code along with animations can be viewed at the website: Charnell (2008b).

The parameterized model described in section 4.2 is a plausible representation of schooling individuals. The schooling model is fully implemented and parameterized. The parameters used are light attenuation decay rate, $\lambda = 100$, fluid viscosity, $\eta = 2$, radius of individuals, r $= 1$, acceleration per unit surface area of an individual, $\mu = 1$, and 128 directional photo-sensors, n. The parameterized reaction zones are displayed in in figure 4.3 and the left repulsion, comfort and attraction regions are, respectively:

$$\mathscr{R}_l = E_{100}\left([0,5]\right) \times \left[0, \frac{\pi}{3}\right] \cup E_{100}\left([0,1]\right) \times \left(\frac{\pi}{3}, \frac{3\pi}{4}\right] \cup E_{100}\left([0,2]\right) \times \left(\frac{3\pi}{4}, \frac{5\pi}{6}\right],$$

$$\mathscr{C}_l = E_{100}\left((5,5]\right) \times \left[0, \frac{\pi}{3}\right] \cup E_{100}\left((1,50]\right) \times \left(\frac{\pi}{3}, \frac{3\pi}{4}\right] \cup E_{100}\left((2,50]\right) \times \left(\frac{3\pi}{4}, \frac{5\pi}{6}\right],$$

and

$$\mathscr{A}_l = E_{100}\left((5,\infty)\right) \times \left[0, \frac{\pi}{3}\right] \cup E_{100}\left((50,\infty)\right) \times \left(\frac{\pi}{3}, \frac{3\pi}{4}\right] \cup E_{100}\left((50,\infty)\right) \times \left(\frac{3\pi}{4}, \frac{5\pi}{6}\right].$$

The parameterized schooling model was initialized with 100 individuals randomly distributed within a circle of radius 100. Each individuals was assigned an initial magnitude of 1 for its velocity in a random direction. Figure 4.4 displays a typical progression of the schooling model. Initially, the individuals congregate towards the centre of the group which requires approximately 100 iterations to complete. Then, for about 1000 iterations, the individuals form a temporary shoal that has within it clusters of parallel oriented individuals. These clusters move through the shoal collecting other individuals and dissipating. It appears that the shoal progresses into a small school when one of the oriented clusters gains a sufficient number of individuals or momentum to continue outside the main group. This small school draws non-schooling individuals into the school until the all the individuals are within the school.

Once a school has formed, it is a very stable structure. Individuals may leave the side of the school only to rejoin further back on the school. This circumstance occurs soon after the school forms and afterwards the school seemingly maintains its configuration indefinitely (more than 50,000 iterations). Model output was viewed

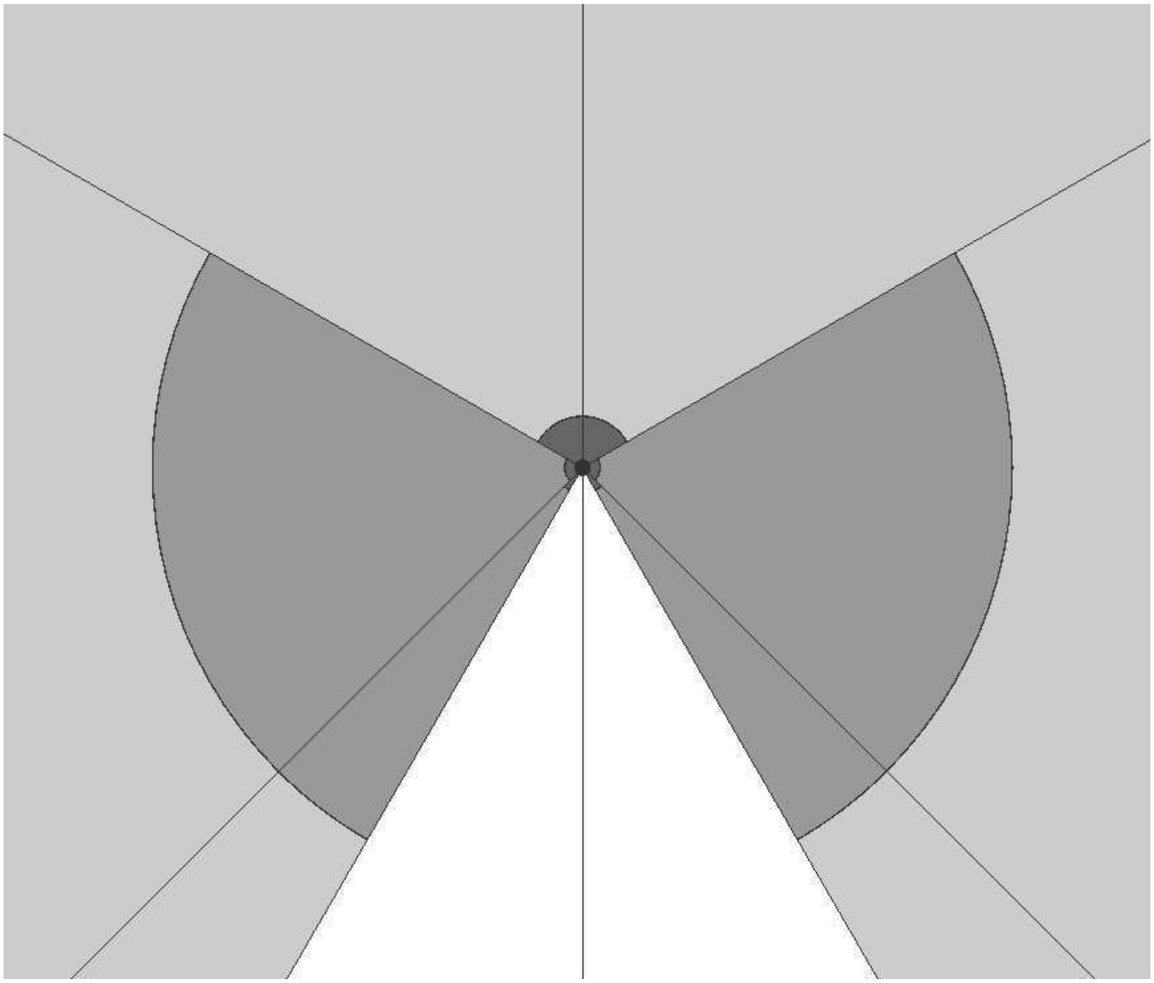

Figure 4.3: Reaction zones of an individual as seen from the perspective of the observer that lead to schooling. The black circle is the individual, the dark gray region is the repulsion zone, the medium gray region is the comfort zone, the light gray region is the attraction zone, and the white region has no effect on the individuals behaviour.

approximately fifty times and there was one occasion where the group of individuals split into two schools. The formation of schools with individuals defined by the model in section 4.2 is an emergent property and the resulting schools are stable.

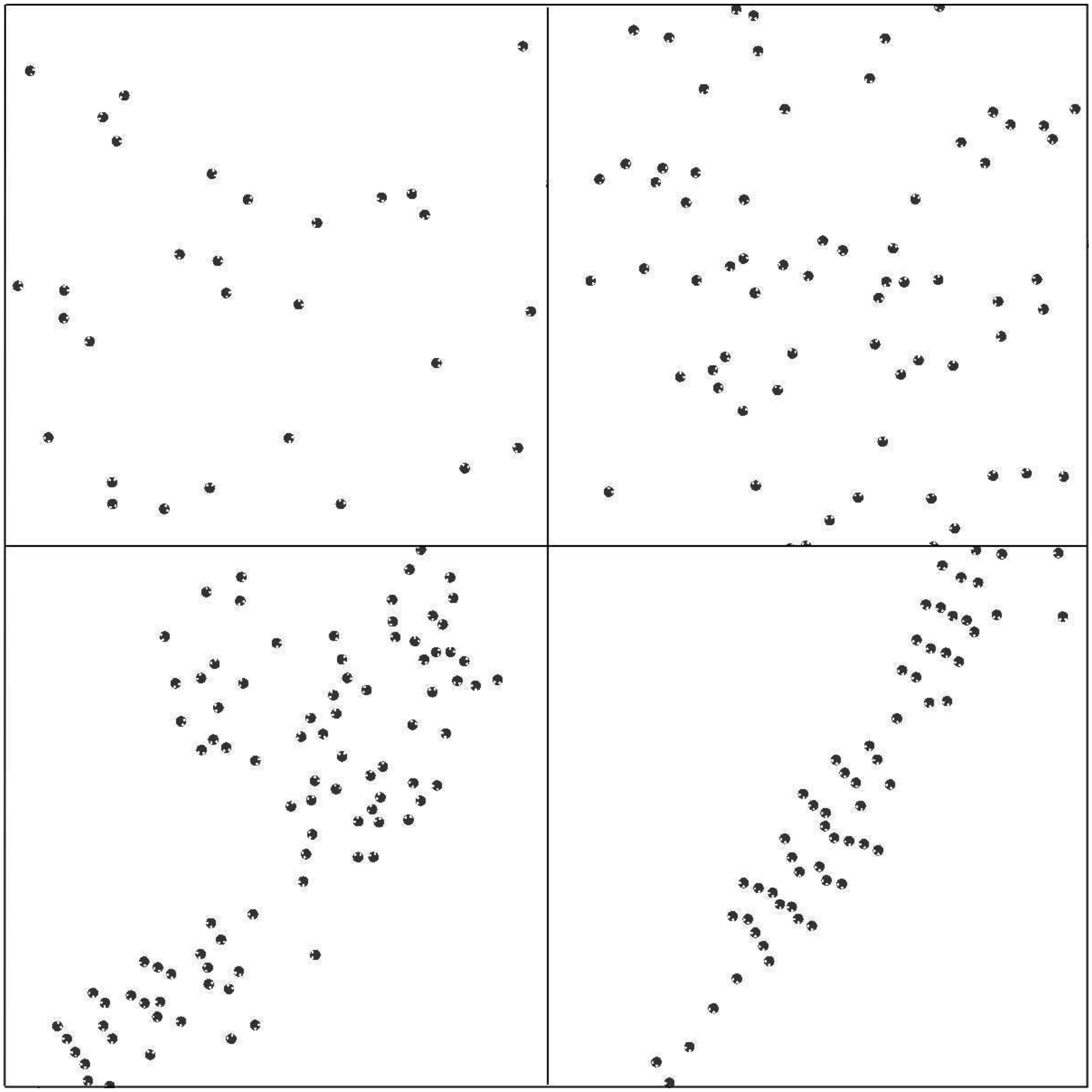

Figure 4.4: Results of the schooling model. Each box width is 50 units square and centered at the centre of mass for a group of 100 individuals. Top left, top right, bottom left and bottom right boxes are snapshots of the schooling model taken at iterations 1, 100, 500, and 1000, respectively.

Chapter 5

Conclusions

The main purpose of this thesis was to present the unambiguous mathematical formalization of individual-based models for generic discrete-time multi-trophic ecosystems that can assess behavioral characteristics that lead to the spatial organization of individuals. Prey that can be attracted to other prey are significantly more fit on an individual basis than prey that do not have this characteristic under the assumptions of the parameterized tritrophic model and individuals do not necessarily require the ability to match the orientations of their neighbours in order to develop schooling formations. These qualities are a consequence of interactions between individuals and individual-based models are able to assess these components of an ecosystem.

The specific models were developed from the perspective that the mobile individuals could perceive their environment and react using simple attraction, comfort, and repulsion rules. Also, prey are able to discern the differences between plants, predators, and other prey; and predators are able to discern prey from themselves. A tritrophic food chain with individuals that have these minimal characteristics seems to be as simple as possible and still able to generate a working ecosystem with the capacity to assess the evolution of gregarious behaviour in a prey species. The concept of individuals providing positive enforcement to each other's current behaviour, comfort, is much simpler than orientation matching and still leads to schooling groups.

The first version of these models used points for individuals as apposed to the use of dense circular disks or spheres. The use of points simplified some of the mathematical expressions and reduced computation time for simulations, but the behaviour of individuals were not as natural. Other complexities that were considered and not implemented in the tritrophic model were the use of field of view for vision and the important aspects of energy and metabolism. Instead of nearest-neighbour distance (vision with depth) an object that occupies a larger field of view would be perceived as closer (vision with width). Vision with depth and width was implemented in the schooling model and it is not clear how modifying vision in the tritrophic would change the dynamics of the system, but the vision in the schooling model is certainly more realistic.

A major aspect of an ecosystem that is not considered in the tritrophic model is energy. Individuals must gather sufficient resources to meet metabolic needs and, if not met, the individual dies (Schmitz 2000). Reproducing prey and predators within the tritrophic model spend, on average, $\sim$40% of the time feeding and $\sim$60% of the time not feeding before these individuals reproduce. Statistics could be used to develop energy budgets for the individuals where the dynamics of the model ecosystem is maintained and metabolic considerations can be incorporated. Death by consumption or aging was considered in the model and death by starvation is another natural part of an ecosystem that could be incorporated in the model.

The specific models presented in this thesis are validated based on their assumptions (Uchmański 1999) and/or their ability to generate stable natural phenomena (Booth 1997). These models were designed to be generic using well-established biological assumptions (Uchmański 1999) and parameterized for the generation of persistent natural phenomena. The parameterized tritrophic model persisted for many thousands of prey generations and the prey grouped together in the appearance of ‘herds’ and ‘shoals’; as well, the predators seemed to behave as ‘packs’. Herds are groups of individuals that act together without coordination and were formed as a result of prey moving from plant to plant. Shoals are groups of individuals that maintain a general position and the prey formed these groups as a result of feeding on plants. Packs of predators resulted from pursuing groups of prey, but these groups only appeared to be packs as these groups were not coordinated. Packs are defined as groups of individuals that hunt using coordinated behaviour and the predators within the tritrophic model do not communicate. The schooling model was designed to produce schooling behaviour, but the structure of the resulting schools was not designed. These group formations are a result of individuals solely acting on the basis of their own perception of the system (Reynolds 1987).

The random behaviour of prey triplets was an aspect of the tritrophic model that was potentially an artifact of the movement portion of the model. Although, the movement of individuals within the tritrophic model appeared 'natural' it is clear that the movement is not 'natural'. The movement in the schooling model is based on the physics of bodies moving in fluids and movement portion of this model should be incorporated into the tritrophic model. The population dynamics in the tritrophic model is a good representation of a generic tritrophic ecosystem with small population sizes and with the incorporation of the movement portion of the schooling model should provide a better simulation environment that assesses gregarious behaviour in a prey species.

The specific models presented in this thesis were formulated using set theory, analytic geometry, and difference equations. Individual-based models are generally intractable, in the sense that analytic mathematical solutions of model behaviour are not possible. Although intractable, statistical models can be derived from individual-based models and capture relationships between model parameters and emergent model behaviour (Flierl et al. 1999). There is still the possibility that new mathematical tools will be developed for the individual-based approach that have analytic solutions (Łomnicki 1992). In the schooling model, the school was clearly a statistically stable structure in the sense that in all simulations the individuals formed schools and it is not clear how the stability of this structure could be proved mathematically.

Parameterization was a major component in the development of the specific models presented in this thesis. Nature has produced an immense variety of persistent systems and it seems clear that there is not a particular set of parameter values that is optimal and that multivariate surfaces of parameter values could be constructed that satisfy optimality conditions. Individual-based models provide an experimental system for investigating the way population phenomena emerge from individual behaviour (Beecham and Farnsworth 1998) and individual-based modellers should take the approach of experimentalists and play with the model (Grimm 1999). The parameter values and initial population sizes displayed in table 3.1 were sufficient for assessing gregarious behaviour in a prey species, but experiments should be performed to assess how changing parameter values changes emergent model behaviour (Flierl et al. 1999). Similarly, the reaction zones displayed in section 4.3 produced a particular type of schooling and varying the configuration of these zones along with the physical parameters of the model could elicit classes of schooling behaviour.

Individual-based models can be used as 'virtual ecosystems' that can be studied and used to investigate methods for studying real ecosystems (Huston et al. 1988, Berger et al. 1999, Grimm et al. 1999, Kostova et al. 2004). 'Virtual laboratories' allow the testing of hypotheses involving spatial relationships (Kostova et al. 2004) that would be infeasible in a real laboratory. The virtual ecologist's approach has a high potential for optimizing field methods, empirical measures, and sampling protocols (Grimm et al. 1999). Virtual ecology could be extremely powerful, especially in combination with statistics (Grimm et al. 1999). The models presented in this thesis could provide such virtual laboratory environments.

The models presented in this thesis have been used as ‘virtual ecosystems’ and preliminary results have been attained. The model framework allows many elaborations and further experimentation and preliminary investigations in several directions have been made. Using the tritrophic model, the intra-specific repulsion distance was varied within the prey population and competition between the prey seemed to support the hypothesis that when the predators are faster then the prey can congregate in tighter groups. The tritrophic model with the movement portion of the schooling model together with energy as a component was used to create surfaces within the parameter space that described the persistence the model. These surfaces had many peaks and there seemed to be evidence to support the contention that ecosystems within a higher viscosity medium require the relative speed of the predators to the prey to also be higher and that within a low viscosity medium the predators are actually slower than the prey. The reaction zones of the schooling model were adjusted and it seems that stable schools will result without the need for a comfort zone. These partial results are still being developed and this modelling paradigm allows inquiry into ecological ideas that may not have been approachable using other methods.

The parameterized models presented in this thesis are working models that requires further revisions and study. As it stands, the tritrophic model provides a realistic representation of a generic tritrophic ecosystem and can be used to address evolutionary aspects of social behaviour. The schooling model sets a new standard for the way individuals perceive one another in a simulated environment. The general model, non-linear map, should provide a framework for communicating individual-based models and the specific models can be used as examples of this communication. This thesis provides evolutionary ecologists new tools to assess animal behaviour and mathematicians new problems that require analytic solutions.

Bibliography

Abbas, I., 1966. The stability of the fourth order runge-kutta method for the solution of systems of differential equations. Communacation of the Association for Computing Machinery 9 (2), 113–116.

Adioui, M., Treuil, J., Arino, O., 2003. Alignment in a fish school: A mixed lagrangian-eulerian approach. Ecological Modelling 167, 19–32.

Beecham, J., Farnsworth, K., 1998. Animal foraging from an individual perspective: An object oriented model. Ecological Modelling 113, 141–156.

Berec, L., 2002. Techniques of spatially explicit individual-based models: Construction, simulation and mean-field analysis. Ecological Modelling 150, 55–81.

Berger, U., Wagner, G., Wolff, W., 1999. Virtual biologist observe virtual grasshoppers: An assessment of different mobility parameters for the analysis of movement patterns. Ecological Modelling 115, 119–127.

Bolker, B., Deutchman, D., Hartvigsen, G., Smith, D., 1997. Individual-based modelling: What is the difference? Trends in Ecology and Evolution 12, 111–111.

Booth, G., 1997. Gecko: A continuous 2d world for ecological modeling. Artificial Life. 3, 147–163.

Breckling, B., Middelhoff, U., Reuter, H., 2006. Individual-based models as tools for ecological theory and application: Understanding the emergence of organisational properties in ecological systems. Ecological Modelling 194, 102–113.

Breckling, B., Müller, F., Reuter, H., Hölker, F., Fränzle, O., 2005. Emergent properties in individual-based ecological models–introducing case studies in an ecosystem research context. Ecological Modelling 186, 376–388.

Charnell, M., 2008a. An individual-based model of a tritrophic ecology. Ecological Modelling 218, 195–206.

Charnell, M., December 2008b. Moshi's home page. `http://www.math.uvic.ca/~moshi/`.

Conrad, M., Pattee, H., 1970. Evolution experiments with an artificial ecosystem. Journal of Theoretical Biology 28, 393–409.

Couzin, I., Krause, 2003. Self-organization and collective behaviour in vertebrates. Advances in the Study of Behaviour 32, 1–75.

Couzin, I., Krause, J., James, R., Ruxton, G., Franks, N., 2002. Collective memory and spatial sorting in animal groups. Journal of Theoretical Biology 218, 1–11.

Dumont, B., Hill, D., 2004. Spatially explicit models of group foraging by herbivores: What can agent-based models offer? Animal Research 53, 419–428.

Emlet, R., Strathman, R., Strickler, J., 1985. Gravity, drag, and feeding currents of small zooplankton. Science 228 (4702), 1016–1017.

Fahse, L., Wissel, C., Grimm, V., 1998. Reconciling classical and individual-based approaches in theoretical population ecology: A protocol for extracting parameters from individual-based models. The American Naturalist 152, 838–852.

Ferreira, J., 1995. Ecowin – an object-oriented ecological model for aquatic ecosystems. Ecological Modelling 79, 21–34.

Flierl, G., Grünbaum, D., Levin, S., Olson, D., 1999. From individuals to aggregations: The interplay between behaviour and physics. Journal of Theoretical Biology 196, 397–454.

GCC, December 2008a. Gcc, the gnu compiler collection - gnu project - free software foundation (fsf). `http://gcc.gnu.org/`.

GCC, December 2008b. Gfortran - gcc wiki. `http://gcc.gnu.org/fortran/`.

GIMP, December 2008. Gimp - the gnu image manipulation program. `http://www.gimp.org/`.

Ginot, V., Le Page, C., Souissi, S., 2002. A multi-agent architecture to enhance end-user individual-based modelling. Ecological Modelling 157, 23–41.

Golub, G., Ortega, J., 1992. Scientific Computing and Differential Equations: An Introduction to Numerical Methods. Academic Press, Toronto, ON.

Grimm, V., 1999. Ten years of individual-based modelling in ecology: What have we learned and what could we learn in the future? Ecological Modelling 115, 129–148.

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Giont, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S., Huse, G., Huth, A., Jespen, J., Jørgensen, C., Mooij, W., Müller, B., Pe'er, G., Piou, C., Railback, S., Robbins, A., Robbins, M., Rossmanith, E., Rüger, N., Strand, E., Souissi, S., Stillman, R., Vabø, R., Visser, U., DeAngelis, D., 2006. A standard protocol for describing individual-based and agent-based models. Ecological Modelling 198, 115–126.

Grimm, V., Wyszomirski, T., Aikman, D., Uchmanski, J., 1999. Individual-based modelling and ecological theory: Synthesis of a workshop. Ecological Modelling 115, 275–282.

Gueron, S., Levin, S., Ruenstein, D., 1996. The dynamics of herds: From individuals to aggregations. Journal of Theoretical Biology 182, 85–98.

Huston, M., DeAngelis, D., Post, W., 1988. New computer models unify ecological theory. BioScience 38, 682–691.

James, R., Bennett, P., Krause, J., 2004. Geometry for mutualistic and selfish herds: The limited domain of danger. Journal of Theoretical Biology 228, 107–113.

Kostova, T., Carlsen, T., Kercher, J., 2004. Individual-based spatially-explicit model of an herbivore and its resource: The effect of habitat reduction and fragmentation. Comptes Rendus Biologies 327, 261–276.

Krebs, C., 1985. Ecology: The Experimental Analysis of Distribution and Abundance, 3rd Ed. HarperCollins, New York, NY.

Kreft, J., Booth, G., Wimpenny, J., 1998. Bacsim, a simulator for individual-based modelling of bacterial colony growth. Microbiology 144, 3275–3287.

Kunz, H., Hemelrijk, C., 2003. Artificial fish schools: Collective effects of school size, body size, and body form. Artificial Life 9, 237–253.

Łomnicki, A., 1992. Population Ecology from the Individual Perspective, in: DeAngelis, D.L., Gross, L. (Ed.) Individual-Based Models and Approaches in Ecology. Chapman & Hall, New York, NY.

Łomnicki, A., 1999. Individual-based models and the individual-based approach to population ecology. Ecological Modelling 115, 191–198.

Long, L., Weiss, H., 1999. The velocity dependance of areodynamic drag: A primer for mathematicians. The American Mathematical Monthly 106 (2), 127–135.

Lorek, H., Sonnenschein, M., 1999. Modelling and simulation software to support individual-based ecological modelling. Ecological Modelling 115, 199–216.

Luria, S., Kinney, J., 1970. Underwater vision. American Association for the Advancement of Science 167 (3924), 1454–1461.

Martin, A., 2004. A malthusian curb on spatial structure in microorganism populations. Journal of Theoretical Biology 230, 343–349.

Minar, N., Burkhart, R., Langton, C., Askenazi, M., 1996. The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations. Working Paper 96-06-042. Santa Fe Institute, Sante Fe, NM.

Mirabet, V., Auger, P., Lett, C., 2007. Spatial structures in simulations of animal grouping. Ecological Modelling 201, 468–476.

Mogilner, A., Edelstein-Keshet, L., Bent, L., Spiros, A., 2003. Mutual interactions, potentials, and individual distance in a social aggregation. Journal of Mathematical Biology 47, 353–389.

Morton, T., Haefner, J., Nugala, V., Decino, R., Mendes, L., 1994. The selfish herd revisited: Do simple movement rules reduce relative predation risk? Journal of Theoretical Biology 167, 73–79.

Murdoch, W., McCauley, E., Nisbet, R., Gurney, W., de Roos, A., 1992. Population Ecology from the Individual Perspective, in: DeAngelis, D.L., Gross, L. (Ed.) Individual-Based Models and Approaches in Ecology. Chapman & Hall, New York, NY.

OpenGL, December 2008. Opengl - the industry standard for high performance graphics. `http://www.opengl.org/`.

Ovaskainen, O., Hanski, I., 2004. From individual behaviour to metapopulational dynamics: Unifying the patchy population and classical metapopulational models. The American Naturalist 164, 364–377.

Poppy, G., 1997. Tritrophic interactions: Improving ecological understanding and biological control? Endeavour 21, 61–65.

R, December 2008. The r project for statistical computing. `http://www.r-project.org/`.

Railsback, S., 2001. Concepts from complex adaptive systems as a framework for individual-based modelling. Ecological Modelling 139, 47–62.

Reluga, T., Vicido, S., 2005. Simulated evolution of selfish herb behaviour. Journal of Theoretical Biology 234, 213–225.

Reynolds, C. W., 1987. Flocks, herds, and schools: A distributed behavioural model. Computer Graphics. 21 (4), 25–34.

Romey, W., 1996. Individual differences make a difference in the trajectories of simulated schools of fish. Ecological Modelling 92, 65–77.

Schmitz, O. J., 2000. Combining field experiments and individual-based modeling to identify the dynamically relevant organizational scale in a field system. Oikos. 89, 471–484.

Schofield, P., Chaplain, M., Hubbard, S., 2005. Evolution of searching and life history characteristics in individual-based models of host-parasitiod-microbe associations. Journal of Theoretical Biology 237, 1–16.

Souissi, S., Seuront, L., Schmitt, F., Giont, V., 2005. Describing space-time patterns in aquatic ecology using ibms and scaling and multi-scaling approaches. Nonlinear Analysis: Real World Applications. 6, 705–730.

Uchmański, J., 1999. What promotes persistence of a single population: An individual-based model. Ecological Modelling 115, 227–241.

Uchmański, J., Grimm, V., 1996. Individual-based modelling in ecology: What makes the difference? Trends in Ecology and Evolution 11, 437–441.

Viscido, S., Parrish, J., Grünbaum, D., 2004. Individual behaviour and emergent properties of fish schools: A comparison of observation and theory. Marine Ecology Progress Series 273, 239–249.

Viscido, S., Parrish, J., Grünbaum, D., 2005. The effect of population size and number of influential neighbors on the emergent properties of fish schools. Ecological Modelling 183, 347–363.

Viscido, S., Parrish, J., Grünbaum, D., 2007. Factors influencing the structure and maintenance of fish schools. Ecological Modelling 206, 153–165.

Vogel, S., 1994. Life in Moving Fluids: The Physical Biology of Flow. Princeton University Press, Princeton, NJ.

Wilson, W., 1996. Lotka's game in predator-prey theory: Linking populations to individuals. Journal of Theoretical Biology 50, 368–393.

Wilson, W., 1998. Resolving discrepancies between deterministic population models and individual-based simulations. The American Naturalist 151, 116–134.

Appendix A

Tritrophic Code

This appendix contains the computer code that was executed to assess the gregarious behaviour discussed in Chapter 3. The proceedures for executing R code and other documentation can be found at http://cran.r-project.org/. The code is executed by sourcing the file tritrophic.R (section A.1) in an R console. The file nncirc.f (section A.13) is compiled into an object file and then dynamically loaded using the file nncirc.R (section A.3)

A.1 tritrophic.R

The following code initializes the simulation environment and contains the loop that steps through the simulation.

```
source("hexgrid.R");
source("polycircle.R");
source("nncirc.R");
source("randpntscirc.R");
plot.size <- 6;
X11(width = plot.size, height = plot.size);
source("parameters.R");
source("initialize.R");
steps <- 1;
while((num.prey != 4 && num.pred != 2 ) && (num.plant != 2 || num.seed != 2)){
  source("nearestneighbours.R");
  source("updateprey.R");
  source("updatepredators.R");
```

```
  source("updateplants.R");
  source("birthsdeaths.R");
  source("plot.R");
  steps <- steps + 1;
}
dev.off()
```

A.2 hexgrid.R

The following code initializes the function that generates the locations of the plants and their seeds.

```
hexgrid <- function(island.radius,between.plant){
  hex.grid <- array(c(0,0),c(1,2))
  start <- array(c(0,0),c(1,2));
  left <- array(c(0,0),c(1,2));
  right <- array(c(0,0),c(1,2));
  for(i in 1:(island.radius/between.plant+5)){
    left[1,1] <- start[1,1] - i*between.plant;
    right[1,1] <- start[1,1] + i*between.plant;
    hex.grid <- rbind(hex.grid,left);
    hex.grid <- rbind(hex.grid,right);
  }
  start1 <- hex.grid;
  start2 <- hex.grid;
  start2[,1] <- start2[,1] + 0.5*between.plant;
  start2[,2] <- start2[,2] + sqrt(3)*between.plant/2;
  hex.grid <- rbind(start1,start2);
  up1 <- start1;
  down1 <- start1;
  up2 <- start2;
  down2 <- start2;
  for(j in 1:(ceiling((sqrt(3)/2)*(island.radius/between.plant))+5)){
    up1[,2] <- start1[,2] + j*sqrt(3)*between.plant;
    up2[,2] <- start2[,2] + j*sqrt(3)*between.plant;
    down1[,2] <- start1[,2] - j*sqrt(3)*between.plant;
    down2[,2] <- start2[,2] - j*sqrt(3)*between.plant;
    hex.grid <- rbind(hex.grid,up1,up2,down1,down2);
  }
  final.hex.grid <- c();
  for(i in 1:nrow(hex.grid)){
```

```
    if((hex.grid[i,1]^2+hex.grid[i,2]^2) <= (island.radius^2 + 10)){
      final.hex.grid <- rbind(final.hex.grid, hex.grid[i,])
    }
  }
  return(final.hex.grid);
}
```

A.3 nncirc.R

The following code dyanmically loads the object file that is a result of compiling nncirc.f (section A.13) and initializes the functions that search for nearest-neighbours.

```
dyn.load("nncirc.so");
nncircw <- function(p,r,v){
  n <- nrow(p)
  dirx <- rep(0,n)
  diry <- rep(0,n)
  dist <- rep(0,n)
  nni <- rep(0,n)
  nnlist <- .Fortran("nncircw",as.double(p[,1]),
    as.double(p[,2]),
    as.double(r),
    as.integer(n),
    as.double(v),
    as.double(dirx),
    as.double(diry),
    as.double(dist),
    as.integer(nni))
  list(dirx=nnlist[[6]],diry=nnlist[[7]],
    dist=nnlist[[8]],nni=nnlist[[9]])
}
nncircb <- function(p1,r1,p2,r2,v){
  n1 <- nrow(p1)
  n2 <- nrow(p2)
  dirx <- rep(0,n2)
  diry <- rep(0,n2)
  dist <- rep(0,n2)
  nni <- rep(0,n2)
  nnlist <- .Fortran("nncircb",as.double(p1[,1]),
    as.double(p1[,2]),
    as.double(r1),
```

```
    as.integer(n1),
    as.double(p2[,1]),
    as.double(p2[,2]),
    as.double(r2),
    as.integer(n2),
    as.double(v),
    as.double(dirx),
    as.double(diry),
    as.double(dist),
    as.integer(nni))
  list(dirx=nnlist[[10]],diry=nnlist[[11]],
    dist=nnlist[[12]],nni=nnlist[[13]])
}
```

A.4 randpntscirc.R

The following code initializes the functions that generates random positions within a circle.

```
randpntscirc <- function(radius,num_pnts){
  rand_pnts <- c()
  i <- 1;
  while(i <= num_pnts){
    temp_pnt <- array(c(0,0),c(1,2));
    temp_pnt[1,1] <- runif(1,-radius,radius)
    temp_pnt[1,2] <- runif(1,-radius,radius)
    if((temp_pnt[1,1]^2+temp_pnt[1,2]^2) <= (radius^2)){
      rand_pnts <- rbind(rand_pnts, temp_pnt);
      i <- i + 1;
    }
  }
  return(rand_pnts);
}
```

A.5 parameters.R

The following code initializes the parameter values used in the simulation.

```
delta.t <- 1;
relative.speed <- 1.375;
num.plant.radius <- 1;
```

```
space.step <- 0.25;
between.plant <- 10;
island.radius <- between.plant*num.plant.radius;
plant.growth <- (between.plant/2)/(75);
plant.density <- 10;
plant.weight.max <- plant.density*pi*(between.plant/2)^2;
plant.weight.min <- plant.weight.max/15;
plant.weight.repro <- plant.weight.max/3;
plant.age.max <- 10000;
plant.age.min <- 1;
prey.density <- 300;
prey.weight.max <- 100;
prey.weight.min <- prey.weight.max/4;
prey.age.max <- 600;
pred.density <- 3000;
pred.weight.max <- 1000;
pred.age.max <- 3000;
plant.attr.prey <- between.plant/2;
prey.attr.pred <- 4*between.plant;
pred.repel.prey <- between.plant/2;
prey.repel.prey <- 0.25;
prey.attr.prey <- between.plant/2;
pred.repel.pred <- 0;
```

A.6 initialize.R

The following code creates all the initial individuals for the simulation.

```
temp.plant <- hex_grid(island.radius,between.plant);
num.plant <- nrow(temp.plant);
plant <- array(1,c(num.plant,5));
plant[,1:2] <- temp.plant;
plant[,3] <- plant.weight.max;
plant[,4] <- round(runif(num.plant,1,plant.age.max/2));
plant[,5] <- sqrt(plant[,3]/(pi*plant.density));
num.prey <- 10*(num.plant);
prey <- array(1,c(num.prey,7));
prey[,1:2] <- rand_pnts_circ((island.radius+between.plant/2),num.prey);
theta <- runif(num.prey,0,2*pi);
prey[,3] <- cos(theta);
prey[,4] <- sin(theta);
```

```
prey[,5] <- round(runif(num.prey,prey.weight.max/2,prey.weight.max));
prey[,6] <- round(runif(num.prey,1,prey.age.max));
prey[,7] <- round(runif(num.prey,0,1));
num.pred <- 4*(num.plant);
pred <- array(1,c(num.pred,6));
pred[,1:2] <- rand_pnts_circ((island.radius+between.plant/2),num.pred);
theta <- runif(num.pred,0,2*pi);
pred[,3] <- cos(theta);
pred[,4] <- sin(theta);
pred[,5] <- round(runif(num.pred,pred.weight.max/2,pred.weight.max));
pred[,6] <- round(runif(num.pred,1,pred.age.max));
plant <- rbind(array(c(-10000,10000,10000,-10000,1,1,1,1,1,1),c(2,5)),plant);
num.plant <- num.plant + 2;
space <- array(c(10000,10000,-10000,-10000,1,1,1,1,1,1),c(2,5));
num.space <- 2;
seed <- array(c(10000,10000,-10000,-10000,1,1,1,1,1,1),c(2,5));
num.seed <- 2;
prey <- rbind(array(c(-10000,10000,10000,-10000,1,1,1,1,1,1,1,1,1,1),c(2,7)),prey);
num.prey <- num.prey + 2;
pred <- rbind(array(c(10000,10000,-10000,-10000,1,1,1,1,1,1,1,1),c(2,6)),pred);
num.pred <- num.pred + 2;
```

A.7 nearestneighbours.R

The following code produces the nearest neighbours within and between populations.

```
pred.near.prey <- nncircb(pred[,1:2],
  sqrt(pred[,5]/(pi*pred.density)),
  prey[,1:2],
  sqrt(prey[,5]/(pi*prey.density)),
  pred.repel.prey);
prey.near.pred <- nncircb(prey[,1:2],
  sqrt(prey[,5]/(pi*prey.density)),
  pred[,1:2],
  sqrt(pred[,5]/(pi*pred.density)),
  prey.attr.pred);
plant.near.prey <- nncircb(plant[,1:2],
  plant[,5],
  prey[,1:2],
  sqrt(prey[,5]/(pi*prey.density)),
  plant.attr.prey);
```

```
prey.near.prey <- nncircw(prey[,1:2],
  sqrt(prey[,5]/(pi*prey.density)),
  prey.attr.prey);
pred.near.pred <- nncircw(pred[,1:2],
  sqrt(pred[,5]/(pi*pred.density)),
  pred.repel.pred);
```

A.8 updateprey.R

The following code modifies the prey during the simulation.

```
j <- 3;
prey.eat.plant <- array(0,c(num.plant,1));
born.prey <- c(); dead.prey <- c();
prey.deaths <- 0; prey.births <- 0;
while(j <= num.prey){
  prey[j,6] <- prey[j,6] + delta.t;
  if(prey[j,5] >= prey.weight.max){
    prey.births <- prey.births + 1;
    prey[j,5] <- prey.weight.max/2;
    born.prey <- rbind(born.prey,
      c((prey[j,1]-space.step*prey[j,3]),
        (prey[j,2]-space.step*prey[j,4]),
        (-prey[j,3]),
        (-prey[j,4]),
        (prey.weight.max/2),
        delta.t,
        prey[j,7]));
  }
  if(prey[j,6] > prey.age.max || prey[j,5] < prey.weight.min){
    dead.prey <- rbind(dead.prey,j);
    prey.deaths <- prey.deaths + 1;
  }
  else{
    traject <- c(0,0);
    traject.prey <- c(0,0); traject.pred <- c(0,0); traject.plant <- c(0,0);
    move <- 0; dying <- 0; feed <- 0;
    attr.plant <- 0; repel.pred <- 0; repel.prey <- 0; attr.prey <- 0;
    if(pred.near.prey$dist[j] <= 0){dying <- 1}
    if(plant.near.prey$dist[j] <= 0){feed <- 1}
    if(plant.near.prey$dist[j] <= plant.attr.prey){attr.plant <- 1}
```

```
if(pred.near.prey$dist[j] <= pred.repel.prey){repel.pred <- 1}
if(prey.near.prey$dist[j] <= prey.repel.prey){repel.prey <- 1}
if(prey.near.prey$dist[j] <= prey.attr.prey){attr.prey <- 1}
if(dying == 1){
  move <- 0;
}
if(dying == 0 && repel.pred == 1){
  move <- 1;
  traject.pred[1] <- -pred.near.prey$dirx[j];
  traject.pred[2] <- -pred.near.prey$diry[j];
}
if(dying == 0 && feed == 1 && repel.pred == 0){
  move <- 0;
  prey[j,5] <- prey[j,5] + delta.t;
  prey.eat.plant[plant.near.prey$nni[j]] <-
    prey.eat.plant[plant.near.prey$nni[j]] + delta.t;
}
if(dying == 0 && feed == 0 && repel.pred == 0 && attr.plant == 1){
  move <- 1;
  traject.plant[1] <- plant.near.prey$dirx[j];
  traject.plant[2] <- plant.near.prey$diry[j];
}
if(dying == 0 && repel.prey == 1){
  move <- 1;
  if(is.na(prey.near.prey$dirx[j]) || is.na(prey.near.prey$diry[j])){
    traject.prey[1] <- prey[j,3];
    traject.prey[2] <- prey[j,4];
  }
  else{
    traject.prey[1] <- -prey.near.prey$dirx[j];
    traject.prey[2] <- -prey.near.prey$diry[j];
  }
}
if(dying == 0 && feed == 0 && repel.pred == 0 &&
  repel.prey == 0 && attr.plant == 0 && attr.prey == 1){
  move <- 1;
  if(prey[j,7] == 1){
    traject.prey[1] <- prey.near.prey$dirx[j];
    traject.prey[2] <- prey.near.prey$diry[j];
  }
}
```

```
    if(dying == 0 && feed == 0 && repel.pred == 0 &&
      repel.prey == 0 && attr.plant == 0 && attr.prey == 0){
      move <- 1;
    }
  }
  if(move == 1){
    traject[1] <- prey[j,3] + traject.prey[1] + traject.pred[1] + traject.plant[1];
    traject[2] <- prey[j,4] + traject.prey[2] + traject.pred[2] + traject.plant[2];
    traject[1] <- traject[1]/((traject[1]^2+traject[2]^2)^0.5)
    traject[2] <- traject[2]/((traject[1]^2+traject[2]^2)^0.5)
    prey[j,1] <- prey[j,1] + delta.t*space.step*traject[1];
    prey[j,2] <- prey[j,2] + delta.t*space.step*traject[2];
    prey[j,3] <- traject[1];
    prey[j,4] <- traject[2];
  }
  j <- j + 1;
}
```

A.9 updatepredators.R

The following code modifies the predators during the simulation.

```
i <- 3;
born.pred <- c(); dead.pred <- c();
pred.deaths <- 0; pred.births <- 0;
while(i <= num.pred){
  traject <- c(0,0); traject.pred <- c(0,0); traject.prey <- c(0,0);
  move <- 0; feed <- 0; attr <- 0; repel <- 0;
  pred[i,6] <- pred[i,6] + delta.t;
  if(pred[i,5] >= pred.weight.max){
    pred.births <- pred.births + 1;
    pred[i,5] <- pred.weight.max/2;
    born.pred <- rbind(born.pred,
      c((pred[i,1] - relative.speed*space.step*pred[i,3]),
      (pred[i,2] - relative.speed*space.step*pred[i,4]),
      (-pred[i,3]),
      (-pred[i,4]),
      (pred.weight.max/2),
      delta.t));
  }
  if(pred[i,6] > pred.age.max){
```

```
  dead.pred <- rbind(dead.pred,i);
  pred.deaths <- pred.deaths + 1;
}
else{
  if(prey.near.pred$dist[i] <= 0){feed <- 1}
  if(prey.near.pred$dist[i] <= prey.attr.pred){attr <- 1}
  if(pred.near.pred$dist[i] <= pred.repel.pred){repel <- 1}
  if(feed == 1){
    move <- 0;
    pred[i,5] <- pred[i,5] + delta.t;
    prey[prey.near.pred$nni[i],5] <- prey[prey.near.pred$nni[i],5] - delta.t;
  }
  if(attr == 1 && feed == 0){
    move <- 1;
    traject.prey[1] <- prey.near.pred$dirx[i];
    traject.prey[2] <- prey.near.pred$diry[i];
  }
  if(repel == 1){
    move <- 1;
    if(is.na(pred.near.pred$dirx[i]) || is.na(pred.near.pred$diry[i])){
      traject.pred[1] <- pred[i,3];
      traject.pred[2] <- pred[i,4];
    }
    else{
      traject.pred[1] <- -pred.near.pred$dirx[i];
      traject.pred[2] <- -pred.near.pred$diry[i];
    }
  }
  if(repel == 0 && move == 0 && feed == 0){
    move <- 1;
  }
}
if(move == 1){
  traject[1] <- pred[i,3] + traject.pred[1] + traject.prey[1];
  traject[2] <- pred[i,4] + traject.pred[2] + traject.prey[2];
  traject[1] <- traject[1]/((traject[1]^2+traject[2]^2)^0.5)
  traject[2] <- traject[2]/((traject[1]^2+traject[2]^2)^0.5)
  pred[i,1] <- pred[i,1] + delta.t*relative.speed*space.step*traject[1];
  pred[i,2] <- pred[i,2] + delta.t*relative.speed*space.step*traject[2];
  pred[i,3] <- traject[1];
  pred[i,4] <- traject[2];
```

```
  }
  i <- i + 1;
}
```

A.10 updateplants.R

The following code modifies the plants, seeds, and potential locations for the seeds and plants during the simulation.

```
k <- 3;
born.seed <- c(); dead.space <- c();
space.deaths <- 0; seed.births <- 0;
while(k <= num.space){
  plant.near.space <- c();
  num.plant.near.space <- 0
  for(i in 1:nrow(plant)){
    distsqr <- ((plant[i,1]-space[k,1])^2+(plant[i,2]-space[k,2])^2);
    if(distsqr <= ((between.plant)^2)){
      plant.near.space <- rbind(plant.near.space, plant[i,])
      num.plant.near.space <- num.plant.near.space + 1;
    }
  }
  if(num.plant.near.space != 0){
    if(min(plant.near.space[,3]) >= plant.weight.repro){
      seed.births <- seed.births + 1;
      space.deaths <- space.deaths + 1;
      dead.space <- rbind(dead.space,k);
      born.seed <- rbind(born.seed, space[k,]);
    }
  }
  k <- k + 1;
}
k <- 3;
born.plant <- c(); dead.seed <- c();
seed.deaths <- 0; plant.births <- 0;
while(k <= num.seed){
  seed[k,4] <- seed[k,4] + delta.t;
  seed.weight <- seed[k,3];
  seed.radius <- sqrt(seed[k,3]/(plant.density*pi));
  delta.seed.weight <- plant.density*pi*
    ((seed.radius + delta.t*plant.growth)^2 - seed.radius^2);
```

```
    seed[k,3] <- seed[k,3] + delta.seed.weight;
    seed[k,5] <- sqrt(seed[k,3]/(plant.density*pi))
    if(seed.weight >= plant.weight.min){
      plant.births <- plant.births + 1;
      seed.deaths <- seed.deaths + 1;
      dead.seed <- rbind(dead.seed,k);
      born.plant <- rbind(born.plant, seed[k,]);
    }
    k <- k + 1;
  }
  k <- 3;
  born.space <- c(); dead.plant <- c();
  plant.deaths <- 0; space.births <- 0;
  while(k <= num.plant){
    plant[k,4] <- plant[k,4] + delta.t;
    if(plant[k,3] < plant.weight.min || plant[k,4] > plant.age.max) {
      space.births <- space.births + 1;
      plant.deaths <- plant.deaths + 1;
      dead.plant <- rbind(dead.plant,k);
      born.space <- rbind(born.space,
        c(plant[k,1], plant[k,2], 1, delta.t,
          sqrt(plant[k,3]/(plant.density*pi))));
    }
    else{
      plant.radius <- plant[k,5];
      delta.plant.weight <- plant.density*pi*
        ((plant.radius + delta.t*plant.growth)^2 - plant.radius^2);
      plant[k,3] <- max(min((plant[k,3] + delta.plant.weight - prey.eat.plant[k]),
        plant.weight.max),0);
      plant[k,5] <- sqrt(plant[k,3]/(plant.density*pi))
    }
    k <- k + 1;
  }
```

A.11 birthsdeaths.R

The following code removes dead and adds newborn individuals during the simulation.

```
  num.pred <- num.pred - pred.deaths + pred.births;
  num.prey <- num.prey - prey.deaths + prey.births;
  num.space <- num.space - space.deaths + space.births;
```

```
num.seed <- num.seed - seed.deaths + seed.births;
num.plant <- num.plant - plant.deaths + plant.births;
if(prey.deaths != 0){prey <- prey[-dead.prey,]};
if(prey.births != 0){prey <- rbind(prey,born.prey)};
if(pred.deaths != 0){pred <- pred[-dead.pred,]};
if(pred.births != 0){pred <- rbind(pred,born.pred)};
if(plant.deaths != 0){plant <- plant[-dead.plant,]};
if(plant.births != 0){plant <- rbind(plant,born.plant)};
if(seed.deaths != 0){seed <- seed[-dead.seed,]};
if(seed.births != 0){seed <- rbind(seed,born.seed)};
if(space.deaths != 0){space <- space[-dead.space,]};
if(space.births != 0){space <- rbind(space,born.space)};
```

A.12 plot.R

The following code produces the plot of the simulation.

```
if(plotit == 1 && num.prey >=3 && num.pred >=3) {
  op <- par(mfrow = c(1, 1), mai = c(0,0,0,0));
  extent <- 2*island.radius;
  scale <- plot.size*11.5/extent
  plot(0, 0, col = "white", pch=16, cex = 1, xlab = "", ylab = "",
    xlim=c(-extent,extent), ylim=c(-extent,extent), axes = FALSE);
  points(plant[,1], plant[,2],
    col=rgb(round(255-200*((sqrt(plant[,3]/(plant.density*pi)))/plant[,5])),
      round(255-0*((sqrt(plant[,3]/(plant.density*pi)))/plant[,5])),
      round(255-200*((sqrt(plant[,3]/(plant.density*pi)))/plant[,5])), max = 255),
    pch=16, cex=scale*(plant[,5]));
  points(seed[,1], seed[,2],
    col="yellow",
    pch=16, cex=scale*(sqrt(seed[,3]/(plant.density*pi))));
  points(prey[,1], prey[,2],
    col=rgb(0, 0, round(255 - 250*(prey[,6]/prey.age.max)), max = 255),
    pch=(15*prey[,7]+1), cex=scale*(sqrt(prey[,5]/(prey.density*pi))));
  points(pred[,1], pred[,2],
    col=rgb(round(255 - 150*(pred[,6]/pred.age.max)),0,0,max=255),
    pch=16, cex=scale*(sqrt(pred[,5]/(pred.density*pi))));
}
```

A.13 nncirc.f

The following code initializes the nearest neighbour search algorithms and is compiled as an object.

```
subroutine nncircw(x,y,r,n,v,dirx,diry,dist,nni)
implicit real*8 (a-h,o-z)
dimension x(n),y(n),r(n)
dimension dirx(n),diry(n),dist(n),nni(n)
do i=1,n
  bmin=1d40
  xmin=1d40
  ymin=1d40
  nbr=0
  xi=x(i)
  yi=y(i)
  ri=r(i)
  do j=1,n
    if(i.ne.j)then
      rij=(r(j)+ri)
      xij=(x(j)-xi)
      if(abs(xij).le.(v+rij))then
        yij=(y(j)-yi)
        if(abs(yij).le.(v+rij))then
          cij=sqrt((xij)**2+(yij)**2)
          bij=(cij-rij)
          if(bij.lt.bmin)then
            bmin=bij
            xmin=(xij/cij)
            ymin=(yij/cij)
            nbr=j
          end if
        end if
      end if
    end if
  end do
  dirx(i)=xmin
  diry(i)=ymin
  dist(i)=bmin
  nni(i)=nbr
end do
```

```
      end
      subroutine nncircb(x1,y1,r1,n1,x2,y2,r2,n2,v,dirx,diry,dist,nni)
      implicit real*8 (a-h,o-z)
      dimension x1(n1),y1(n1),r1(n1),x2(n2),y2(n2),r2(n2)
      dimension dirx(n2),diry(n2),dist(n2),nni(n2)
      do i=1,n2
        bmin=1d40
        xmin=1d40
        ymin=1d40
        nbr=0
x2i=x2(i)
        y2i=y2(i)
r2i=r2(i)
        do j=1,n1
          rij=(r1(j)+r2i)
          xij=(x1(j)-x2i)
          if(abs(xij).le.(v+rij))then
            yij=(y1(j)-y2i)
            if(abs(yij).le.(v+rij))then
              cij=sqrt((xij)**2+(yij)**2)
              bij=(cij-(r1(j)+r2i))
              if(bij.lt.bmin)then
                bmin=bij
                xmin=(xij/cij)
                ymin=(yij/cij)
                nbr=j
              end if
            end if
          end if
        end do
        dirx(i)=xmin
        diry(i)=ymin
        dist(i)=bmin
        nni(i)=nbr
      end do
      end
      end
```

Appendix B

Schooling Code

This appendix contains the computer code that was executed to assess the schooling behaviour discussed in Chapter 4. The proceedures for compiling C code and other documentation can be found at http://gcc.gnu.org/. The following code is compiled using file Makefile (section B.4). This code requires the OpenGL libraries from http://www.opengl.org/resources/libraries/ and the random number generator code (rvgs.c, rvgs.h, rngs.c, rngs.h) from http://www.cs.wm.edu/˜va/software/park/.

B.1 schooling.c

The following code initializes the animation environment in OpenGL and contains the loop for the simulation.

```
#include "draw.h"
#include <string.h>
#include <unistd.h>
#include "/usr/include/gd.h"
int screen_width = 600;
int screen_height = 600;
void init(void){
  eco_time = 0.0;
  init_Ecosystem();
  ECO = nearest(ECO);
  glClearColor(255.0/255.0, 255.0/255.0, 255.0/255.0, 0.0);
  glClearAccum(0.0, 0.0, 0.0, 0.0);
  glViewport(0.0, 0.0, screen_width, screen_height);
```

```
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  gluPerspective(45.0f, (GLfloat)screen_width/(GLfloat)screen_height, 1.0f, 1000.0f);
}
void resize(int width, int height){
  screen_width = width;
  screen_height = height;
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
  glViewport(0.0, 0.0, screen_width, screen_height);
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  gluPerspective(45.0f, (GLfloat)screen_width/(GLfloat)screen_height, 1.0f, 1000.0f);
  glutPostRedisplay();
}
void initLight(void){
  GLfloat white[4] = { 1.0, 1.0, 1.0, 1.0 };
  GLfloat lightPos[4] = {0.0, 0.0, RADIUS, 0.0};
  GLfloat mat_ambient[] = { 1.0, 1.0, 1.0, 1.0 };
  glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
  glEnable(GL_LIGHTING);
  glEnable(GL_LIGHT0);
  glLightfv(GL_LIGHT0, GL_DIFFUSE, white);
  glLightfv(GL_LIGHT0, GL_POSITION, lightPos);
}
void display(void){
  double look_x, look_y, count;
  ECO = nearest(ECO);
  move_max = 0.0;
  look_x = 0.0; look_y = 0.0, count = 0.0;
  tempEco = ECO->head;
  while (tempEco != NULL){
     direct_Individual(tempEco);
     move_Individual(tempEco);
     count = count + 1.0;
     look_x = (look_x*(count - 1.0) + tempEco->x_pos)/count;
     look_y = (look_y*(count - 1.0) + tempEco->y_pos)/count;
     tempEco = tempEco->next;
  }
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
  glEnable(GL_DEPTH_TEST);
  glEnable(GL_CULL_FACE);
```

```
  glMatrixMode(GL_PROJECTION);
  glEnable(GL_BLEND);
  glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
  glMatrixMode(GL_MODELVIEW);
  glLoadIdentity();
  initLight();
  gluLookAt(look_x, look_y, look_radius, look_x, look_y, 0.0, 1.0, 1.0, 0.0);
  tempEco = ECO->head;
  while(tempEco != NULL){
    draw_Individual(tempEco);
    tempEco = tempEco->next;
  }
  draw_Grid();
  glFlush();
  glutSwapBuffers();
  ECO = clean(ECO);
  time_draw = time_draw + delta_t;
  eco_time = eco_time + delta_t;
  fflush(stdout);
  if(eco_time >= 5000.0){
    remove_Population(DEAD);
    remove_Population(ECO);
    free(DEAD);
    free(ECO);
    free(PREY);
    init();
  }
}
void mouse_menu(int option){
  switch(option){
    case 666:
      remove_Population(DEAD);
      remove_Population(ECO);
      free(DEAD);
      free(ECO);
      free(PREY);
      exit(0);
    case 333:
      eco_time = 1000000.0;
  }
  glutPostRedisplay();
```

```
}
int main(int argc, char **argv){
  PlantSeeds(-1); //random number seed
  look_phi = M_PI/2.0;
  look_theta = 0.0;
  look_radius = 150.0;
  glutInit(&argc, argv);
  glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEPTH | GLUT_ALPHA);
  glutInitWindowSize(screen_width,screen_height);
  glutInitWindowPosition(0,0);
  glutCreateWindow("SCHOOL");
  glutDisplayFunc(display);
  glutCreateMenu(mouse_menu);
  glutAddMenuEntry("RESET", 333);
  glutAddMenuEntry("QUIT", 666);
  glutAttachMenu(GLUT_RIGHT_BUTTON);
  glutIdleFunc(display);
  glutReshapeFunc(resize);
  init();
  glutMainLoop();
  return(0);
}
```

B.2 draw.c

The following code draws an individual and a grid.

```
#include "draw.h"
void draw_Individual(struct individual *ind){
    if(ind->pop == PREY){draw_Prey(ind);}
}
void draw_Prey(struct individual *ind){
  int a;
  int draw_zone = 0;
  double da;
  GLUquadricObj *qobj;
  x_p = ind->x_pos;
  y_p = ind->y_pos;
  z_p = 0.0;
  r = ind->size;
  x_v = ind->x_vel;
```

```
  y_v = ind->y_vel;
  z_v = 0.0;
  v = sqrt(x_v*x_v + y_v*y_v + z_v*z_v);
  GLfloat black[4] = { 0.0/255.0, 0.0/255.0, 0.0/255.0, 1.0};
  GLfloat white[4] = { 255.0/255.0, 255.0/255.0, 255.0/255.0, 1.0};
  if(x_v == 0.0){
    if(y_v < 0.0){
      phi = 90.0;
    }
    if(y_v > 0.0){
      phi = -90.0;
    }
  }
  if(x_v > 0.0){
    phi = atan(y_v/x_v)*180.0/M_PI;
  }
  if(x_v < 0.0){
    phi = 180.0 + atan(y_v/x_v)*180.0/M_PI;
  }
  glPushMatrix ();
  glTranslatef(x_p, y_p, z_p);
  glRotatef(phi, 0.0, 0.0, 1.0);
  qobj = gluNewQuadric();
  glTranslatef(r/2.0,0.0,0.0);
  glTranslatef(0.0,r/2.0,0.0);
  glMaterialfv(GL_FRONT, GL_DIFFUSE, white);
  gluPartialDisk(qobj, 0.0, r/4.0, 200, 4, 0.0, 360.0);
  glTranslatef(0.0,-r/1.0,0.0);
  glMaterialfv(GL_FRONT, GL_DIFFUSE, white);
  gluPartialDisk(qobj, 0.0, r/4.0, 200, 4, 0.0, 360.0);
  glTranslatef(0.0,r/2.0,0.0);
  glTranslatef(-r/2.0,0.0,0.0);
  glMaterialfv(GL_FRONT, GL_DIFFUSE, black);
  gluPartialDisk(qobj, 0.0, r, 200, 4, 0.0, 360.0);
  glPopMatrix ();
}
void draw_Grid(void){
  int a;
  double da;
  GLfloat black[4] = { 0.0/255.0, 0.0/255.0, 0.0/255.0, 1.0};
  glMaterialfv(GL_FRONT, GL_DIFFUSE, black);
```

```
    for(a = 0; a < 100; a++){
    da = a;
      glBegin(GL_LINES);
        glVertex3f(-30.0*RADIUS, -30.0*RADIUS + 60.0*da*RADIUS/50.0, 0.0);
        glVertex3f(30.0*RADIUS, -30.0*RADIUS + 60.0*da*RADIUS/50.0, 0.0);
      glEnd();
    }
    for(a = 0; a < 100; a++){
    da = a;
      glBegin(GL_LINES);
        glVertex3f(-30.0*RADIUS + 60.0*da*RADIUS/50.0, -30.0*RADIUS, 0.0);
        glVertex3f(-30.0*RADIUS + 60.0*da*RADIUS/50.0, 30.0*RADIUS, 0.0);
      glEnd();
    }
  }
```

B.3 draw.h

The following code declares the functions that draw the individuals and the background.

```
#include "move.h"
void draw_Individual(struct individual *ind);
void draw_Grid(void);
void draw_Prey(struct individual *ind);
```

B.4 Makefile

The following code is used by a compiler to link all the C code into a single executable file.

```
CC=gcc
CFLAGS1=
EXTRA_CFLAGS=-g

CFLAGS=$(CFLAGS1) $(EXTRA_CFLAGS)
LIBS= -O3 -lm -lglut -lGLU -lGL -lXmu -lX11 -lXi -lgd-lz

INCLUDES=
```

```
.c.o:
        $(CC) -c $(CFLAGS) $(INCLUDES) $<

OBJS=main.o draw.o move.o update.o structs.o parameters.o neighbour.o rngs.o rvgs.o


sort: $(OBJS)
        $(CC) -o school $(OBJS) $(LIBS)


clean:
      rm -f $(OBJS) school
```

B.5 move.c

The following code adjusts the acceleration of an individual in responce to its photosensor values and moves the individual using a Fourth-Order Runge-Kutta approximation. This code also contains the functions for a plastic collisions.

```
#include "move.h"
void direct_Individual(struct individual *ind){
  double traject[2];
  double traject_repel[2];
  double traject_neutral[2];
  double traject_attr[2];
  double tempdist, tempangle, ind_dir, di;
  int a;
  double da, decay;
  traject[0] = 0.0; traject_repel[0] = 0.0;
  traject_neutral[0] = 0.0; traject_attr[0] = 0.0;
  traject[1] = 0.0; traject_repel[1] = 0.0;
  traject_neutral[1] = 0.0; traject_attr[1] = 0.0;
  x_v = ind->x_vel;
  y_v = ind->y_vel;
  if(x_v == 0.0){
    if(y_v == 0.0){
      ind_dir = Uniform(0.0, 2*M_PI);
    }
    if(y_v < 0.0){
      ind_dir = 3.0*M_PI/2.0;
```

```
  }
  if(y_v > 0.0){
    ind_dir = M_PI/2.0;
  }
}
if(x_v > 0.0){
  ind_dir = atan(y_v/x_v);
}
if(x_v < 0.0){
  ind_dir = atan(y_v/x_v) - M_PI;
}
if(ind_dir < 0.0){ind_dir = ind_dir + 2*M_PI;}
decay = sight_decay;
for(a = 0; a < DEGREES_I; a++){
  if(ind->direction[a].sight_ind != NULL){
    if(ind->direction[a].sight_ind->pop == PREY){
      da = a;
      tempdist = ind->direction[a].sight_dist;
      tempangle = 2.0*da*M_PI/DEGREES_D - ind_dir;
      if(tempangle < 0.0){tempangle = tempangle + 2.0*M_PI;}
      if(tempangle <= ind->front_angle ||
        tempangle >= 2.0*M_PI - ind->front_angle){
        if(tempdist <= ind->front_repel){
          traject_repel[0] = traject_repel[0] +
            exp(-tempdist/decay)*1.0*cos(2.0*da*M_PI/DEGREES_D);
          traject_repel[1] = traject_repel[1] +
            exp(-tempdist/decay)*1.0*sin(2.0*da*M_PI/DEGREES_D);
        }else{
          if(tempdist <= ind->front_neutral){
            traject_neutral[0] = traject_neutral[0] +
              exp(-tempdist/decay)*cos(ind_dir);
            traject_neutral[1] = traject_neutral[1] +
              exp(-tempdist/decay)*sin(ind_dir);
          }else{
            if(tempdist <= ind->front_attract){
              traject_attr[0] = traject_attr[0] +
                exp(-tempdist/decay)*1.0*cos(2.0*da*M_PI/DEGREES_D);
              traject_attr[1] = traject_attr[1] +
                exp(-tempdist/decay)*1.0*sin(2.0*da*M_PI/DEGREES_D);
            }
          }
```

```
  }
}else{
  if(tempangle <= ind->side_angle ||
    tempangle >= 2.0*M_PI - ind->side_angle){
    if(tempdist <= ind->side_repel){
      traject_repel[0] = traject_repel[0] +
        exp(-tempdist/decay)*1.0*cos(2.0*da*M_PI/DEGREES_D);
      traject_repel[1] = traject_repel[1] +
        exp(-tempdist/decay)*1.0*sin(2.0*da*M_PI/DEGREES_D);
    }else{
      if(tempdist <= ind->side_neutral){
        traject_neutral[0] = traject_neutral[0] +
          exp(-tempdist/decay)*cos(ind_dir);
        traject_neutral[1] = traject_neutral[1] +
          exp(-tempdist/decay)*sin(ind_dir);
      }else{
        if(tempdist <= ind->side_attract){
          traject_attr[0] = traject_attr[0] +
            exp(-tempdist/decay)*1.0*cos(2.0*da*M_PI/DEGREES_D);
          traject_attr[1] = traject_attr[1] +
            exp(-tempdist/decay)*1.0*sin(2.0*da*M_PI/DEGREES_D);
        }
      }
    }
  }else{
    if(tempangle <= ind->back_angle ||
      tempangle >= 2.0*M_PI - ind->back_angle){
      if(tempdist <= ind->back_repel){
        traject_repel[0] = traject_repel[0] +
          exp(-tempdist/decay)*1.0*cos(2.0*da*M_PI/DEGREES_D);
        traject_repel[1] = traject_repel[1] +
          exp(-tempdist/decay)*1.0*sin(2.0*da*M_PI/DEGREES_D);
      }else{
        if(tempdist <= ind->back_neutral){
          traject_neutral[0] = traject_neutral[0] +
            exp(-tempdist/decay)*cos(ind_dir);
          traject_neutral[1] = traject_neutral[1] +
            exp(-tempdist/decay)*sin(ind_dir);
        }else{
          if(tempdist <= ind->back_attract){
            traject_attr[0] = traject_attr[0] +
```

```
                    exp(-tempdist/decay)*1.0*cos(2.0*da*M_PI/DEGREES_D);
                  traject_attr[1] = traject_attr[1] +
                    exp(-tempdist/decay)*1.0*sin(2.0*da*M_PI/DEGREES_D);
                }
              }
            }
          }
        }
      }
      if(tempdist <= 0.0){
        plastic_Collision(ind, ind->direction[a].sight_ind);
      }
    }
  }
}
  x_d = ind->x_dir; y_d = ind->y_dir;
  x_d = x_d + traject_neutral[0] + traject_attr[0] - traject_repel[0];
  y_d = y_d + traject_neutral[1] + traject_attr[1] - traject_repel[1];
  d = sqrt(x_d*x_d + y_d*y_d);
  ind->x_dir = x_d/d;
  ind->y_dir = y_d/d;
}
void plastic_Collision(struct individual *ind1, struct individual *ind2){
  double x_v1, y_v1, m1, x_v2, y_v2, m2;
  x_v1 = ind1->x_vel;
  y_v1 = ind1->y_vel;
  m1 = (4.0/3.0)*M_PI*pow(ind1->size,3);
  x_v2 = ind2->x_vel;
  y_v2 = ind2->y_vel;
  m2 = (4.0/3.0)*M_PI*pow(ind2->size,3);
  x_v = (x_v1*m1 + x_v2*m2)/(m1 + m2);
  y_v = (y_v1*m1 + y_v2*m2)/(m1 + m2);
  ind1->x_vel = x_v;
  ind1->y_vel = y_v;
  ind2->x_vel = x_v;
  ind2->y_vel = y_v;
}
void move_Individual(struct individual *ind){
  x_p = ind->x_pos;
  y_p = ind->y_pos;
  x_v = ind->x_vel;
```

```
  y_v = ind->y_vel;
  x_d = ind->x_dir;
  y_d = ind->y_dir;
  r = ind->size;
  c = 6.0*M_PI*viscosity*r;
  g = ind->speed*4.0*M_PI*r*r;
  w = (4.0/3.0)*M_PI*r*r*r;
  h = delta_t;
  ind->x_pos = x_p + h*x_v + h*h*(3.0 - (c/w)*h +
    (c/w)*(c/w)*h*h/4.0)*(g*x_d - (c/w)*x_v)/6.0;
  ind->y_pos = y_p + h*y_v + h*h*(3.0 - (c/w)*h +
    (c/w)*(c/w)*h*h/4.0)*(g*y_d - (c/w)*y_v)/6.0;
  ind->x_vel = x_v + h*(6.0 - 3.0*(c/w)*h +
    (c/w)*(c/w)*h*h - (c/w)*(c/w)*(c/w)*h*h*h/4.0)*(g*x_d - (c/w)*x_v)/6.0;
  ind->y_vel = y_v + h*(6.0 - 3.0*(c/w)*h +
    (c/w)*(c/w)*h*h - (c/w)*(c/w)*(c/w)*h*h*h/4.0)*(g*y_d - (c/w)*y_v)/6.0;
  x_d = ind->x_pos - x_p;
  y_d = ind->y_pos - y_p;
}
```

B.6 move.h

The following code declares the functions for the movement of individuals.

```
#include "neighbour.h"
void direct_Individual(struct individual *ind);
void plastic_Collision(struct individual *ind1, struct individual *ind2);
void move_Individual(struct individual *ind);
```

B.7 neighbour.c

The following code determines the values of the photo-sensors.

```
#include "neighbour.h"
struct ecosystem *nearest(struct ecosystem *eco) {
  struct individual *ind;
  struct individual *checkind;
  double test1, test2, test3, test4;
  double n_x_p, n_y_p, n_r;
  double da;
  int a;
```

```
min_dist = LARGE_NUMBER;
temp_dist = LARGE_NUMBER;
ind = eco->head;
while(ind != NULL){
  for(a = 0; a < DEGREES; a++){
    ind->direction[a].sight_ind = NULL;
    ind->direction[a].sight_dist = LARGE_NUMBER;
  }
  x_p = ind->x_pos;
  y_p = ind->y_pos;
  r = ind->size;
  checkind = eco->head;
  while(checkind != NULL){
    if(checkind != ind){
      n_x_p = checkind->x_pos;
      n_y_p = checkind->y_pos;
      n_r = checkind->size;
      if((pow(n_x_p-x_p,2) + pow(n_y_p-y_p,2)) <= pow(sight_dist,2)){
        for(a = 0; a < DEGREES_H; a++){
          da = a;
          test2 = ((n_x_p - x_p)*cos(2.0*M_PI*da/DEGREES_D) +
            (n_y_p - y_p)*sin(2.0*M_PI*da/DEGREES_D));
          test1 = pow(n_r,2) + pow(test2,2) -
            pow((n_x_p - x_p), 2) - pow((n_y_p - y_p), 2);
          if(test1 >= 0.0){
            test3 = test2 - sqrt(test1);
            test4 = test2 + sqrt(test1);
            if(test3 >= 0.0 && test4 > 0.0){
              if(test3 - r <= ind->direction[a].sight_dist){
                ind->direction[a].sight_ind = checkind;
                ind->direction[a].sight_dist = test3 - r;
              }
            }
            if(test3 < 0.0 && test4 > 0.0){
              if(-test3 < test4){
                if(-test3 - r <= ind->direction[a].sight_dist){
                  ind->direction[a].sight_ind = checkind;
                  ind->direction[a].sight_dist = test3 - r;
                }
              }else{
                if(test4 - r <= ind->direction[a + DEGREES_H].sight_dist){
```

```
                    ind->direction[a + DEGREES_H].sight_ind = checkind;
                    ind->direction[a + DEGREES_H].sight_dist = test4 - r;
                  }
                }
              }
              if(test3 < 0.0 && test4 <= 0.0){
                if(-test4 - r <= ind->direction[a + DEGREES_H].sight_dist){
                  ind->direction[a + DEGREES_H].sight_ind = checkind;
                  ind->direction[a + DEGREES_H].sight_dist = -test4 - r;
                }
              }
            }
          }
        }
      }
      checkind = checkind->next;
    }
    ind = ind->next;
  }
  return(eco);
}
```

B.8 neighbour.h

The following code declares the function that determines the values of the photosensors.

```
#include "update.h"
struct ecosystem *nearest(struct ecosystem *eco);
```

B.9 parameters.c

The following code initializes the parameter values used in the schooling simulation.

```
#include "parameters.h"
void init_Parameters(void){
  colour = 0.0;
  RADIUS = 100.0;
  LARGE_NUMBER = 1000000.0;
  sight_decay = 100.0;
  sight_dist = LARGE_NUMBER;
```

```
    front_angle = M_PI/3.0;
    side_angle = 3.0*M_PI/4.0;
    back_angle = 5.0*M_PI/6.0;
    front_repel = 5.0;
    front_neutral = 5.0;
    front_attract = LARGE_NUMBER;
    side_repel = 1.0;
    side_neutral = 50.0;
    side_attract = LARGE_NUMBER;
    back_repel = 2.0;
    back_neutral = 50.0;
    back_attract = LARGE_NUMBER;
    size = 1.0;
    speed = 1.0;
    viscosity = 2.0;
    double runge_kutta =
      (1.0/3.0)*(cbrt(4.0*sqrt(2349.0)+172.0)-cbrt(4.0*sqrt(2349.0)-172.0)+4.0);
    delta_t_stable = runge_kutta*(2.0/9.0)*(pow(size/cbrt(2.0),2)/viscosity);
    delta_t_move = (9.0/(16.0*M_PI))*(viscosity/speed)*(1.0/pow(size,3));
    if(delta_t_stable < delta_t_move){
      delta_t = delta_t_stable;
    }else{
      delta_t = delta_t_move;
    }
  }
```

B.10 parameters.h

The following code declares the function that initializes the parameters.

```
  #include "structs.h"
  void init_Parameters(void);
```

B.11 structs.c

The following code allocates the computer memory for the simulation and intializes the individuals.

```
  #include "structs.h"
  void malloc_Ecosystem(void){
    DEAD = (struct ecosystem *)malloc(sizeof(struct ecosystem));
```

```
  ECO = (struct ecosystem *)malloc(sizeof(struct ecosystem));
  PREY = (struct population *)malloc(sizeof(struct population));
}
void init_Ecosystem(void){
  init_Parameters();
  malloc_Ecosystem();
  ECO->head = NULL;
  ECO->tail = NULL;
  DEAD->head = NULL;
  DEAD->tail = NULL;
  PREY->head = NULL;
  PREY->tail = NULL;
  PREY->size = size;
  PREY->speed = speed;
  PREY->front_angle = front_angle;
  PREY->side_angle = side_angle;
  PREY->back_angle = back_angle;
  PREY->front_repel = front_repel;
  PREY->front_neutral = front_neutral;
  PREY->front_attract = front_attract;
  PREY->side_repel = side_repel;
  PREY->side_neutral = side_neutral;
  PREY->side_attract = side_attract;
  PREY->back_repel = back_repel;
  PREY->back_neutral = back_neutral;
  PREY->back_attract = back_attract;
  PREY->colour = colour;
  ECO = init_Population(PREY, ECO, 100);
}
struct ecosystem *init_Population(struct population *pop,
  struct ecosystem *eco, int number){
  struct individual *ind;
  int a;
  for(a = 0; a < number; a++){
    ind = init_Individual(pop, ind);
    eco = append_Individual(eco, pop, ind);
  }
  return(eco);
}
struct individual *malloc_Individual(struct individual *ind){
  if(DEAD->head != NULL){
```

```
    ind = DEAD->head;
    DEAD->head = DEAD->head->next;
    if(DEAD->head != NULL){
      DEAD->head->prev = NULL;
    }
  }else{
    ind = (struct individual *)malloc(sizeof(struct individual));
  }
  return(ind);
}
struct individual *init_Individual(struct population *pop, struct individual *ind){
  int a;
  ind = malloc_Individual(ind);
  ind->pop = pop;
  theta = Uniform(0.0,2.0*M_PI);
  phi = Random();
  ind->x_pos = RADIUS*(sqrt(phi))*cos(theta);
  ind->y_pos = RADIUS*(sqrt(phi))*sin(theta);
  theta = Uniform(0.0,2.0*M_PI);
  ind->x_dir = cos(theta);
  ind->y_dir = sin(theta);
  theta = Uniform(0.0,2.0*M_PI);
  ind->x_vel = cos(theta);
  ind->y_vel = sin(theta);
  ind->size = PREY->size;
  ind->speed = PREY->speed;
  ind->front_angle = PREY->front_angle;
  ind->side_angle = PREY->side_angle;
  ind->back_angle = PREY->back_angle;
  ind->front_repel = PREY->front_repel;
  ind->front_neutral = PREY->front_neutral;
  ind->front_attract = PREY->front_attract;
  ind->side_repel = PREY->side_repel;
  ind->side_neutral = PREY->side_neutral;
  ind->side_attract = PREY->side_attract;
  ind->back_repel = PREY->back_repel;
  ind->back_neutral = PREY->back_neutral;
  ind->back_attract = PREY->back_attract;
  ind->colour = Uniform(-10.0,10.0);//PREY->colour;
  for(a = 0; a < DEGREES_I; a++){
    ind->direction[a].sight_ind = NULL;
```

```
    ind->direction[a].sight_dist = LARGE_NUMBER;
  }
  ind->next = NULL;
  ind->prev = NULL;
  ind->next_pop = NULL;
  ind->prev_pop = NULL;
  return(ind);
}
struct ecosystem *append_Individual(struct ecosystem *eco,
  struct population *pop, struct individual *ind) {
  if(eco->head == NULL){
    eco->head = ind;
    ind->prev = NULL;
  }else{
    eco->tail->next = ind;
    ind->prev = eco->tail;
  }
  eco->tail = ind;
  ind->next = NULL;
  if(pop != NULL){
    if(pop->head == NULL){
      pop->head = ind;
      ind->prev_pop = NULL;
    }else{
      pop->tail->next_pop = ind;
      ind->prev_pop = pop->tail;
    }
    pop->tail = ind;
    ind->next_pop = NULL;
  }
  return(eco);
}
```

B.12 structs.h

The following code declares the functions that allocate memory; initialize the individuals; and declares the structures and variables used in the simulation.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
```

```
#include <unistd.h>
#include <sys/types.h>
#include <time.h>
#include <GL/glut.h>
#include <time.h>
#include <string.h>
#include "rngs.h"
#include "rvgs.h"
#ifndef M_PI
#define M_PI 3.14159265358979323846
#endif
#ifndef DEGREES
#define DEGREES 128
#endif
#ifndef DEGREES_I
#define DEGREES_I 128
#endif
#ifndef DEGREES_H
#define DEGREES_H 64
#endif
#ifndef DEGREES_D
#define DEGREES_D 128.0
#endif
double RADIUS;
double LARGE_NUMBER;
struct ecosystem{
  struct individual *head;
  struct individual *tail;
};
struct direction{
  struct individual *sight_ind;
  double sight_dist;
};
struct population{
  struct individual *head;
  struct individual *tail;
  double size, speed;
  double front_angle, side_angle, back_angle;
  double front_repel, front_neutral, front_attract;
  double side_repel, side_neutral, side_attract;
  double back_repel, back_neutral, back_attract;
```

```
  double colour;
};
struct individual{
  struct population *pop;
  double x_pos, y_pos;
  double x_dir, y_dir;
  double x_vel, y_vel;
  double size, speed;
  double front_angle, side_angle, back_angle;
  double front_repel, front_neutral, front_attract;
  double side_repel, side_neutral, side_attract;
  double back_repel, back_neutral, back_attract;
  double colour;
  struct direction direction[DEGREES];
  struct individual *next;
  struct individual *prev;
  struct individual *next_pop;
  struct individual *prev_pop;
};
struct ecosystem *DEAD;
struct ecosystem *ECO;
struct population *PREY;
struct individual *deadhead, *deadtail;
struct individual *Prey;
struct individual *tempPrey, *tempEco;
double size, speed;
double front_angle, side_angle, back_angle;
double front_repel, front_neutral, front_attract;
double side_repel, side_neutral, side_attract;
double back_repel, back_neutral, back_attract;
double colour;
int draw, jpegs;
double x_p, y_p, z_p, p;
double x_v, y_v, z_v, v;
double x_d, y_d, z_d, d;
double r, sum_r;
double theta, phi;
double delta_t, move_max;
double sight_dist, sight_decay;
double c, g, h, w;
double x_dist, y_dist;
```

```
double diff_x, diff_y;
double min_dist, temp_dist;
double angle, radius;
double eco_time;
double viscosity;
double delta_t_stable, delta_t_move;
double time_draw;
int time_jpegs, pause_eco;
double counter_jpegs;
double look_theta, look_phi, look_radius;
void malloc_Ecosystem(void);
void init_Ecosystem(void);
struct individual *malloc_Individual(struct individual *ind);
struct ecosystem *append_Individual(struct ecosystem *eco,
  struct population *pop, struct individual *ind);
struct individual *init_Individual(struct population *pop,
  struct individual *ind);
struct ecosystem *init_Population(struct population *pop,
  struct ecosystem *eco, int number);
```

B.13 update.c

The following code manages memory deallocation for the simulation.

```
#include "update.h"
struct ecosystem *clean(struct ecosystem *eco){
  struct individual *ind;
  int a;
  ind = eco->head;
  while(ind != NULL){
    for(a = 0; a < DEGREES_I; a++){
      ind->direction[a].sight_ind = NULL;
    }
    ind = ind->next;
  }
  return(eco);
}
struct individual *remove_Individual(struct ecosystem *eco, struct individual *ind) {
  struct individual *returnind;
  returnind = ind->next;
  if(ind->prev != NULL){
```

```
    ind->prev->next = ind->next;
  }else{
    eco->head = ind->next;
  }
  if(ind->next != NULL){
    ind->next->prev = ind->prev;
  }else{
    eco->tail = ind->prev;
  }
  if(ind->prev_pop != NULL){
    ind->prev_pop->next_pop = ind->next_pop;
  }else{
    ind->pop->head = ind->next_pop;
  }
  if(ind->next_pop != NULL){
    ind->next_pop->prev_pop = ind->prev_pop;
  }else{
    ind->pop->tail = ind->prev_pop;
  }
  if(eco != DEAD){
    DEAD = append_Individual(DEAD, NULL, ind);
  }else{
    free(ind);
  }
  return(returnind);
}
void remove_Population(struct ecosystem *eco){
  struct individual *checkind;
  int a;
  checkind = eco->head;
  while(checkind != NULL){
    for(a = 0; a < DEGREES_I; a++){
      checkind->direction[a].sight_ind = NULL;
    }
    checkind = checkind->next;
  }
  while(eco->head != NULL){
    remove_Individual(eco, eco->head);
  }
}
```

B.14 update.h

The following code declares the functions that deallocate memory for the simulation.

```
#include "parameters.h"
struct ecosystem *clean(struct ecosystem *eco);
struct individual *remove_Individual(struct ecosystem *eco, struct individual *ind);
void remove_Population(struct ecosystem *eco);
```