# Structured Population Models

## Applied to Predator Prey Population Dynamics

Karel de Wit
3694607

July 6, 2015

**Abstract**

Predator prey population dynamics can be modeled using a structure that is bought forth by characteristics of the individual. In this thesis we explore such individual state based models by both providing a theoretical background and by examining satiation and handling time based models. The former entails providing a framework underlying such models. The latter entails deriving properties of these models from mathematical derivations inspired by intuition from simulations. In doing so we clarify how one can use such models to describe and analyze a population dynamic both from a theoretical and practical point of view.

# Contents

# 1 Introduction

Ecosystems exist in all shapes and sizes. They can have just a few scarcely interacting organisms, but can also have a variety of organisms whose existence is tightly interwoven. The way in which they are interwoven can vary. Some organisms work together while others compete for the same source of food. Of our particular interest however are predator prey relations. In such relations predators are dependent on the availability of prey as a source of food. When there is an abundance of prey, predators can thrive as a result of the plentiful food availability. When prey are scarce, the capacity of this ecosystem is lower and thus the population size of predators is bounded.

Ecologists are often interested in modeling populations of predators and their prey. The way in which biological and environmental processes influence the size and structure of a population is called a population dynamic. When modeling a population dynamic one must therefore be able to determine the most important factors and incorporate these in the model. Predator prey relations are very diverse and for that reason formulating a model can be difficult. In general however, the predator and prey population sizes have a large influence on each other by both enabling and restricting growth. This influence is often exploited in models as it can adequately model a population dynamic. More advanced models however try to incorporate various different influences as well.

One such advanced way of modeling is by incorporating the structure of a population which is brought forth by characteristics of the individual. For example: a population of animals might be structured by their age, behaving differently as they get older. Populations whose structure seem to have an influence on a population dynamic can thus be described using a model that utilizes this underlying structure. These kinds of models are called structured population models. In this thesis we aim to explore structured population models to provide insight into the workings and the use of such models. We will explore structured population models as follows.

Chapter 2 is centered around the theoretical background of structured population models. We will explain how individual characteristics can be used to model an entire population. Then a theoretical framework underlying structured population dynamics is constructed. This is of our interest from both a theoretical and a practical point of view, as it enables us to construct custom models and helps us understand examples of structured population models. In chapter 3 and 4 we put theory to practice as we will discuss

examples of structured population models based on satiation and handling time. This is done with mathematical derivations and simulation techniques. Finally in the chapter 5 we will compare these examples and show how they are and how they are not similar. This does not only enable us to deepen our understanding of the specific models we have chosen to elaborate on, but also deepens our understanding of structured population models in general.

Our thesis is highly influenced by the following book written by S. Levin: *Lecture Notes in Biomathematics: The Dynamics of Physiologically Structured Populations*. This book arose out of a colloquium on the dynamics of structured populations and further expands on the idea's that where brought forth in this colloquium[3]. Despite the already extensive examination of structured population models by this book, we've been able to elaborate and extend upon some of the theory and models proposed in this book. This means that some results are based on or taken from the book, but most of the results and all of the simulations (and techniques) are our own results. In particular, all the theory, results and off course the simulations from chapters 4 and onward are entirely conceived by us.

# 2 Modeling structured population dynamics

When modeling a population dynamic, ecologists try to utilize properties of the populations and its dynamic to describe the changes of the populations over time. With structured population models, the dynamic can be modeled by making use of the underlying structures that are inherent to populations. We could for instance structure a population by the level of satiation of the individuals. It is reasonable to assume that the satiation of an individual affects how much hunger it has which in turn affects it's willingness to hunt for prey. However, to be able to quantify this relation, we must turn to a formal notion of what a structured population model is. This will be introduced in this chapter.

We have organized this chapter as follows: we will explicate the basic components of structured population models in the first section of this chapter. In the second section we will introduce the famous 'law of mass action' which will prove to be useful throughout this thesis. Then in the third section we formulate how changes in populations are generated. This differential equation based generator is mostly based on theory from Levin's book [3], in particular on chapter 3. In the fourth section we will formulate boundary conditions. Finally in the final section we will put this all together and propose a four step plan you can follow to construct your own structured population model.

## 2.1 Modeling structure

In this first section we will clarify why and how we structure populations and we take the first steps in utilizing this structure for modeling a population dynamic.

### 2.1.1 The individual as the basic unit

A population is defined by Odum [5] as "a collective group of organisms of the same species". Consequently the basic unit of a population is the individual. When modeling structured population dynamics it therefore reasonable to keep track of this basic unit. However, McCauley, Wilson and De Roos describe in their article [4] that ecologist often need to question population models used in practice because they tend to be overly simplistic. Models often disregard the complexity of individual behavior by using a top down

approach. This means that they model the entire population as a whole and derive individual behavior from this model. Individuals however have their own traits, such as genotype, age or size and this influences their behavior. It is for that reason that we acknowledge the complex mechanisms on the individual level and instead of ignoring these mechanisms, we try to build our model from the individual up. In doing so we recognize the inherent structure of populations and use it in a bottom up approach of modeling.

### 2.1.2 The states, state spaces and p-equations

Describing the structure of a population is done by formulation properties of the individual and using these properties to structure a population.

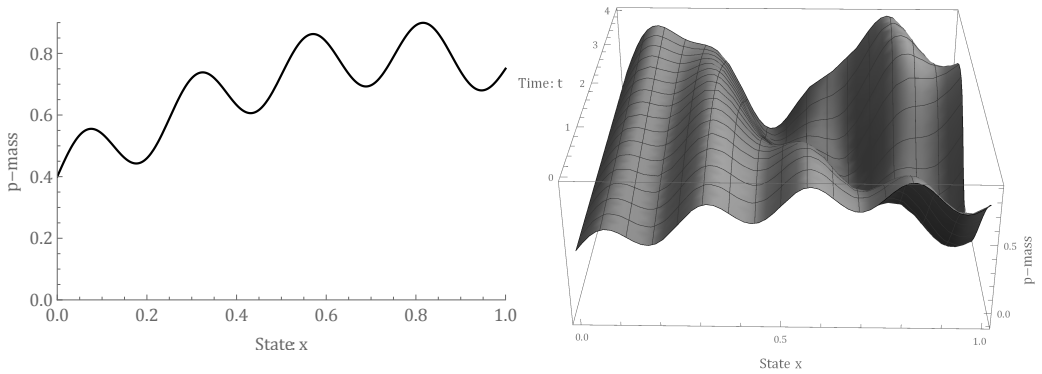
The individual is modeled by quantifying the properties that affect its population dynamic related behavior. These properties are combined in what we call the individual state or i-state. Examples of i-state variables are age, size or satiation. In general, we will denote the i-state as $x$ with $x = (x_1, \cdots, x_k)$, where all $x_i$ with $i \in \{1, 2, \cdots, k\}$ are i-state characteristics. The space of possible i-states $(x)$ is called the state space and this is denoted as $\Omega$. When for instance an organism has an i-state defined as it's size, which can range from 1 to 50, then $\Omega = [1, 50]$.

The population is then modeled by using this state space $\Omega$, since all individuals must by definition be characterized by a point in $\Omega$. The population can be described by the frequency distribution of i-states along $\Omega$. This is called the distribution, population state or p-state.

$$\text{p-state:} x \mapsto n(t, x)$$

Finally, we need to model change in this population over time. After initialization, future p-states are brought forth by p-equations. These are differential equations that describe the change of the p-state as a function of i-state variables. These functions of i-state variables are in turn determined by i-state based behavior.

Lets illustrate these definitions with an example. Say we can structure a population of predators according to their age as we observe that older predators catch more prey. This hunting efficiency is quantified with an increasing function $f(x)$. Our i-state is therefore $x$ (age) and our initial p-state is the amount of predators with age $x$ for all ages in $\Omega$. Changes in this p-state $p(x, t)$ are brought forth by the p-equations. For instance:




(a) p-state at initialization $t = 0$

(b) p-state changing over time.

Figure 2.1.1: Example of a p-state as a function of i-state $x$ and time $t$.

$\frac{\partial p(x,t)}{\partial t}$ = [animals getting older] + [caught prey]$(f(x))$. An example of a p-state and changes of a p-state over time is provided in figure 2.1.1. Notice how the initialized p-state in 2.1.1a is precisely equal to 2.1.1b for $t = 0$.

Together this information enables us to model our population over time as it produces all the information we desire: population sizes and distributions. If we so desire, we can zoom in on the individual level and deduce the exact amount of organisms with a specific i-state at a specific time. This way we capture the overall properties of a population dynamic without disregarding the complexity of an individual.

It is important to point out that in this thesis we will only discuss models and techniques based on ordinary differential equations and partial differential equations (ODE's and PDE's). Using DE's to model population dynamics is common practice in ecology, but it is not the only practice. Stochastic and agent based models are becoming more popular and could be used for structured population models as well. However, this thesis is not about describing similar dynamics with different techniques, but about describing different dynamics with similar techniques. For this purpose, using only ODE's and PDE's is reasonable. For this reason our p-equations are always differential equations.

The p-equation we just provided is obviously not specific enough to model change. How to formulate the p-equation is discussed in section 2.3. Before we can do this, we first need to acquire knowledge about the mass action law.

## 2.2 The law of mass action

Population dynamics affect the p-states of predators and prey. The total amount of organisms, regardless of their i-state, is what we call the population size. By using a structured population model, we can keep track of the number of animals with a specific i-state as well. This local population 'size' is called the mass or p-mass. A law that is very useful when modeling (changes of) mass is the law of mass action. This law originates from the field of chemistry where it describes how a chemical reaction takes place under ideal conditions. Say $A$ and $B$ are concentrations of two reacting substances to form substance $C$. The law states that:

$$\text{Production of C} = k \cdot A \cdot B \tag{2.2.1}$$

where $k$ is a rate constant describing the speed of the reaction. In this reaction A and B typically decrease until this reaction comes to an end. This formula implies that the production of $C$ is proportional to the amount of particles of both $A$ and $B$. The intuition behind this is fact that the more particles there are, the bigger the change they will collide and react. Even though this law is very straight forward and simple, it enables us to to describe complex dynamics [6].

This law isn't only applicable to chemistry. It is applied in many different fields of science, among which is ecology. In this field the law describes the dynamics of two interacting species as a product of the two population sizes and a constant. The reasoning behind this generalization is the fact that the likeliness of animals interacting is brought forth by the same type of dynamic as two substances reacting. Say $A$ and $B$ are now population sizes of two interacting species. If on one hand these population sizes are large, then surely, animals will come across each other more often and therefore interact more often. If on the other hand $A$ and $B$ are small, then they will not come across each other nearly as often, and therefore less interactions will occur. So if a quantity $C$ results from interactions of $A$ and $B$, then it will abide by the law of mass action.

With regards to ecology, the law isn't restricted to the form of equation (2.2.1). Examples of the uses of the mass action law taken from this article

and Levin's book [3] are:

$$\alpha n \tag{2.2.2a}$$

$$\int \lambda(x,y)n(y)dy \tag{2.2.2b}$$

$$\int \alpha_0(x,y)m(y)dy \cdot n(x) \tag{2.2.2c}$$

Example (2.2.2a) describes the number of births (or deaths) in a population. This is usually proportional to the population size, since if there are more individuals then there are more individuals that have a chance of dying. Again, this is not a one-to-one application of the mass action law as proposed in equation (2.2.1), but a generalization based on the same principle.

Example (2.2.2b) describes the (local) birth rate. This example is more similar to example (2.2.2a) than one might expect. The term $\lambda(x,y)$ describes the proportion (which now depends on $x$ and $y$) and this is multiplied by the population size. The mere fact that these number now depend on $x$ and $y$ and that they are integrated with respect to $y$ doesn't affect the underlying principle: a rate times the size; again an application of the mass action law.

Example (2.2.2c) describes the (local) disappearance rate due to predation. Again the same principle underlies this formula even though it is bi-linear.

All in all, the mass action law enables us to model various phenomena in ecology. We will use it throughout this thesis.

## 2.3 Modeling dynamics

Now that we are familiar with the law of mass action, we are able to describe how change occurs in terms of p-equations. These equations account for the changes in the p-state and thereby allow us to generate the p-state over time. Specifying the p-equations ultimately comes down to appropriately accounting for changes in p-mass. This means tracking all transitions within the i-state-space $\Omega$. We distinguish between two types of transitions. The first type is transport, also called flow or movement. This type of change is caused by individuals continuously flowing along $\Omega$. For example, if the i-state of an individual is his age, the individual's i-state age would continuously increase as time increased. The second type is jumps. This is an instantaneous change of p-mass at a specific i-state because of spontaneous

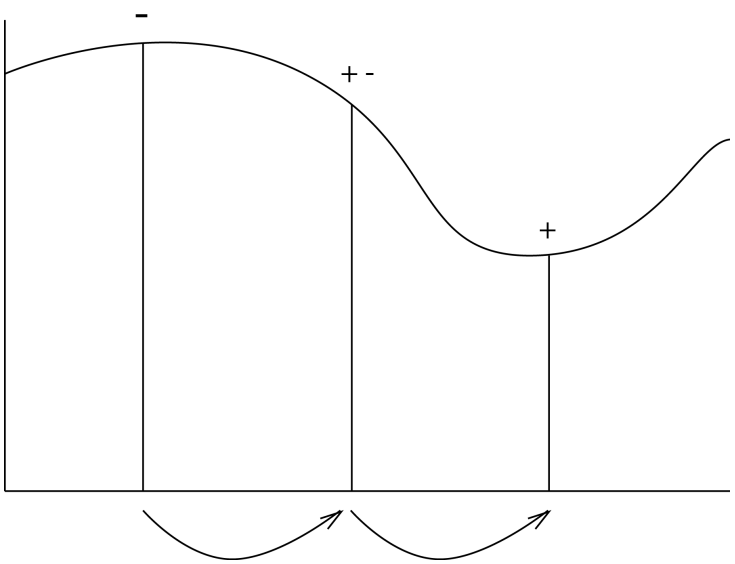



Figure 2.3.1: The effect of jumps on a p-state. The symbols $-$, $+-$ and $+$ indicate whether p-mass is lost, lost and gained or gained due to jumps from or to this i-state.

generation or disappearance, or a jump might be originating from somewhere else within $\Omega$. Examples are births, deaths and (instantaneous) ingestion of prey. Internal jumps are also illustrated by figure 2.3.1

### 2.3.1 The general p-equation

As stated before, by using infinitesimally small time intervals we can independently model the birth, death and migration of the population, but also less random changes such as aging or the decrease of satiation. We can therefore simply define a differential generator as the sum of these transitions. Hence we have the following framework for p-equations of structured population models:

$$\frac{\partial n}{\partial t} = \delta_t n - \delta_{s-} n + \delta_{s+} n \tag{2.3.1}$$

where the following parts of the equation are contributions to local change due to:

$$\begin{aligned}
\delta_t n &:= \text{continuous deterministic i-state movement (t stands for transport)} \\
\delta_{s-} n &:= \text{deaths or jumps to elsewhere in } \Omega \text{ (s stands for sink)} \\
\delta_{s+} n &:= \text{births or jumps from elsewhere in } \Omega \text{ (s stands for source)}
\end{aligned}$$

This generator describes all possible transitions of p-mass. We will elaborate on the three terms after introducing some notation.

### 2.3.2 Notation

To start off we introduce notation surrounding p-mass:

$$n(x)\Pi dx_i := \text{number of individuals in infinitesimal rectangle } \Pi(x_i, x_i + dx_i) \tag{2.3.2}$$

We can use $n(x)\Pi dx_i$ to approximate the p-mass of a region. Next we need to define the speed:

$$v(x) := \text{speed of individuals with i-state } x$$

The speed represents the rate at which flow occurs. The flow needn't occur at a constant rate. The speed of this flow can depend on the i-state. The speed function quantifies this relation.

### 2.3.3 Mass transport due to continuous i-state movement: $\delta_t n$

Decrease of satiation over time is an example of individuals flowing through i-states. To formulate such flow, we will formulate changes within $\Omega$ for infinitesimally small patches of i-state. We shall examine the box with edge $x$ and $x + e^{(i)}dx_i$ and determine the change of volume in this box over time. This provides us with change of p-mass due to flow.

As defined before, flow of i-state occurs with speed $v(x)$. The effect of transportation should therefore be applied to this box. We get:

$$\begin{aligned}
x \overset{v}{\mapsto} x + \text{change due to v} &= x + d^v x \\
&= x + \frac{dx}{dt} dt \\
&= x + v(x)dt
\end{aligned} \tag{2.3.3}$$

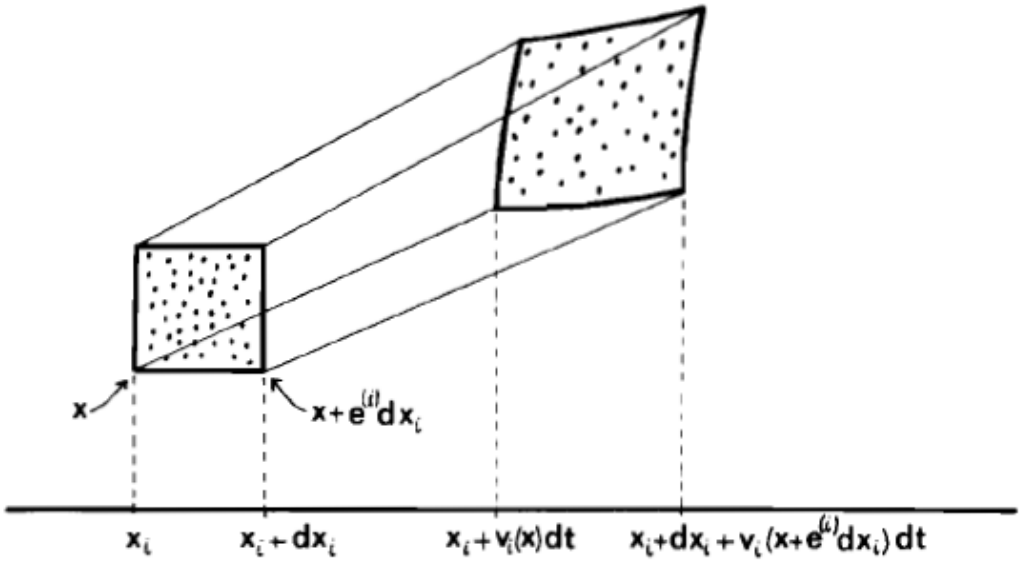



Figure 2.3.2: Changes of an infinitesimal volume of p-mass when it flows through $\Omega$. Graphic taken from Levin page 93 [3].

similarly we get:

$$\begin{aligned} x + e^{(i)} \overset{v}{\mapsto} & \; x + e^{(i)}dx_i + v(x + e^{(i)}dx_i)dt \\ & = x + e^{(i)}dx_i + \big(v(x) + v(e^{(i)}dx_i)\big)\, dt \\ & = x + e^{(i)}dx_i + \left(v(x) + \frac{\partial v}{\partial x_i}dx_i\right) dt \end{aligned} \tag{2.3.4}$$

where $e^{(i)}$ describes a change of the i-state in one of the $k$ dimensions. This change is also illustrated by figure 2.3.2. To model the change in p-mass, we need to model the change in volume that occurs because of this map. We can do this by comparing the points before and after the map and determine the change in volume in the $i^{th}$-direction.

$$\begin{aligned} \text{Volume before} &= (x_i + dx_i) - x_i = dx_i \\ \text{Volume after} &= \left(x + e^{(i)}dx_i + \left(v(x) + \frac{\partial v}{\partial x_i}dx_i\right) dt\right) - \big(x + e^{(i)}dx_i\big) \\ &= \left(e^{(i)} + \frac{\partial v}{\partial x_i}dt\right) dx_i \end{aligned} \tag{2.3.5}$$

If we single out only the $i$-th coordinate we find:

$$\begin{aligned}\text{Volume after} &= \left(1 + \frac{\partial v_i}{\partial x_i} dt\right) dx_i \\ &= \left(1 + \frac{\partial v_i}{\partial x_i} dt\right) \cdot [\text{volume before}] \end{aligned} \tag{2.3.6}$$

Clearly, the volume increases with the factor $\frac{\partial v_i}{\partial x_i} dt$. If we ignore higher order terms, we can add up the contributions of the volume changes in the various directions $i$ to arrive at a total relative volume change of $\sum_i \frac{\partial v_i}{\partial x_i} dt$. The change of p-mass because of i-state movement is therefore:

$$n(t + dt, x + v\, dt) \Pi dx_i \left(1 + \sum \frac{\partial v_i}{\partial x_i} dt\right) = n(t, x) \Pi dx_i \tag{2.3.7}$$

We will now rewrite (2.3.7) to arrive at $\delta_t n$.

$$\begin{aligned} n(t + dt, x + v\, dt) \left(1 + \sum \frac{\partial v_i}{\partial x_i} dt\right) - n(t, x) &= 0 \\ n(t + dt, x + v\, dt) + n(t + dt, x + v\, dt) \sum \frac{\partial v_i}{\partial x_i} dt - n(t, x) &= 0 \\ n(t + dt, x + v\, dt) + [-n(t, x + v\, dt) + n(t, x + v\, dt)] - n(t, x) + & \\ n(t + dt, x + v\, dt) \sum \frac{\partial v_i}{\partial x_i} dt &= 0 \\ [n(t + dt, x + v\, dt) - n(t, x + v\, dt)] + [n(t, x + v\, dt) - n(t, x))] + & \\ n(t + dt, x + v\, dt) \sum \frac{\partial v_i}{\partial x_i} dt &= 0 \end{aligned}$$

Dividing by $dt$ finally gives

$$\frac{\partial n}{\partial t} + \sum \frac{\partial n}{\partial x_i} v_i + n \sum \frac{\partial v_i}{\partial x_i} = 0$$

Hence:

$$\delta_t n = -\sum \frac{\partial n}{\partial x_i} v_i - n \sum \frac{\partial v_i}{\partial x_i} \tag{2.3.8}$$

which is the desired result: the change of p-mass caused by continuous i-state movement. The two terms of this expression also have a different notation

we will use from now on instead.

$$\text{Dilation} := n \sum \frac{\partial v_i}{\partial x_i} = n \bigtriangledown .v$$
$$\text{Convection} := n \sum \frac{\partial v_i}{\partial x_i} = \bigtriangledown n.v$$

The dilation accounts for the contraction or expansion of the volume. The convection accounts for pure transportation of the individuals along the p-state.

### 2.3.4 Internal disappearance and appearance of p-mass

Changes in p-state are not only caused by movement through $\Omega$. P-mass can change due to disappearance or appearance of p-mass as well. The disappearance or loss of p-mass can be caused by death or emigration, but also by internal jumps to somewhere else in $\Omega$. For example in the satiation example, if at one time an animal were to predate a prey its satiation would (almost) instantly jump to a higher satiation. This means an instant loss of p-mass at the starting satiation and an instant increase of p-mass at the resulting satiation.

To quantify the disappearance due to deaths, emigration and internal jumps, we can use the law of mass action. We get:

$$\delta_{s-} n(x) = \alpha(x) n(x) \tag{2.3.9}$$

Here $\alpha$ is the sum of the different causes of loss of p-mass: $\alpha = \alpha_{jumps} + \alpha_{deaths} + \alpha_{emigration}$. In general we call both deaths and emigrations simply deaths.

Similar to the disappearance of p-mass, p-mass can also (re)appear. P-mass can appear because of births, but also because of internal jumps. In the satiation example a jump results in an increase of p-mass at the touch-down position.

The appearance of p-mass in quantified as:

$$\delta_{s+} n(x) = b(x) \tag{2.3.10}$$

Again a mass action law argument underlies the explicit expression of $b(x)$ since death and jumps are like births proportional to the p-mass. This relation however isn't explicitly quantified in $\delta_{s+} n(x)$ because $b(x)$ is dependent

of the disappearance term $\delta_{s-}n(x)$. For instance the appearance of p-mass due to internal jumps is equal to the amount of p-mass lost due to internal jumps. This means that $b(x)$ is a function of $\delta_{s-}n(x)$. The quantity $\delta_{s-}n(x)$ is however already based on the law of mass action and this term therefore shouldn't be multiplied with $n(x)$ again. The mass action law is therefore not explicated in $\delta_{s+}n(x)$. A more explicit formulation of $b(x)$ can be found in appendix A.1.

### 2.3.5 Putting together the general p-equation

Now we have all the required quantities we can complete our p-equation. This enable us to model change over time. We combine equations (2.3.1), (2.3.8), (2.3.9) and (2.3.10) to obtain:

$$\frac{\partial n}{\partial t} = - \bigtriangledown n.v - n \bigtriangledown .v - \alpha(x)n(x) + b(x) \tag{2.3.11}$$

To finalize our generalized model, we will now formulate boundary conditions.

## 2.4 Flow of p-mass through the boundary of $\Omega$

Thus far we have only considered change that occurred due to changes within $\Omega$. These changes can be accounted for by what we call the p-equations. Not all changes can however be tracked this way. P-mass can also flow through the boundary of $\Omega := \partial\Omega$. If p-mass flows out of $\Omega$, this p-mass is lost since it is only defined within $\Omega$. For example: say we track predators with a satiation of $\epsilon > 0$. If we wait long enough, the satiation might go below 0 because these predators didn't catch any prey. At this point the satiation will turn negative and the predators will disappear from our model because $\Omega$ doesn't allow a negative satiation. Conversely if p-mass flows into $\Omega$ there is an increase of the population size, because outside of $\Omega$ this p-mass was not taken into account. Since these changes occur only at the boundaries, they cannot be modeled using p-equations which account for all p-mass *within* $\Omega$. To model these kinds of changes, we need so called boundary conditions. We will now formulate these boundary conditions in general.

### 2.4.1 Disappearance of p-mass through $\partial\Omega$

We can model p-mass leaving the boundary by using flux. This describes the flow of a volume through a space. In our case, it describes the flow of

p-mass. More rigorously, the flux equals the speed times the local p-mass: $\sigma(x) = v(x)n(x)$. In particular we are interested in the flux through $\partial\Omega$. P-mass can only leave $\Omega$ when the local speed points out of $\Omega$. The part of the boundary where this is the case is called $\partial_-\Omega$. We are interested in the amount of mass lost because of this outward flow. Note that if the flux is almost parallel to the boundary then not a lot of mass is lost. Only a small part of the displacement is actually through the boundary of $\Omega$. If however the flux is directed orthogonal to the boundary, the flow out through $\partial_-\Omega$ equals the local flux. To account for this difference, we should take the dot product of the flux with the vector $\nu(x)$ orthogonal to $\partial\Omega$. Combining these finding, we get:

$$\text{local flux out of } \Omega \text{ at } x \in \partial_-\Omega = \nu(x) \cdot v(x)n(x) \tag{2.4.1}$$

such that the total mass leaving $\Omega$ is the following:

$$\text{local mass leaving } \Omega \text{ across } x \in \partial_-\Omega = \int_{\partial_-\Omega} \nu(x) \cdot v(x)n(x)d\sigma$$

Dealing with this loss can be done in several ways. In some cases the loss will be actual loss and interpreted as deaths or emigrations. In other cases it will result in a jump back to somewhere within $\Omega$. An animal could for instance behave only when some boundary value $x \in \Omega$ is reached and jump back to somewhere within $\Omega$ after reaching that state.

### 2.4.2 The appearance of p-mass through $\partial\Omega$

The appearance through the boundary is roughly equivalent to the disappearance of p-mass through the boundary. However in this case, the flow inwards can only happen in the part of the boundary where the speed vector points inwards. We will denote this as $\partial_+\Omega$. Now the arrival rate of mass should equal the inwards flux corrected for the negative normal $\nu(x)$ since it should now point inwards. We obtain:

$$\text{local flux into } \Omega \text{ at } x \in \partial_+\Omega = -\nu(x) \cdot v(x)n(x) \tag{2.4.2}$$

and

$$\text{local mass entering } \Omega \text{ across } x \in \partial_+\Omega = \int_{\partial_+\Omega} -\nu(x) \cdot v(x)n(x)d\sigma$$

Together, the formulation of a population as proposed in 2.1, the general p-equation and the boundary conditions we just proposed allow us to construct a structured population model of our own. The steps are reiterated in the following section.

## 2.5 Summary: the basic components of a structured population model

In summary, we can define a structured population model in four steps.

1. The states and states spaces need to be defined. This entails choosing characteristics of the individual that determine population related behavior and combining these characteristics in the i-state. Then the allowed values of $x$ should be determined to construct the i-state space $\Omega$.

2. The p-equations need to be constructed. These account for internal transitions of p-mass. They should satisfy equation (2.3.11).

3. The boundary conditions need to be formulated. These account for the flow into and out of $\Omega$ through $\partial\Omega$. They should satisfy equations (2.4.1) or (2.4.2).

4. The population should be initialized. An actual initialization is only necessary when we want to simulate the model. In the case of a model description, we only need to formulate how this model should be initialized.

In the following chapters we will introduce examples of structured population models that will be set up by using these steps. The steps won't always be stated explicitly. Also step 2 and 3 are often combined as they are generally strongly connected.

# 3 Univariate structured population models

In the previous chapter we formulated the structured population model and derived some general properties. In this chapter we examine actual structured population models like holling disk based models[2] to show how one can use structured population models in practice. This chapter consists of three sections. In the first section we'll make some assumptions and introduce some notation. In the two following sections we respectively introduce a handling time and satiation based model.

## 3.1 Assumptions and notation

Before diving into predator prey populations dynamics we will make three assumptions regarding the structure of such populations. On one hand these assumptions will allow us to simplify the problem which eases the process of formulating, simulating and analyzing the models. On the other hand they standardize the problem which enables us to compare the properties of the models.

First, we assume there are at most two types of predators. Predators that are busy searching for prey $P_0$ and predators that are busy handling prey $P_1$. Searching for prey means waiting and looking out for prey to hunt. Handling prey is the combination of hunting on a prey, eating a prey and defending a prey item against other predators. Most models we propose keep track of both of these subpopulations, but one will use only one of these types of states. Second, we assume there are no births and deaths. This means that p-mass is only transported (through jumps or flow), but not really lost or gained. Third, we will assume that the prey density is unaffected by the predation. It is therefore kept at the constant value $x$.

We will also introduce some general notation. The numbers 0 and 1 correspond to the subpopulations of respectively searching predators and predators busy handling prey. These will show up in many different ways: as subscript of functions, as subscript of population denotations, etc. Furthermore, a capital $P$ is used to denote an entire (sub)population. So for instance $P_0$ is used to denote the subpopulation of searching predators as a whole. Capital letters are also used to denote the quantity or size of this (sub)population. For example $P_1(t)$ denotes the total p-mass of the subpopulation of predators handling prey at time $t$. Finally, a lower case $p$ is used to describe the local p-mass at a specific i-state. For the sake of completeness:

$p_0(t, y)$ would denote the local p-mass at some i-state $y$ at time $t$.

## 3.2 Handling time based structured population model

The first model we will examine is a handling time based model. In this (type of) model we assume that the handling time is the characteristic that determines an individual's population related behavior and can therefore be used to structure a population. Here the handling time is the total amount of time it takes to handle a prey item. When formulating a handling time based model we should therefore be able to bookkeep the handling time that has passed so far as a sort of internal stopwatch and relate this to the chance of transitioning to searching for prey. This relation between the passed handling time and the occurrence of jumps will form the basis of our handling time based model.

We construct the model in the following way. The group of searching predators is assumed to have no structure and can thus simply be represented by its total p-mass. Searching predators sometimes spot a prey, and when they do they start handling it. For simplicity we will assume this takes a fixed time $b$[1], after which they jump back to searching again. To set up this model we will follow the steps described in section 2.5, then we will simulate the model and finally we will derive observed properties from the simulations mathematically as well.

### 3.2.1 Model formulation

The first step in formulating a structured population model is to define the states and state space. At any time $t$, let $P_0(t)$ be the number of predators searching for prey and let the number of predators handling prey be $P_1(t)$. Within the population of handling predators we can distinguish between the different individuals by considering the time already spent on handling prey. We shall bookkeep this time with $\tau$. This is our i-state. The total needed handling time is assumed to be constant: $b$. Therefore we have $0 \leq \tau \leq b$.

[1] In reality, predators tend to have a handling time that is strongly peaked around some fixed value.

So our state space is $\Omega = [0, b]$.[2] Now let $p_1(t, \tau)$ be such that:

$$P_1(t) = \int_0^b p_1(t, \tau) d\tau \tag{3.2.1}$$

or more generally, for internal clock times $\alpha$ and $\beta$, the number of predators with clock times between these times is equal to: $\int_\alpha^\beta p_1(t, \tau) d\tau$.

*Constructing the p-equations*
With the i-states and state space defined we will proceed with step 2: constructing the p-equation. As stated before, this boils down to appropriately accounting for changes in p-mass.[3] With this in mind note that:

$$p_1(t, \tau) = p_1(t - \Delta, \tau - \Delta) \tag{3.2.2}$$

Assuming a fixed handling time, handling prey in this model only means waiting $b$ long and finally returning to searching when $\tau = b$ has passed. This means that predators don't change along the way and are simply flowing through $\tau \in [0, b]$. Thus the number of predators handling prey at time $t$ with handling time $\tau$ must equal the number of predators handling prey at time $t - \Delta$ with handling time $\tau - \Delta$. To generalize this result, we see that:

$$p_1(t, b) = p_1(t - b, 0) \tag{3.2.3}$$

Again, p-mass simply moves along $\tau$, but doesn't change during its course. Therefore the inflow of $P_0$ at time $t$ equals the inflow of $P_1$ at time $t - b$. Figure 3.2.1 further illustrates this fact depicting this system as a conveyor belt. We will rewrite equation (3.2.2) to obtain a p-equation:

$$\begin{aligned} p_1(t, \tau) &= p_1(t - \Delta, \tau - \Delta) \\ p_1(t, \tau) - p_1(t - \Delta, \tau) &= p_1(t - \Delta, \tau - \Delta) - p_1(t - \Delta, \tau) \\ \Delta^{-1} \left[ p_1(t, \tau) - p_1(t - \Delta, \tau) \right] &= \Delta^{-1} \left[ p_1(t - \Delta, \tau - \Delta) - p_1(t - \Delta, \tau) \right] \end{aligned}$$

Now by taking the limit $\Delta \downarrow 0$ we obtain the following partial differential equation:

$$\frac{\partial p_1}{\partial t}(t, \tau) = -\frac{\partial p_1}{\partial \tau}(t, \tau) \tag{3.2.4}$$

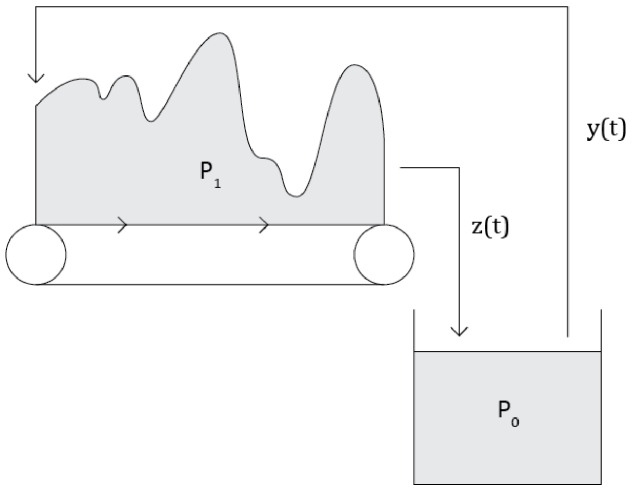



Figure 3.2.1: Illustration of the p-state transformation of an infinite state handling time model as a conveyor belt. The terms $y(t)$ and $z(t)$ are volumes of transitioning p-mass.

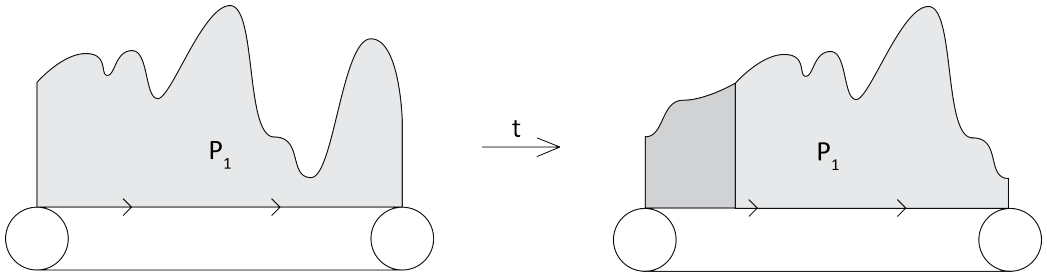



Figure 3.2.2: Transportation of p-mass across the conveyor belt.

*Formulating the boundary conditions*

Now in step 3 we will construct the boundary conditions. Since p-mass moves through $[0, b]$ without changing we can determine the p-mass at any $\tau$ by determining the corresponding inflow when $\tau$ was 0. If the inflow equals $y(t)$ which occurs at $p_1(t, 0)$, we get $p_1(t, 0) = y(t)$. Now what should this $y(t)$ be? Inflow of $P_1$ comes from an outflow of $P_0$. Since $P_0$ is a scalar, a

[2]Since we also have a state 'searching', our state space is actually $\Omega = [0, b] \cup searching$. For simplicity we will however usually refer to the simplified notion of $\Omega$ as being $[0, b]$

[3]The procedure used is very similar to the procedure used when we set up the general p-equation described in 2.3.3.

volume of p-mass, we can formulate the unknown function $y(t)$ by using the law of mass action. We get:

$$y(t) = axP_0(t) \tag{3.2.5}$$

where the prey density $x$ is assumed to be constant as stated before and $a$ is the rate constant of outflow. Lastly, we need to formulate the rates of change for $P_0$. Analogously to the previous steps, outflow out of $p_1(t, b)$ equals the inflow of $P_0$. So if $z(t) = p_1(t, b)$, then we combine this with (3.2.5):

$$\frac{dP_0}{dt} = -axP_0(t) + z(t) \tag{3.2.6}$$

*Initializing the system of population equations*
We've arrived at step 4. Equations (3.2.4), (3.2.5) en (3.2.6) together form our desired system of p-equations. If we provide initial conditions then our system is uniquely defined. These initial conditions are the following:

$$\begin{aligned} P_0(0) &= \Psi_0 \\ p_1(0, \tau) &= \psi_1(\tau) \end{aligned} \tag{3.2.7}$$

These equations together quantify the amount of predators in this system at the start. The quantity $\Psi_0$ is simply a number, the initial population size of $P_0$. The function $\psi_1(\tau)$ is the initial distribution of i-states in $\Omega = [0, b]$.[4]

### 3.2.2 Model simulation

To get an idea of the general characteristics of this model, we will run simulations to show how the populations behave over time. However, solving the system cannot (easily) be done analytically and we will therefore do numerical experiments using Mathematica.

We'll first rewrite the system to a recursion function. In order to do so we discretize our time in steps of $dt = 0.01$ and we also discretize the i-state space $\Omega$ into partitions of size $d\tau = 0.01$. This means that in our simulation, $P_1(t)$ is a list of partitions which is changed according to the derivative determined with steps of $dt = 0.01$. We will refer to the mass attributed to all values of the discretized $\Omega$ simply as $P_1(t)$. So we have $P_1(t) = (pmass_1, \ldots, pmass_{b/d\tau})$. Obviously $P_0(t)$ is still simply a number. Using this $\Omega$, we can recursively define the effect of an increase of $dt$ in time:

[4]From this point on the initialization is not explicated like this anymore, but is simply demonstrated in the simulation sections.

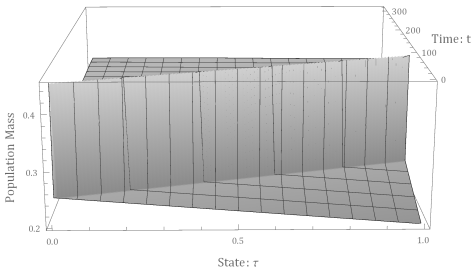
- $P_0(t) = P_0(t - dt) + \text{Dequeue}(P_1(t)) - ax \cdot dt \cdot P_0(t - dt)$
- $P_1(t) = \text{Enqueue}(\text{DeleteLast}(P_1(t))\ ,\ ax \cdot dt \cdot P_0(t - dt))$

Here Dequeue means, determine and take the last element from the queue $P_1(t)$, Enqueue means add a new first element to the queue (being $ax \cdot dt \cdot P_0(t - dt)$), and Delete-last means remove the last element. This way of simulating captures the conveyor belt like behavior. At the beginning mass is added, at the ending mass is removed, and everywhere in between mass is simply transported.

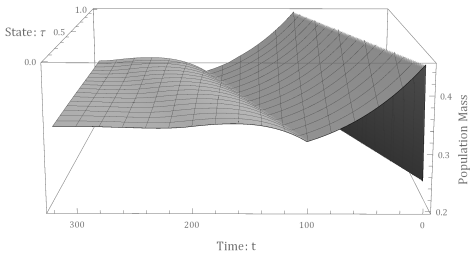
For the simulations we now only to quantify the initial conditions and the parameter values. We have chosen two initializations. Let $\Omega = [0, 1]$. Now for example 1 we have $P_0(0) = 50$ and $p_1(0, \tau) = 25 - 5\tau$ and for example 2 we have $P_0(0) = 30$ and $p_1(0, \tau) = 20 + 2\text{Sin}(10s)$. Furthermore, let $a = 0.09$ and $x = 10$. Now we can simulate this model to see the change of distribution over time. We programmed this simulation in Mathematica. The coding of this simulation can be found in appendix A.2.[5]

Running the simulation provided us with figure 3.2.3. The figure shows two examples of the change in distribution of $P_1$ and the total population size of subpopulation $P_0$. At $t = 0$, subfigures 3.2.3a and 3.2.3c show the initialization of p-mass in $P_1$. The change of the distribution from the initialization and onward can be seen in these figures for increasing values of $t$. Inspecting figure 3.2.3 enables us to make the following observations and claims about this model. First of all, the total population size does not change over time. If we look at subfigures 3.2.3e and 3.2.3f then all changes of the subpopulation sizes seem to cancel each other out. Second of all, the initial distribution of mass does not have an impact on the distribution on the long run. As soon as all the mass has jumped to $P_0$, all former structure is lost because searching predators spot prey at a rate that is independent of the time they have been searching up until that moment. So all predators have the same chance of spotting a prey item, regardless of the time they have been searching before. Third of all, we notice that the system stabilizes over time where $P_1$ converges to a uniform distribution and $P_0$ to a fixed value. Figures 3.2.3e and 3.2.3f show diverging population sizes for $t \in \Omega = [0, 1]$. Beyond $t = b$, convergence occurs rapidly. Fourth of all, the total handling time $b$ has a strong influence on the speed of convergence, because all initial

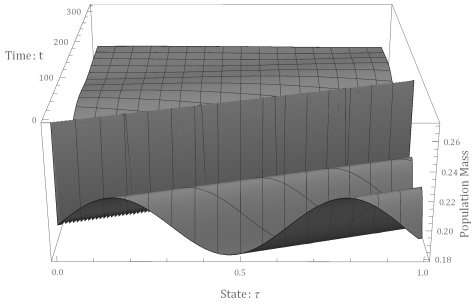
[5]Note that since we discretized time, a time of 100 corresponds to actual time $t = 100 * dt = 1$.



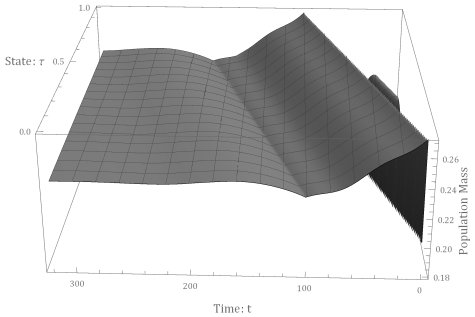

(a) Example 1, angle 1.



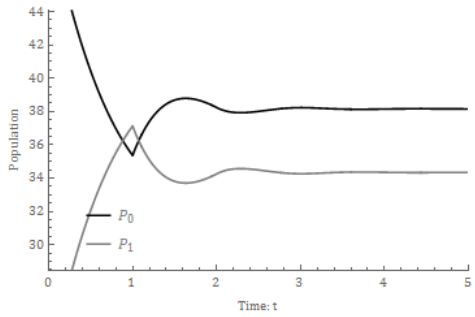

(b) Example 1, angle 2.



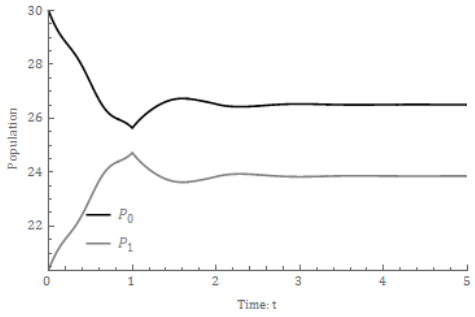

(c) Example 2, angle 1.




(d) Example 2, angle 2.




(e) Change of the total population sizes of $P_0$ and $P_1$ of example 1 over time.




(f) Change of the total population sizes of $P_0$ and $P_1$ of example 2 over time.

Figure 3.2.3: Change of population sizes and distributions of the infinite state handling time population model. Example 1 is initialized with $P_0(0) = 50$ and $p_1(0, \tau) = 25 - 5\tau$. Example 2 is initialized with $P_0(0) = 30$ and $p_1(0, \tau) = 20 + 2\text{Sin}(10s)$.

p-mass needs to transition to $P_0$ before this system can converge. After $b$ the convergence is rather 'bumpy' as it keeps overshooting the convergent value instead of converging from above or below only. This is caused by the fact that the convergence is delayed by the predators that are handling prey. The inflow of $P_0$ corresponds to the inflow of $P_1$ a time $b$ ago. An increase or decrease of $P_0$ results in a higher inflow only after a time $b$. Over time this bumpy convergence flattens out.

### 3.2.3 Mathematical derivation of properties

Some of the observations made by doing simulations can be derived analytically as well. We will first show that the total mass of the population ($P$) does not change. This means that $P$ does not change as a function of $t$. Under the assumption of differentiability, we should have that $\frac{d}{dt} \cdot (P_0(t) + \int_0^b p_1(t,\tau)d\tau) = 0$. We will therefore evaluate $\frac{d}{dt} \cdot (\int_0^b p_1(t,\tau)d\tau)$ and $\frac{dP_0}{dt}$. Let's start with the first expression. By substituting equation (3.2.4) into (3.2.1) we get:

$$\begin{aligned}
P_1(t,\tau) &= \int_0^b p_1(t,\tau)d\tau \\
\frac{dP_1(t,\tau)}{dt} &= \int_0^b \frac{\partial p_1}{\partial t}(t,\tau)d\tau \\
&= -\int_0^b \frac{\partial p_1}{\partial \tau}(t,\tau)d\tau \\
&= -\, p_1(t,\tau)|_0^b \\
&= p_1(t,0) - p_1(t,b) \\
&= y(t) - z(t) \qquad (3.2.8)
\end{aligned}$$

Now we combine (3.2.5), (3.2.6) and (3.2.8) to determine $\frac{\partial}{\partial t} \cdot (P_0(t) + \int_0^b p_1(t,\tau)d\tau)$. We obtain:

$$\begin{aligned}
\frac{dP}{dt} &= \frac{d}{dt} \cdot (P_0(t) + \int_0^b p_1(t,\tau)d\tau) \\
&= \frac{dP_0(t)}{dt} + \int_0^b \frac{\partial p_1}{\partial t}(t,\tau)d\tau \\
&= -y(t) + z(t) + y(t) - z(t) = 0
\end{aligned} \tag{3.2.9}$$

Next we will show that the model converges independent of the initial distribution. We can do this by determining the equilibrium and show what it does depend on to see that the initialization is not one of these dependents. In an equilibrium, the population size does not change over time. This means that in an equilibrium the derivatives with respect to time must equal zero. Then using (3.2.4) we find that in the equilibrium $\frac{\partial p_1}{\partial \tau}(t,\tau) = 0$. This implies that different values of $\tau$ are not associated with different p-masses. In other words, the distribution $p_1(t,\tau)$ converges to a uniform distribution with a local p-mass $\bar{p_1}$ that is independent of $\tau$. Equation (3.2.5) is now altered, as $y(t) = \bar{p_1}$. This leads to $\bar{p_1} = ax\bar{P}_0$.[6] Finally by using equation (3.2.9) we can combine the former equation with $b\bar{p_1} + \bar{P}_0 = P$. This gives our equilibrium value:

$$\bar{P}_0 = \frac{P}{1 + abx} \tag{3.2.10}$$

We can see that this equilibrium depends on the total population size and the quantities $a$, $b$ and $x$. So besides the initial population mass, the initial distribution of this mass itself along $P_0$ and $P_1$ has no influence on the final distribution. Using this equation we can calculate the equilibrium values of this model. The example of figure 3.2.3 has the following equilibrium:

$$\begin{aligned}
\bar{P}_0 &= \frac{10 + \int_0^b (10 - 5\tau)d\tau}{1 + 0.002 \cdot 1 \cdot 100} \\
\bar{P}_0 &= \frac{17.5}{1 + 0.2} = 14.5833 \\
\bar{P}_1 &= 17.5 - 14.5833 = 2.9167
\end{aligned}$$

[6]Note: equivalently we could have used equation (3.2.6) using the fact that all time derivatives equal zero and also $z(t) = \bar{p_1}$

## 3.3 Satiation based structured population model

Another interesting example of structured population models is a model using satiation as an i-state. The reason for looking at satiation is research conducted by Holling in 1959[2]. In his research he noticed that the lower the satiation of invertebrate predators, the lower their search rate. Hungry predators have a stronger incentive to go hunting and are therefore willing to put in more effort. Satiation as characteristic thus influences the population related behavior and is therefore a good basis for a structured population model.

We want the model to have the following structure. P-mass should decrease as a function of time, because digestion causes the satiation to decrease. Conversely, ingestion of prey causes the satiation to rise.

As before, we will first formulate the model, then simulate the model and finally derive (some of) its properties analytically. Note that from this section on, the steps needed to set up a structured population model as proposed in section 2.5 are not explicated anymore. The steps are however carried out in the same way as before.

### 3.3.1 Model formulation

Let the i-state, the satiation of an individual, be denoted as $s$. Then the corresponding i-state space is the range $(0, s_m]$ where $s_m$ is the highest satiation a predator can reach. In order to set up the p-equations, we need to make some assumptions and define a couple of functions.

We'll first formulate the search rate and the corresponding search function. The search rate quantifies the rate at which prey are spotted and the subsequent pursuits are started. Since the prey density $x$ is fixed we can quantify the search rate by using the mass action law: [prey capturing constant]$\cdot x$. We established that the prey capturing constant should depend on $s$. This is quantified with the search function $g(s)$. Hence the rate at which an individual predator captures prey is $xg(s)$. We assume that $g$ satisfies the following properties: $g$ is a continuous, decreasing function of $s$ and $g$ vanishes for $s \geq c$ for some $c > 0$. This $c$ is called the hunger threshold. Beyond this point predators will have no incentive to hunt at all. Furthermore, we assume $w$, the weight of a prey item, to be constant and to be small enough to not fill the gut (as Holling did in his research [2]).

We also need to quantify the rate of digestion. Note how this functions

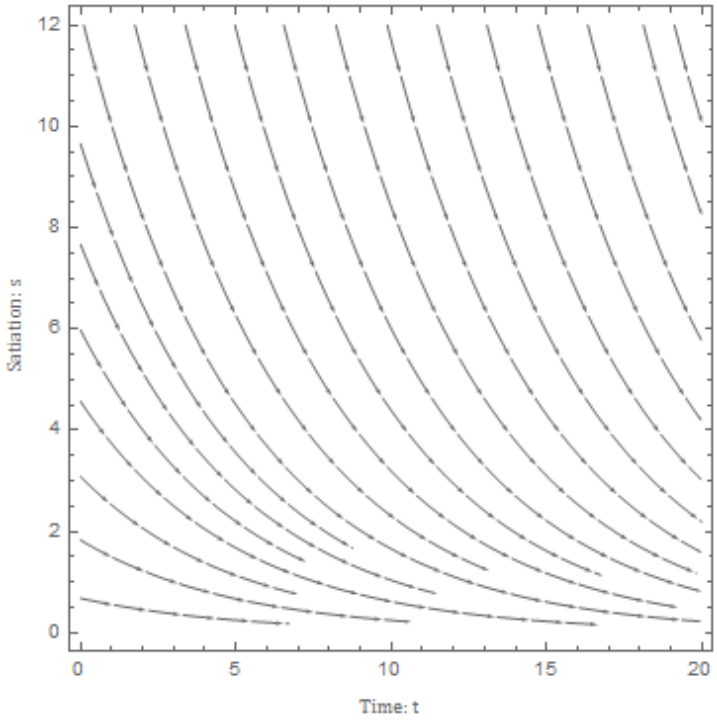



Figure 3.3.1: Stream diagram of the digestion function (3.3.1).

is equivalent to the speed function in the general framework described in chapter 2. If $a$ is the rate constant of digestion, then we have:

$$\frac{ds}{dt} = f(s) = -as \tag{3.3.1}$$

Furthermore, we'll assume the handling time to be 0.[7] The absence of handling time implies an upward jump in satiation brought forth by $xg(s)$ which will equal the size of the prey $w$. In between meals, the satiation will drop exponentially as a result of the exponential solution of differential equation $f(s)$.

*Setting up the p-equations*
With these assumptions and functions we can now formulate the p-equations.

[7]This is obviously in sharp contrast with our previous model which was purely based on the fact that handling time is not equal to zero in general. In chapter 4 we will propose a model that will incorporate both handling time and satiation.

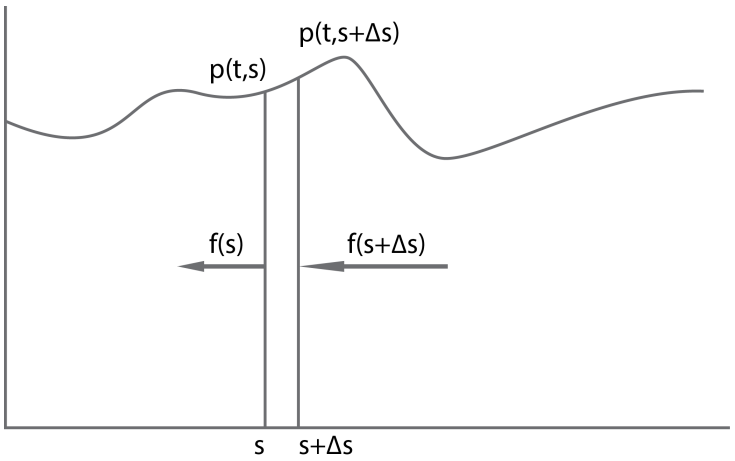



Figure 3.3.2: Change of p-mass in a strip $[s, s + \Delta s]$ due to digestion with speed $f(s)$.

Let $p$ be the density function of predators busy searching for prey which is defined for $\Omega = (0, s_m]$. Since we assume that the handling of prey is instantaneous, we only need to keep track of this single distribution of predators to describe the entire dynamic. The population mass of predators between two levels of satiation equals: $P(t, (u, v)) = \int_u^v p(t, s)ds$. This implies that the total population size $P(t)$ is:

$$P(t) = \int_{\Omega} p(t, s)ds \tag{3.3.2}$$

Predators flow through the i-states from right to left as digestion causes satiation to decreases over time. In other words: $f(s) \leq 0$. For now, we'll assume predators don't consume any prey after initialization. We'll get back on this later on. Consider the mass of a small strip of p-state, say $[s, s + \Delta_s]$. The mass of this strip by definition equals $P(t, (s, s + \Delta_s))$. We will account for the change of mass in this strip. We obtain the following relation:

$$\frac{dP(t, (s, s + \Delta_s))}{dt} = f(s)p(t, s) - f(s + \Delta_s)p(t, s + \Delta_s) \tag{3.3.3}$$

The first term of the RHS describes the rate at which mass leaves the strip: the local density (mass) times the speed at which this mass is leaving. Similarly, the second term of the RHS describes the rate at which mass enters

the strip from the right: the local density times the speed at which this mass is entering. Let us now rewrite the second term. We can approximate $f(s+\Delta_s)p(t,s+\Delta_s)$ with $f(s)p(t,s)$. Since these are very close, they are not likely to be very different. We can however approximate the difference they do have by using $\frac{\partial f(s)p(t,s)}{\partial s}\Delta_s$, the derivative in this point times the distance. Hence:

$$f(s+\Delta_s)p(t,s+\Delta_s) = f(s)p(t,s) + \frac{\partial f(s)p(t,s)}{\partial s}\Delta_s + o(\Delta_s)$$

We can also approximate $P(t,(s,s+\Delta_s))$ by multiplying $f(s)p(t,s)$ by the distance $\Delta_s$. Hence:

$$P(t,(s,s+\Delta_s)) = p(t,s)\Delta_s + o(\Delta_s)$$

Substituting these equations in (3.3.3), dividing by $\Delta_s$ and by letting $\Delta \downarrow 0$ results in our sought equation:

$$\frac{\partial p(t,s)}{\partial t} = -\frac{\partial}{\partial s}\left(f(s)p(t,s)\right) \tag{3.3.4}$$

We can rewrite this equation with the product-rule.[8] This will prove useful when simulating this model. We get:

$$\frac{\partial p(t,s)}{\partial t} = -f(s)\frac{\partial p(t,s)}{\partial s} - \frac{\partial f(s)}{\partial s}p(t,s) \tag{3.3.5}$$

Now that we have a way of describing the change caused by transportation along $\Omega$, we should relax our assumption that predators don't consume prey and incorporate this in our model. Ingesting prey with satiation $s$ will cause a jump to satiation $s+w$. We defined before that the rate constant of capturing prey is equal to $g(s)$ making $xg(s)$ the rate of capturing prey for a single predator. Again using the mass action law, we find that $xg(s)p(t,s)$ must then be the local amount of jumps away from $s$ as a result of catching prey by predators with i-state $s$. Using this relation, jumps onto $s$ can easily be derived as well. Jumps to $s$ must be caused by jumps coming from $s-w$. The appearance of p-mass at $s$ should then equal $xg(s-w)p(t,s-w)$. By combining the appearance and disappearance of p-mass into our previous equation 3.3.4, we get the following p-equation:

$$\frac{\partial p(t,s)}{\partial t} = -\frac{\partial}{\partial s}\left(f(s)p(t,s)\right) - xg(s)p(t,s) + xg(s-w)p(t,s-w) \tag{3.3.6}$$

[8]Note that the first term is the convection and that the second term is the dilation.

*Formulating the boundary conditions*
When initial distribution $p(0, s)$ is provided, this PDE can be numerically solved. We do however have to determine what happens at the boundaries of $\Omega$. We have two boundaries to discuss. If predators were to flow beyond the left boundary $s = 0$ because of digestion, we shall assume them to accumulate at $s = 0$. Similarly, if a predator where to jump beyond $s_m$ because of ingestion at a satiation of $s > s_m - w$, we will assume them to occur at $s_m$. This implies the following side condition.

$$-f(s_m)p(t, s_m) = \int_{s_m - w}^{s_m} xg(s)p(t, s)ds \tag{3.3.7}$$

With this formulation of the model we could make many generalizations. It might however be beneficial to first gain an intuition of how the model behaves. In order to obtain this intuition, we will use simulation to solve some examples.

### 3.3.2 Model simulation

As with the infinite state handling time model, we cannot (easily) solve this system analytically. We will therefore run numerical simulations to approximate it's behavior.

The simulation is performed as follows: we have rewritten the system to a recursion function and we use small time and satiation steps to determine all future states starting from our initialization. The time and satiation steps are $dt = ds = 0.01$. The recursion function isn't as straight forward as the recursion function of the infinite state handling time model. This is because equation (3.3.6) contains a term $\frac{\partial f(s)p(t,s)}{\partial s}$. In our previous simulation, we were able to divide the handling predators into parts of size $d\tau$. Like with a conveyor belt, an increase of time could simply be simulated by changing the first and last elements of our queue and moving all other elements one step further. This was possible because the speed did not change along $\Omega$. However, in the case of satiation, the flow isn't linearly proportional with time anymore so this conveyor belt has different speeds along $\Omega$. This can however be dealt with by simulating this system in a different way. Instead of using a conveyor belt like approach, we can approximate the local derivative $\frac{\partial f(s)p(t,s)}{\partial s}$ by using the definition of a derivative. So we approximate $\frac{\partial p(t,s)}{\partial s}$ with $\frac{p(t,s+ds)-p(t,s)}{ds}$ since our $ds$ is small. This is called a finite difference type

of approximation. A difficulty with this way of approximation is the fact that the derivative of the final partition has no (formal) successor within $\Omega$. However we can use the fact that all p-mass outside $\Omega$ is equal to 0. The final partition is therefore based on this value outside of $\Omega$. We obtain the following recursion function:

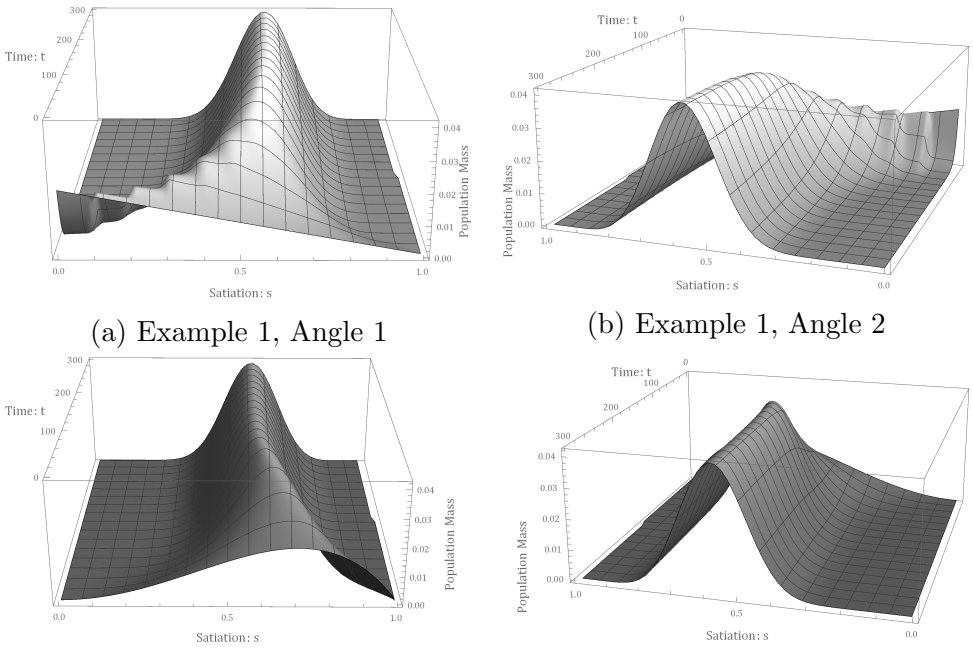
$$p(t,s) = p(t-dt,s) + dt\left(-f(s)\left(\frac{p(t-dt,s+ds)-p(t-dt,s)}{ds}\right)\right.$$
$$\left. - p(t-dt,s)f'(s) - xg(s)p(t-dt,s) + xg(s-w)p(t-dt,s-w)\right) \tag{3.3.8}$$

To apply this procedure, we first we need to set the parameters. We let digestion parameter $a = 0.5$, prey density $x = 10$ and prey size $w = 0.1$. Therefore $f(s) = -0.5s$. Furthermore, the satiation is $s \in \Omega = [0,1]$ with hunger threshold $c = 0.8$. This is combined in the search function:

$$g(s) = \begin{cases} 1 - 1.25s & : 0 < s < 0.8 \\ 0 & : \text{otherwise} \end{cases}$$

The basic shape of this linearly decreasing function is based on Hollings observations of actual predators[2]. Finally, we only need to decide on an initial distribution from which to start. We have chosen two initial distributions. Example 1: $p_{example\,1}(0,s) = 2 - 2s$ and example 2: $p_{example\,2}(0,s) = s^2 - s^4$. Using these initializations we have ran simulations in Mathematica. The relevant coding can be found in appendix A.2.

The results of the simulations are shown in figure 3.3.3. The figure shows the two examples of the satiation based model from two different angles. At $t = 0$ the initial distribution can be seen. Over time this distribution changes and this change is shown by plotting it against $t$. By closer inspection of the plots, some characteristics seem to apply to this model. First of all, the system smoothly converges to a peaked distribution with narrow tails. Second of all, the initial distribution has no impact on the shape of this equilibrium distribution. In both example 1 and example 2, the shape of the initial distribution is abandoned fast and both examples take on exactly the same shape in the long term. Last of all, the total p-mass does not appear to change.




(a) Example 1, Angle 1

(b) Example 1, Angle 2

(c) Example 2, Angle 1)

(d) Example 2, Angle 2

Figure 3.3.3: Change of the p-state of the satiation based structured population model over time. Example 1 is initialized as $p(s, 0) = 2 - 2s$, example 2 is initialized as $p(s, 0) = s^2 - s^4$.

### 3.3.3 Mathematical derivation of properties

The characteristics of the models we observed in the simulations can for a large part be derived analytically as well. In this section we will discuss and derive some of the characteristics of this model.

The first thing to notice is that when $s > c + w$, $p(t, s) = 0$. Predators can only jump from $s < c$ because the search rate of predators is zero at this point and beyond. Since all prey weigh $w$, jumps can only occur up to $s = c + w$. This leaves the part $[c + w, s_m]$ with only outflow, but no inflow. This causes all initial mass beyond $s = c + w$ to flow through $s = c + w$ in the long run, leaving the range $[c + w, s_m]$ empty.

We noticed before that the total number of predators is constant. This is caused by the fact that jumps and flow only occur within $\Omega$. We can also derive this property analytically. Let $P(t) = \int_0^{s_m} p(t, s)ds$. We will show that this total population size doesn't change by taking the derivative of the

total population size. If there is no change, than the derivative should be zero. We get:

$$P(t) = \int_0^{s_m} p(t,s)ds$$
$$\frac{dP(t)}{dt} = \int_0^{s_m} \frac{\partial p(t,s)}{\partial t} ds$$

Note that $f(s) \downarrow 0$ as $s \downarrow 0$. Thus p-mass can never reach $s = 0$ (which is also not within $\Omega$ and should as such equal 0). This indicates that $\lim_{s\downarrow 0} p(t,s) = 0$. Combining this observation with equations (3.3.6) and (3.3.7) we obtain the desired result:

$$\begin{aligned}
\frac{dP(t)}{dt} &= \int_0^{s_m} \left( -\frac{\partial}{\partial s}\left(f(s)p(t,s)\right) - xg(s)p(t,s) + xg(s-w)p(t,s-w) \right) ds \\
&= -\int_0^{s_m} \left( \frac{\partial}{\partial s}\left(f(s)p(t,s)\right) \right) ds - \int_0^{s_m} \left(xg(s)p(t,s)\right) ds + \\
&\quad \int_0^{s_m} \left(xg(s-w)p(t,s-w)\right) ds \\
&= -\int_0^{s_m} \left( \frac{\partial}{\partial s}\left(f(s)p(t,s)\right) \right) ds - \int_{s_m-w}^{s_m} \left(xg(s)p(t,s)\right) ds \\
&= -f(s)p(t,s)|_0^{s_m} + f(s_m)p(t,s_m) \\
&= -f(s_m)p(t,s_m) + f(s_m)p(t,s_m) \\
&= 0
\end{aligned} \tag{3.3.9}$$

Thus no change of the total population size occurs due to changes over time.

Like with the model before, this model also converges to a globally stable equilibrium. The proof of this statement goes beyond the scope of this thesis, but can be found in Heijmans [1]. However, we will show how to calculate the equilibrium. Since this problem isn't (easily) solved analytically, instead we will provide a numerical method. As we will see, the initialization has no effect on the convergent p-state as it doesn't show up in the procedure of determining the convergent distribution.

When determining the stable distribution, all derivatives with respect to time vanish. This implies we should solve the following equation:

$$-\frac{\partial}{\partial s}\left(f(s)p(s)\right) - xg(s)p(s) + xg(s-w)p(s-w) = 0 \tag{3.3.10}$$

Note that $t$ has also vanished as arguments of functions, since we assume $t$

no longer influences any of the functions. To approximate the solution, we shall discretize the state space $\Omega$ like we did when we simulated the model. This means we should partition $\Omega$ in points with size $ds$. Now choose one of such points $\psi$ between 0 and $w$ and assign to it the value $\theta$. So $p(\psi) = \theta$. Now we will use an Euler forward type of approach. For this method we need to obtain the derivative of $p(t, s)$ with respect to $s$. We rewrite (3.3.10):

$$\begin{aligned} f'(s)p(s) + f(s)p'(s) &= -xg(s)p(s) + xg(s-w)p(s-w) \\ p'(s) &= \frac{1}{f(s)} \cdot (-f'(s)p(s) - xg(s)p(s) + xg(s-w)p(s-w)) \end{aligned} \tag{3.3.11}$$

With equation (3.3.11) we can calculate points left from and right from our initialization. We can respectively use the following recursive formula's:

$$\begin{aligned} \text{points left from } \psi : p(s) &= p(s+ds) - ds \cdot p'(s+ds) \\ \text{points right from } \psi : p(s) &= p(s-ds) + ds \cdot p'(s-ds) \end{aligned} \tag{3.3.12}$$

Using these equations, first we will calculate all points in $(0, \psi]$, then we continue by calculating all points in $(\psi, s_m]$. The reason for first determining all points left from $w$ is that $xg(s-w)p(s-w) = 0$ for all $s < w$. So all terms of p'(s) are known for $s < w$. Beyond $s = w$ this term is not zero, but the values can then be obtained from the prior calculations. The resulting approximation $\overline{p(s)}$ shows how the p-masses relate between all values of $s$, but the total population size is incorrect because of our arbitrary initialization. This should be corrected. The convergent distribution $\widehat{p(s)}$ is determined as follows:

$$\widehat{p(s)} = \overline{p(s)} \cdot \frac{P}{\int_0^{s_m} \overline{p(s)} ds} \tag{3.3.13}$$

Figure 3.3.4 shows an approximation of $\overline{p(s)}$ for example 1 and 2 as formulated in figure 3.3.3. This was programmed using Mathematica. The coding can be found in appendix A.2.

Looking back on the procedure we just described, it becomes clear that the equilibrium $\widehat{p(s)}$ depends on $g(s)$, $x$ and $f(s)$. This implies that the equilibrium does not depend on the initial distribution. As long as our initial distribution meets some requirements regarding it's continuity and shape, it will always convert to the same final p-state. This corresponds to our observation from the simulation, where $g(s)$, $f(s)$ and $x$ were kept the same in both examples.

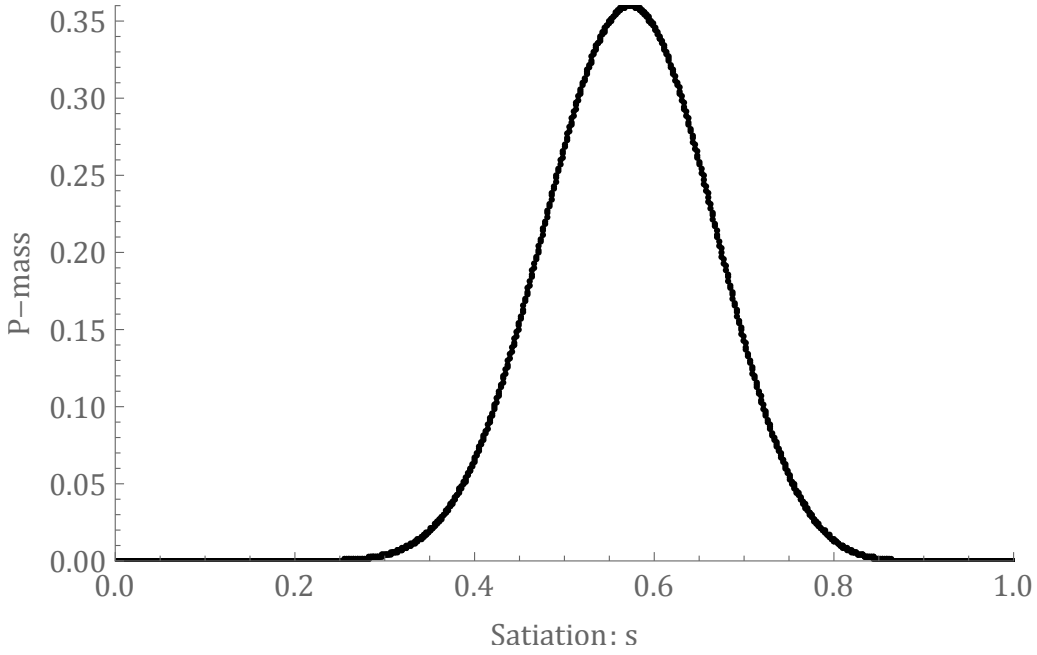



Figure 3.3.4: The approximation of $\overline{p(s)}$, the shape of the convergent distribution, for examples 1 and 2 of figure 3.3.3.

# 4 Multivariate structured population model

In chapter 3 we described two types of structured predator prey population models. The models were based on the assumption of either the satiation or the handling time being the determining factor of the population dynamic. However, there assumptions are very contradictory. In both models, the p-state was not influenced by the defining property of the other model. In the case of handling time, satiation was assumed to have no influence. In the case of satiation, handling time was (even explicitly) assumed to be zero. This means that ingestion of prey was assumed to be instantaneous.[9] If there are good reasons for both models to assume that their i-state determines the p-state over time, then incorporating both variables in an i-state could result in an even more comprehensive model. In this chapter we will therefore propose our own multivariate structured population model which combines the variables satiation and handling time in an i-state using the techniques described in chapter 2.

We will combine the satiation and handling time based models in such a way, that the original models are mostly unchanged. In the satiation based model we assumed handling time to be zero. This implied that immediately after spotting a prey item, it was ingested and had caused the satiation of the predator to increase according to the size of the prey item. But now if a prey item is spotted the predator jumps to a different subpopulation. In this subpopulation satiation increases over time as a result of ingestion. After completing the handling of a prey, it jumps back to the searching subpopulation of predators, but to its increased level of satiation.

As before, we will first formulate the model and then we will simulate it to assist us in mathematically deriving general properties of this model.

## 4.1 Model formulation

Let $P_0$ be the subpopulation of predators that are searching prey and let $P_1$ be the subpopulation of predators that are busy handling prey. Like before the prey density $x$ is assumed to be fixed. We will therefore only model the predator population dynamic. The i-state is defined as the combination of the level of satiation $s$ and the passed handling time $\tau$. Since we base this

[9] In the infinite handling time model, satiation was not explicitly assumed to have no influence, but not incorporating a variable in the model implies the assumption that this variable does not influence the p-state.

model on the univariate structured population models, we let $s \in (0, s_m]$ and $\tau \in [0, b]$ as before. Thus the satiation is bounded by $s_m$ and the handling of prey is a fixed set of events always taking exactly $b$ long. The state space is then $\Omega = (0, s_m] \times [0, \tau_m]$. The i-state variables aren't determining factors in both subpopulations of predators. In the case of handling prey $P_1$, handling time increases at a constant rate, and the satiation increases over time as well. Therefore, the p-mass of subpopulation 1, $p_1$, is a function of $s$ and $\tau$, thus $p_1(t, s, \tau)$. In the case of $P_0$ predators are looking for prey and must therefore have a $\tau = 0$ since they haven't spotted any prey to handle yet. We could therefore write $p_0$ as a function of $s$ and $\tau$ such that $p_0(t, s, 0)$, but instead we will simply denote the predators handling prey as $p_0(t, s)$.

The next step in defining our model is making some assumptions about the predator behavior based on its state. Like in the satiation model, we assume the satiation to decrease as a function of $s$. Therefore the satiation for searching predators decreases as follows: $\frac{ds}{dt} = f_0(s) = -as$. Now, when a prey item is spotted, it is eaten completely at constant rate $u > 0$. Thus when eating, the satiation of predators handling prey rises with speed $f_1 = f_0 + u$ or $f_1(s) = -as + u$. Note that the satiation is assumed to rise immediately after spotting a prey. This means that we assume that the time of pursuit of is negligible. We can also simply state that hunting is a part of searching prey. This means predators only start handling prey if they already caught it. The main reason for wanting to assume that the satiation rises for all $\tau \in [0, b]$ is because it simplifies our simulation without resulting in an unrealistic model. Especially considering the fact that our univariate satiation model assumed the entire handling time to be zero.

Finally we need to specify the rate at which jumps to handling prey occur. Again this is based on the same reasoning as the satiation based model. The search function $g(s)$ quantifies the rate at which prey is spotted which initiates a jump to $p_1$.

### 4.1.1 Setting up the p-equations

To account for the change in the population we will set op the p-equations. The p-equations of $p_0(t, s)$ and $p_1(s, \tau)$ are constructed separately using not only the assumptions and definitions of the previous section, but also using the results of setting up the p-equations of the related univariate population models.

For the predators searching prey $p_0(t, s)$ we can rewrite the satiation

based p-equation such that it can describe the behavior of this population dynamic. By using equation (3.3.6), we obtain the following:

$$\frac{\partial p_0(t,s)}{\partial t} = -\frac{\partial}{\partial s}\left(f_0(s)p_0(t,s)\right) - xg(s)p_0(t,s) + [\text{jumps back to searching}]$$

In words: satiation is still the only generating factor of population dynamic of searching predators. Also, the rate of disappearance is quantified using a law of mass action argument using the same $g(s)$ as with the satiation model. This leaves us with the appearance of mass. As a predator enters $P_1$, its satiation continuously rises resulting from a positive flow speed $f_1(s)$. When a predator is done handling prey at $\tau = b$, it will then jump back to $P_0$ with its increased satiation. Therefore, inflow of p-mass is brought forth by jumps from $p_1(t,s,b)$ at the same satiation $s$. We can thus rewrite our first p-equation.

$$\frac{\partial p_0(t,s)}{\partial t} = -\frac{\partial}{\partial s}\left(f_0(s)p_0(t,s)\right) - xg(s)p_0(t,s) + p_1(t,s,b) \tag{4.1.1}$$

Next we set up the p-equation of the predators handling prey $p_1(s,\tau)$. Again we will alter the p-equations brought forth by the associated univariate model, in particular equation (3.2.4). We obtain the following:

$$\frac{\partial p_1(t,s,\tau)}{\partial t} = -[\text{change due to digestion/ingestion}] - \frac{\partial p_1}{\partial \tau}(t,s,\tau)$$

Like before, the mass is transported along $\tau$ like a conveyor belt, however, now the mass flows towards a higher satiation as well. This change is generated by the flow function $f_1(s)$ defined in the previous section. Since this follows the same reasoning as the digestion function of the univariate satiation model, we obtain that this term must equal: $-\frac{\partial}{\partial s}\left(f_1(s)p_1(t,s,\tau)\right)$. Thus we obtain our next p-equation:

$$\frac{\partial p_1(t,s,\tau)}{\partial t} = -\frac{\partial}{\partial s}\left(f_1(s)p_1(t,s,\tau)\right) - \frac{\partial p_1}{\partial \tau}(t,s,\tau)$$

Together, this leads to the following two p-equations:

$$\begin{aligned}
\frac{\partial p_0(t,s)}{\partial t} &= -\frac{\partial}{\partial s}\left(f_0(s)p_0(t,s)\right) - xg(s)p_0(t,s) + p_1(t,s,b) \\
\frac{\partial p_1(t,s,\tau)}{\partial t} &= -\frac{\partial}{\partial s}\left(f_1(s)p_1(t,s,\tau)\right) - \frac{\partial p_1}{\partial \tau}(t,s,\tau)
\end{aligned} \tag{4.1.2}$$

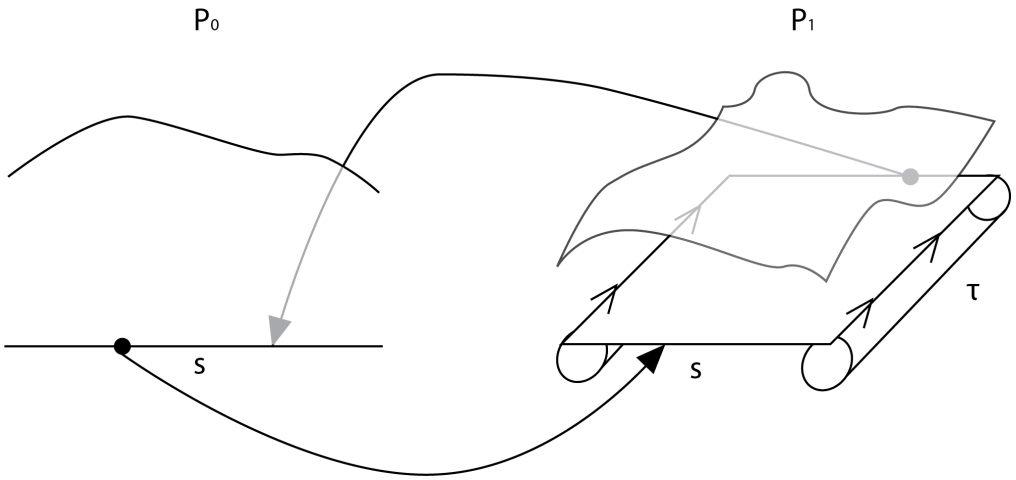



Figure 4.1.1: Illustration of transitions between $P_0$ and $P_1$ in the handling time and satiation based structured population model.

### 4.1.2 Formulating the boundary conditions

The last step of setting up this model is formulating the behavior at the boundaries. Firstly, we will describe the behavior at the boundaries of $P_0$. Predators that appear to flow through the left boundary $s = 0$ are assumed to accumulate at satiation $s = 0$. For the right boundary recall the reasoning behind (3.3.7). Here all mass that was to jump beyond $s_m$ was collected at $s_m$. Because now the inflow occurs from $p_1(t, s, b)$ where obviously $s \in (0, s_m]$ there is no more mass that can jump beyond $s_m$. This leaves us with the following boundary condition:

$$p_0(t, s_m) = 0 \tag{4.1.3}$$

Secondly, we will describe the behavior at the boundaries of $P_1$. Where $\tau = 0$, there is an inflow of p-mass from jumps out of $P_0$. These jumps occur at a rate $xg(s)p_0(t, s)$, therefore the inflow of predators is formulated as:

$$p_1(t, s, 0) = xg(s)p_0(t, s) \tag{4.1.4}$$

Similarly, we have an outflow of p-mass at $p_1(t, s, b)$ where mass jumps back to the searching subpopulation. We have already accounted for this change in the p-equation of $p_0(t, s)$. This leaves us with the two satiation related boundaries to consider. These will be dealt with analogously to the boundary

conditions of $P_0$. Predators that appear to flow through the left boundary $s = 0$ are assumed to accumulate at this satiation. At the right boundary we obtain the following equation again based on the same reasoning as equation (3.3.7):

$$p_1(t, s_m, \tau) = 0 \tag{4.1.5}$$

Changes due to transitions within and between $P_0$ and $P_1$ are also illustrated in figure 4.1.1.

## 4.2 Model simulation

Now that we have defined our model, we are interested in it's behavior in the short and long run. However, since this model doesn't seem fit for solving analytically we will approximate its change with simulations. We'll first initialize the parameters of the population dynamic.

Since we have barely any new parameters compared to the univariate satiation and handling time based models, we can simply copy the parameters chosen before. Some minor differences are the result of fine tuning the parameters to obtain clear results. The state space is based on the satiation $s \in (0, 1]$ and handling time $\tau \in [0, 1]$. This results in $\Omega = (0, 1] \times [0, 1]$. Furthermore, as we saw before the search function is:

$$g(s) = \begin{cases} 1 - 1.25s & : 0 < s < 0.8 \\ 0 & : \text{otherwise} \end{cases}$$

Moreover we choose the following parameters:

- Digestion parameter $a = 0.5$
- Prey density: $x = 10$
- Ingestion parameter $u = 0.5$

This results in the digestion and ingestion functions to take on the following form:

$$f_0(s) = -0.5s \qquad f_1(s) = -0.5s + 0.5$$

It is important to note that at lower satiations the flow is faster than at higher satiations. This will have an impact on the behavior of this population dynamic.

The simulation is executed by combining the techniques from the simulations of the univariate models. Subpopulation $P_0$ is modeled equivalently to the satiation based model and thus needs no further explanation. Subpopulation $P_1$ however is more complex as it depends on both $s$ and $\tau$. The change due to the preceding of time is: $\frac{\partial p_1}{\partial \tau}(t, s, \tau)$ which implies a conveyor belt like behavior described before in section 3.2. This means p-mass is simply shifted $d\tau$ along $\Omega$ at every time step. Since this rate is constant, we can simply model this with a queue. The change due to ingestion/digestion is then modeled using the finite difference technique described in the univariate satiation model. Together, we obtain the following recursive functions:

$$\begin{aligned} p_0(t,s) &= p_0(t-dt,s) + dt\left(-f_0(s)\left(\frac{p_0(t-dt,s+ds)-p_0(t-dt,s)}{ds}\right)\right. \\ &\quad \left. - p_0(t-dt,s)f_0'(s) - xg(s)p_0(t-dt,s) + p(t-dt,s,b)\right) \\ p_1(t,s,\tau) &= \text{Enqueue}[\text{DeleteLast}[p_1(t-dt,s,\tau) \\ &\quad + dt\left(-f_1(s)\left(\frac{p_1(t-dt,s,\tau)-p_1(t-dt,s-ds,\tau)}{ds}\right)\right. \\ &\quad \left. - p_1(t-dt,s,\tau)f_1'(s)\right)]\ ,\ dt\cdot xg(s)p_0(t-dt,s)] \end{aligned} \tag{4.2.1}$$

To simulate this population we need to initialize the p-state. We will provide two initializations. In the first example both $P_0$ and $P_1$ are initialized as uniform distributions except for $s = s_m$ for which $p(t, s_m, \tau) = 0$. In the second example we initialize $p_0$ as a linear function $2 - 2s$, and $p_1$ with the more exotic:

$$p_1(0,s,\tau) = \begin{cases} (\frac{x}{0.7})^2 - (\frac{x}{0.7})^4 & : 0 < s < 0.7 \\ 0 & : \text{otherwise} \end{cases}$$

The coding of the simulations can be found in appendix A.2.

The results of the simulations are displayed in figure 4.2.1 and 4.2.2. Subfigures 4.2.1a to 4.2.1d and 4.2.2a to 4.2.2d display the changes of $P_1$ for the two examples. The distribution of p-mass is plotted for $t = 0, 100, 200$ and 800. Furthermore, subfigures 4.2.1e to 4.2.1f and 4.2.2e to 4.2.2f show the distributions of $P_0$ at time $t = 0$ and $t = 800$. Finally, the change that has occurred in the distribution of $P_0$ at is displayed in subfigures 4.2.1g to 4.2.1h and 4.2.2g to 4.2.2h. These subfigures display two viewpoints of the change of $P_0$ for the two examples.

By inspecting these figures we have obtained the following properties of the model. In both examples we can see that the distributions converges over time to the distributions displayed in subfigures d and f. This suggests that the convergent p-state does not depend on the initial distribution.

Interestingly, the convergence isn't as smooth as the univariate satiation based model. Soon after initialization a peak emerges in both $P_0$ and $P_1$ as can be seen in subfigures b, g and h. Even though subfigures b and c do seem to imply a very gradual change, subfigures g and h show that in fact the convergence is rather 'bumpy'. Like the handling time based model, this is caused by the delay brought forth by the handling time taking $b$ long. In these particular examples the total population size of $P_0$ is considerably lower at initialization than the total population size of $P_1$. Recall that we initialized $P_0$ as a uniform distribution, so before $t = 100$, the inflow of p-mass in $P_0$ is constant. However, beyond $t = 100$ the inflow is reduced to the outflow of $P_0$ at $t = 0$ which was much lower just after the initialization. This causes a decrease of the total p-mass of $P_0$ since the outflow is now suddenly much higher than the inflow. The impact of this sudden decrease of inflow continues to impact the distribution throughout our simulation. The impact however diminishes as $t$ increases. The change of the total p-mass is also displayed in figure 4.2.3.

In addition it seems that the total population size is unaffected by the changes generated in this model. A decrease in the total p-mass of $P_0$ results in an increase of the total p-mass of $P_1$ and vice versa. This implies that the total population size of both subpopulation combined is not altered by the changes that occur in the p-state over time.

Lastly there are observations to be made about the convergent distribution itself. Like with the satiation based model, the searching predators $P_0$ converge to a peaked distribution. The predators handling prey however do not converge to a mere peak. As we can see by inspecting subfigures d and figure 4.2.4, at the inflow $\tau = 0$ the distribution is roughly the same as the distribution of searching predators. But as time progresses, the kurtosis of the distribution increases as the p-mass is shifted along $(0, s_m]$. This is caused by the difference in the digestion rate which depends on $s$. Higher levels of satiation have a higher digestion rate. This reduced the net increase of satiation caused by the ingestion of the prey item. In other words: ingestion rate $u$ isn't much larger than the digestion rate $-as$. At lower levels of satiation the reverse is true. Digestion doesn't happen as fast as ingestion so p-mass is transported along $(0, s_m]$ relatively fast. This causes the initial distribution

at $\tau = 0$ to be compressed as $\tau$ increases. Hence the final distribution has a higher kurtosis. This process is also illustrated in figure 4.2.5.

## 4.3 Mathematical derivation of properties

Many of the characteristics observed in the simulations can be derived analytically as well. In this section we will concern us with the observations that the total population size does not change over time and that the convergent distribution does not to depend on the initial distribution.

The total population size can be quantified in the following way:

$$\begin{aligned} P(t) &= P_0(t) + P_1(t) \\ &= \int_0^{s_m} p_0(t,s)ds + \int_0^{s_m}\int_0^{b} p_1(t,s,\tau)d\tau ds \end{aligned} \tag{4.3.1}$$

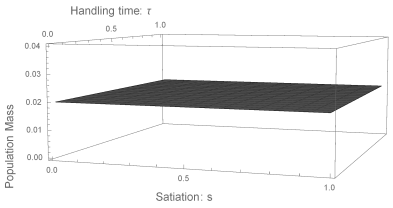
Since we want to show that the total population size doesn't change we will show that the derivative equals 0. We have the following derivative:

$$\frac{dP(t)}{dt} = \int_0^{s_m} \frac{\partial p_0(t,s)}{\partial t}ds + \int_0^{s_m}\int_0^{b} \frac{\partial p_1(t,s,\tau)}{\partial t}d\tau ds \tag{4.3.2}$$

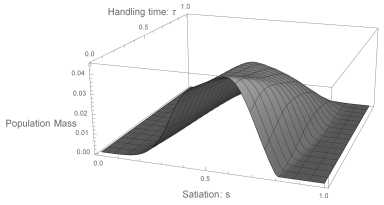
We can substitute (4.1.2) in equation (4.3.2). We'll rewrite the two terms of the resulting equation separately.

$$\int_0^{s_m} \frac{\partial p_0(t,s)}{\partial t}ds = \int_0^{s_m} \left( -\frac{\partial}{\partial s}\left(f_0(s)p_0(t,s)\right) - xg(s)p_0(t,s) + p_1(t,s,b) \right) ds$$

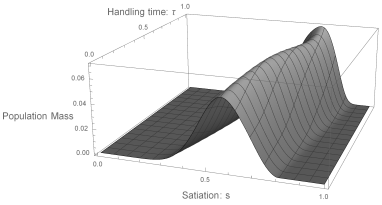
$$\begin{aligned} \int_0^{s_m} \frac{\partial p_0(t,s)}{\partial t}ds &= -\int_0^{s_m} \left( \frac{\partial}{\partial s}\left(f_0(s)p_0(t,s)\right) \right) ds - \int_0^{s_m} xg(s)p_0(t,s)ds + \\ &\quad \int_0^{s_m} p_1(t,s,b)ds \\ &= -f(s)p(t,s)|_{s=0}^{s=s_m} - \int_0^{s_m} xg(s)p_0(t,s)ds + \int_0^{s_m} p_1(t,s,b)ds \end{aligned}$$



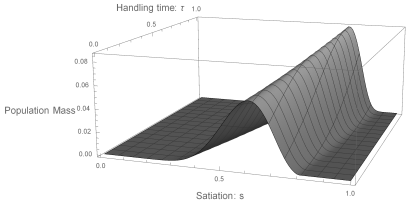

(a) $P_1$ at $t = 0$



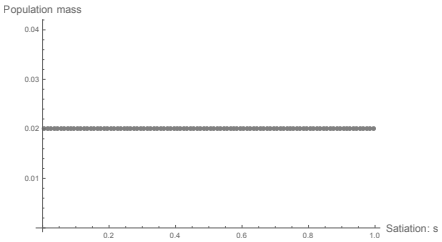

(b) $P_1$ at $t = 100$



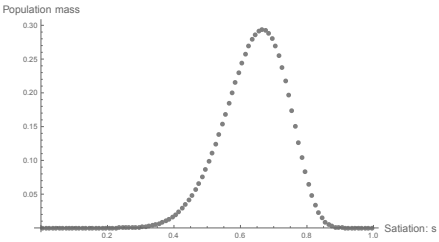

(c) $P_1$ at $t = 200$



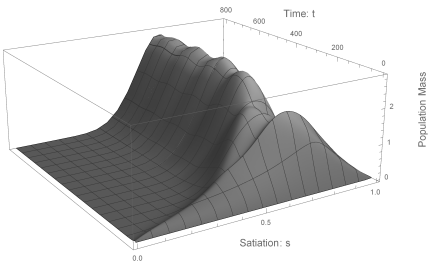

(d) $P_1$ at $t = 800$



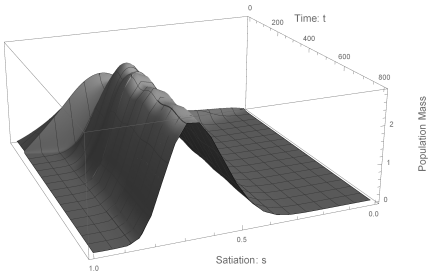

(e) Initialization of $P_0$




(f) Distribution of $P_0$ at $t = 800$

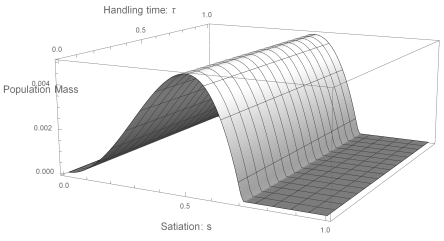



(g) Change of $P_0$ over time, angle 1

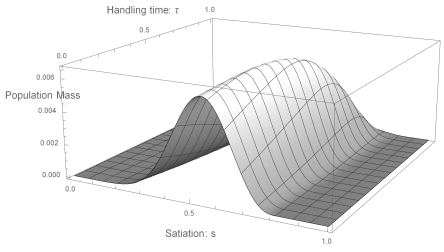



(h) Change of $P_0$ over time, angle 2

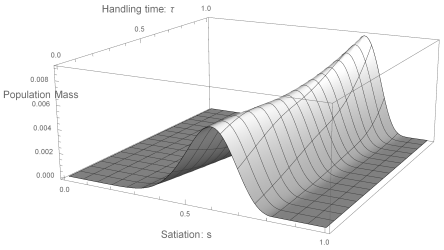
Figure 4.2.1: Change of p-state over time of example 1. The figures are initialized with a uniform distribution. Subfigures 4.2.1a to 4.2.1d display the change of $P_1$. Subfigures 4.2.1e to 4.2.1h display the states and changes of $P_0$.



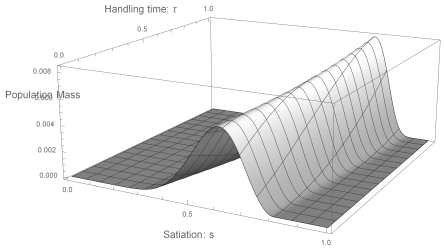

(a) $P_1$ at $t = 0$



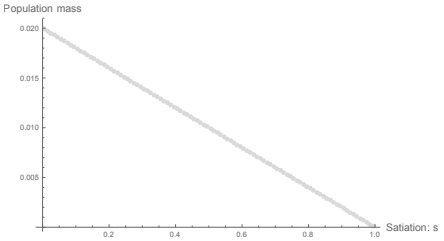

(b) $P_1$ at $t = 100$



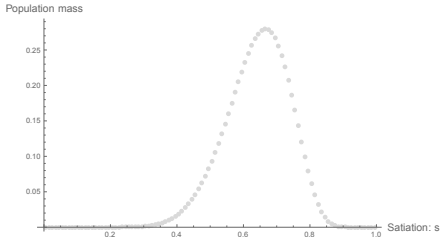

(c) $P_1$ at $t = 200$



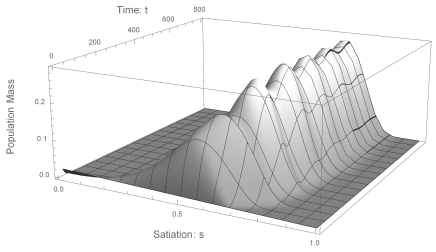

(d) $P_1$ at $t = 800$



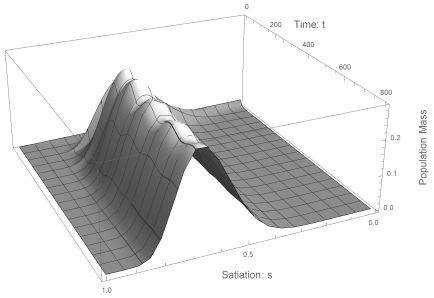

(e) Initialization of $P_0$




(f) Distribution of $P_0$ at $t = 800$




(g) Change of $P_0$ over time, angle 1




(h) Change of $P_0$ over time, angle 2

Figure 4.2.2: Change of p-state over time of example 2. Subpopulation $P_0$ is initialized as $0.01 * (2 - 2s)$, a diagonal line, $P_1$ is initialized with $(\frac{x}{0.7})^2 - (\frac{x}{0.7})^4$. Subfigures 4.2.1a to 4.2.1d display the change of $P_1$. Subfigures 4.2.1e to 4.2.1h display the states and changes of $P_0$.

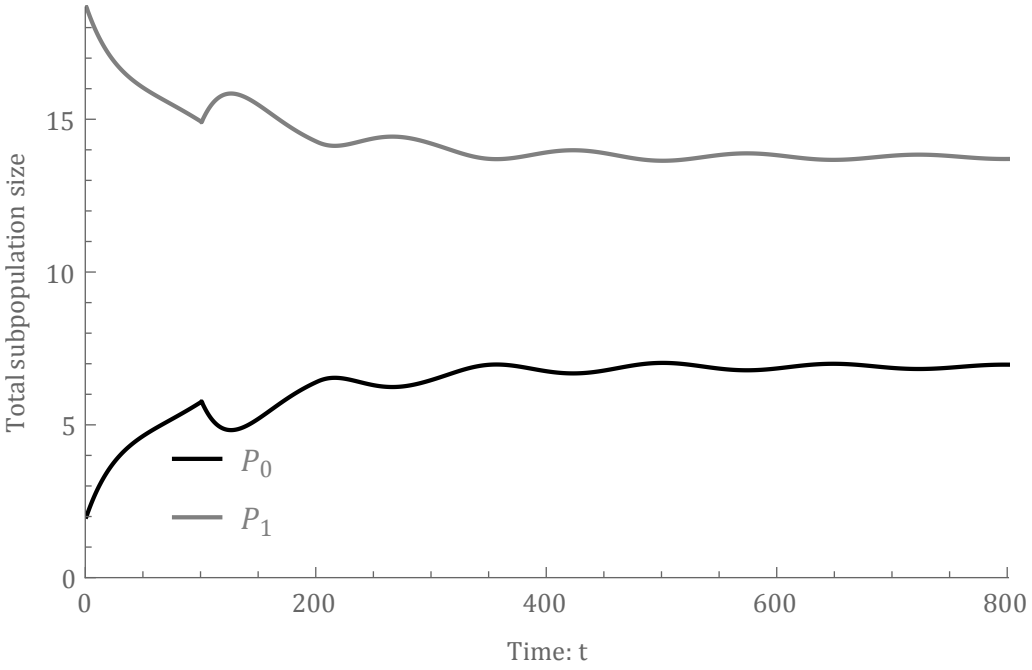



Figure 4.2.3: The change of the total subpopulation size of $P_0$ and $P_1$ of example 2 over time.

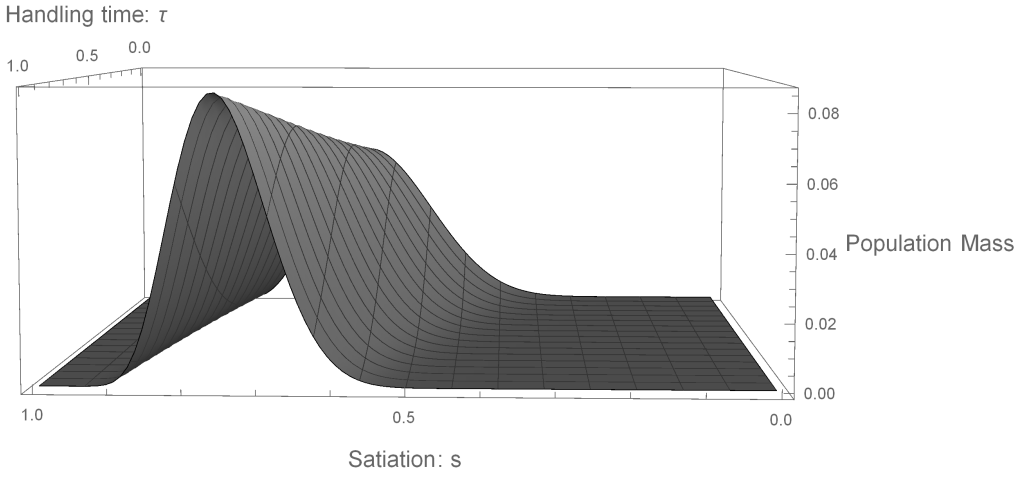



Figure 4.2.4: The limit of the multivariate structured population model form another angle.

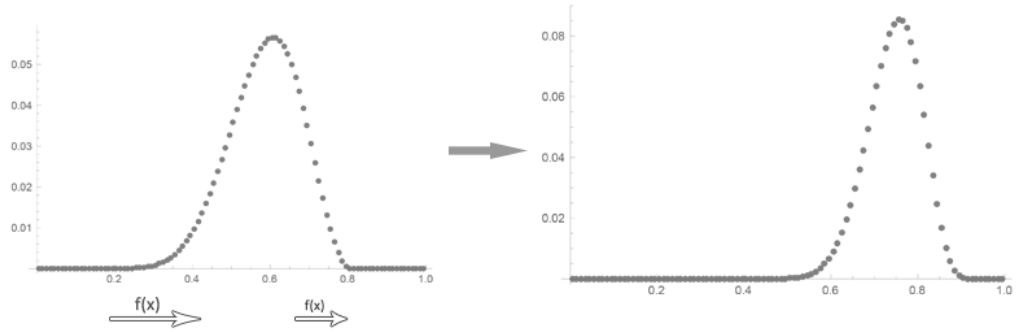



Figure 4.2.5: The emergence of a narrower peak as a result of a difference in the digestion speed. The image on the left shows the convergent distribution of $P_1$ at $\tau = 0$. The image on the right shows the convergent distribution of $P_1$ at $\tau = 1$.

Using the same reasoning as with the satiation based model, we can claim that $\lim_{s\downarrow 0} p_0(t,s) = 0$. If we then also use boundary condition (4.1.3), the first term equals 0 and we obtain our first intermediate result.

$$\frac{dP_0(t)}{dt} = -\int_0^{s_m} xg(s)p_0(t,s)ds + \int_0^{s_m} p_1(t,s,b)ds \tag{4.3.3}$$

Next we rewrite the second term of (4.3.2). We get:

$$\begin{aligned}
\int_0^{s_m}\int_0^{b} \frac{\partial p_1(t,s,\tau)}{\partial t} d\tau ds &= -\int_0^{s_m}\int_0^{b} \left( -\frac{\partial}{\partial s}\left(f_1(s)p_1(t,s,\tau)\right) - \frac{\partial p_1}{\partial \tau}(t,s,\tau)\right) d\tau ds \\
&= -\int_0^{b}\int_0^{s_m} \frac{\partial}{\partial s}\left(f_1(s)p_1(t,s,\tau)\right) ds d\tau - \int_0^{s_m}\int_0^{b} \frac{\partial p_1}{\partial \tau}(t,s,\tau) d\tau ds \\
&= -\int_0^{b} f_1(s)p_1(t,s,\tau)|_{s=0}^{s=s_m} d\tau - \int_0^{s_m} p_1(t,s,b)ds + \int_0^{s_m} p_1(t,s,0)ds \\
&= -\int_0^{b} f_1(s)p_1(t,s,\tau)|_{s=0}^{s=s_m} d\tau - \int_0^{s_m} p_1(t,s,b)ds + \int_0^{s_m} xg(s)p_0(t,s)ds
\end{aligned}$$

Note that for all $s$, $f(s) > 0$. Thus p-mass can never reach $s = 0$ (which is also not within $\Omega$). This indicates that $\lim_{s\downarrow 0} p(t,s,\tau) = 0$. If we then use

boundary condition (4.1.5), again the first term equals 0 and we obtain our second intermediate result.

$$\frac{dP_1(t)}{dt} = -\int_0^{s_m} p_1(t,s,b)ds + \int_0^{s_m} xg(s)p_0(t,s)ds \tag{4.3.4}$$

If we then substitute intermediate results (4.3.3) and (4.3.3) into (4.3.1), we obtain:

$$\begin{aligned}\frac{dP(t)}{dt} &= -\int_0^{s_m} xg(s)p_0(t,s)ds + \int_0^{s_m} p_1(t,s,b)ds \\ &\quad -\int_0^{s_m} p_1(t,s,b)ds + \int_0^{s_m} xg(s)p_0(t,s)ds \\ &= 0\end{aligned} \tag{4.3.5}$$

So in fact the total p-mass does not change over time. This is in line with what we observed in the simulations.

To conclude this section, we will provide a method to obtain the convergent distribution of this system. The procedure has similarities with the univariate satiation based model, but is considerably more complex. The convergent p-state is brought forth in particular by the system of PDEs (4.1.2) and boundary condition (4.1.4). In the equilibrium all derivatives with respect to time vanish. We therefore obtain:

$$\begin{aligned}0 &= -\frac{\partial}{\partial s}\left(f_0(s)p_0(s)\right) - xg(s)p_0(s) + p_1(s,b) \\ 0 &= -\frac{\partial}{\partial s}\left(f_1(s)p_1(s,\tau)\right) - \frac{\partial p_1}{\partial \tau}(s,\tau)\end{aligned} \tag{4.3.6}$$

Note that we cannot simply initialize a point on, say $P_0$, and approximate the other values with a Euler forward type of approach. If we would try this for $P_0$, we would lack the equilibrium values of $p_1(t,s,b)$. This implies that we should start with $P_1$ instead. However, the inflow of $P_1$ depends on the equilibrium of $P_0$ because of boundary condition (4.1.4). This means we should initialize $P_0$ instead. But we already established that $P_0$ needs the equilibrium values of $P_1$, so that's not possible. Apparently the equations circle reference. This problem can however be solved regardless.

We will still use an Euler forward (EF) type of approach combined with a finite difference method, but in a different way. First we need to discretize

$\Omega$. Subpopulation $P_0$ should be partitioned in $ds$ sized parts, subpopulation $P_1$ should be partitioned in $ds \times d\tau$ sized parts. To deal with the circular referencing, instead of initializing with a constant we initialize the partitioned $P_0$ with $n = \frac{s_m}{ds}$ variables. If the variable corresponding to part $i$ of $P_0$ is $v_i$, then $P_0 = (v_1, v_2, \ldots, v_i, \ldots, v_n)$. With this initialized value of $P_0$, we can determine $P_1$. Recall that the inflow of $P_1$ is only generated by $P_0$. We get $p_1(s, 0) = xg(s)p_0(s)$. Now we have our first list of values that describe $P_1$ in the equilibrium. Namely when $\tau = 0$. To obtain the states of all other values of $\tau$, we proceed in a way similar to the simulations: by combining a conveyor belt and finite difference type of approach. However instead of change occurring over time $t$, change occurs over $\tau$. We get:

$$\begin{aligned} p_1(s,\tau) = p_1(s,\tau - d\tau) \\ + dt \left( -f_1(s) \left( \frac{p_1(s,\tau - d\tau) - p_1(s - ds, \tau - d\tau)}{ds} \right) \right. \\ \left. - p_1(s,\tau - d\tau) f_1'(s) \right) \end{aligned} \tag{4.3.7}$$

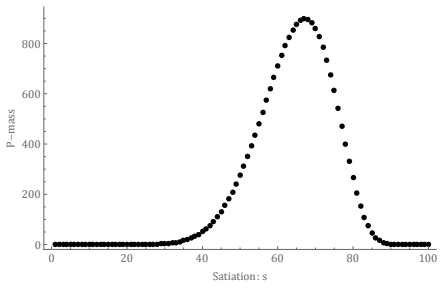
This way we obtain all equilibrium values of $P_1$, in particular $p_1(s, b)$. Now we return to $P_1$. By now $p_1(s, b)$ is no longer unknown as it is now vector of linear combinations of $(v_1, \ldots, v_n)$. Recall that the only reason we were unable to use the EF approach on $P_0$ before, was because $p_1(s, b)$ was unknown. Now since we do know the values this list of points take on, we apply EF the same way as with the satiation based model. For $P_0$ we obtain from the first subequation of (4.3.6):

$$p_0'(s) = \frac{1}{f_0(s)} \cdot (-f_0'(s)p_0(s) - xg(s)p_0(s) + p_1(s, b)) \tag{4.3.8}$$

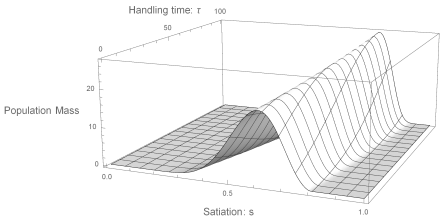
Distribution $P_0$ is then computed by using $p_1(s, b)$ we just obtained and applying EF starting from point 1: $v_1$. So for any point of $P_0$ except the first, we have:

$$p_0(s) = p_0(s - ds) + ds \cdot p_0'(s - ds) \tag{4.3.9}$$

Now finally we should realize that the values of $p_0(s)$ we just determined are redundant in the sense that we already defined what these values are, namely $(v_1, \ldots, v_n)$. The values we just computed with 4.3.9 in fact refer to the same points, but they include the change such points endure over time. Both quantities of these points should therefore be equal, because in




(a) P-state of $P_0$.




(b) P-state of $P_1$.

Figure 4.3.1: The shapes of the convergent p-states of subpopulation $P_0$ and $P_1$.

an equilibrium no change is allowed. We therefore need to solve the following system of equations:

$$\begin{aligned} v_1 &= \theta \\ v_2 &= v_1 + \frac{ds}{f_0(s_1)} \cdot (-f_0'(s_1)v_1 - xg(s_1)v_1 + p_1(s_1, b)) \\ &\vdots \\ v_n &= v_{n-1} + \frac{ds}{f_0(s_{n-1})} \cdot (-f_0'(s_{n-1})v_{n-1} - xg(s-ds)v_{n-1} + p_1(s_{n-1}, b)) \end{aligned} \tag{4.3.10}$$

where $\theta > 0$ can be picked arbitrary, but a $\theta$ close to 0 gives the best approximation of the actual convergent distribution as $v_1$ tends to 0 for $ds \downarrow 0$. Solving this system and substituting all the solutions $v_i$ in $P_0$ and $P_1$ provides us with the approximated shape of the convergent distribution: $\overline{P_0}$ and $\overline{P_1}$. To obtain the actual convergent distributions $\widehat{p_0(s)}$ and $\widehat{p_1(s,\tau)}$ we should correct $\overline{P_0}$ and $\overline{P_1}$ for the actual total population size $P$. We get:

$$\begin{aligned} \widehat{p_0(s)} &= \overline{p_0(s)} \cdot \frac{P}{\int_0^{s_m} \overline{p_0(s)} ds} \\ \widehat{p_1(s,\tau)} &= \overline{p_1(s,\tau)} \cdot \frac{P}{\int_0^{s_m} \int_0^b \overline{p_1(s,\tau)} d\tau ds} \end{aligned} \tag{4.3.11}$$

We see that this procedure does not depend on the initial distribution. Thus the shape of the convergent distribution does not depend on the shape of the initial distribution.

As an example we calculated $\overline{P_0}$ and $\overline{P_1}$ for the examples provided in figures 4.2.1 and 4.2.2. This is coded in Mathematica and the coding can be found in appendix A.1. The result is displayed in figure 4.3.1. By inspecting these results we can clearly see that these distributions are equal to the p-states we observed in subfigures d and f of figures 4.2.1 and 4.2.2.

# 5 Comparison of the uni- and multivariate predator prey population models

In the previous chapters we have formulated, simulated and analyzed three i-state based predator prey population models. The complexity of the models gradually increased as we progressed, concluding with a model that incorporated all features of the univariate models we formulated before. In this chapter we will compare the univariate and multivariate population models by showing similarities and differences. In doing so we can show the added value of using not only these specific models, but of structured population models in general.

## 5.1 Characteristics of the models

Before comparing the characteristics of the example models, we will revisit the different models and give a short overview of some of their most important characteristics.

### 5.1.1 The univariate handling time based model

The handling time model is based on the idea that the time spent on handling a prey item brings forth the change in a population. Using an i-state $\tau$ to bookkeep the handling time that has passed up until that moment, we were able to set up a model in which we distinguished between two types of subpopulations: predators busy handling prey and predators busy searching for prey. The former could be structured with the use of $\tau$ since predators had been handling prey for $0 \leq \tau \leq b$ long. The latter was not internally structured and was simply a memoryless collection of individuals. Change of the p-state is generated by three processes. First of all, as time progresses predators handling prey are transported along $[0, b]$. Second of all, predators that are done handling prey jump to searching prey. Last of all, predators that spot a prey item start searching for prey.

This model has the following properties. For one thing, the population distribution converges to a uniform distribution in a somewhat bumpy fashion. The convergent distribution is not affected by the initial distribution, as all initial structure is lost by the time all the initialized p-mass of the predators handling prey has transitioned to the searching subpopulation of predators. The loss of the initial structure is caused by the lack of memory

of the searching predators. All predators have the same chance of spotting a prey item, regardless of the time they have already been searching. The speed of convergence itself is for a large part determined by the total time $b$ it takes to handle a prey item. This is the time it takes for the initial distribution to jump to the subpopulation of searching predators. Finally, the total population size is unaffected by the change generated over time. The model is a closed system where p-mass is only transported, but not altered in total quantity.

### 5.1.2 The univariate satiation based model

The satiation model is based on the idea that satiation affects the need and desire for food and the speed at which the satiation decreases over time. By using an i-state $s \in (0, s_m]$ to bookkeep the satiation of a predator, we described this population as a distribution of searching predators with satiation $s$. Change occurs as a result of three generating factors. Firstly, internal jumps occur when predators spot prey. We assumed the ingestion to be instantaneous which causes the satiation to increase with fixed quantity $w$. This causes a decrease of p-mass at the source. Secondly, after a jump the predators appeared at $s + w$ which causes an increase of p-mass at that i-state. Thirdly, digestion caused transportation of predators along $\Omega$.

We then continued to derive some properties. It turned out that the population distribution converged smoothly to a peaked distribution over time. This convergence was not affected by the initial distribution as it smoothly transitioned from the initial distribution to the peaked distribution. Also the total population size was unaffected by the change over time, since p-mass was simply moved but not altered.

### 5.1.3 The multivariate satiation and handling time based model

Finally, the multivariate satiation and handling time model is based on the combination of the univariate models. Like in the univariate satiation model we assumed that the predation rate and digestion rate are influenced by the level of satiation. However now we also assumed there was an ingestion rate independent of the satiation $s$. This caused the satiation to increase when a predator was handling prey. Now the i-state is the combination of the level of satiation $s$ and the passed handling time $\tau$. We distinguished between two types of predators, predators that are searching prey and predators that

are handling prey. Only the latter is influenced and thus structured by both i-state variables. The former is affected by satiation only since all predators of this type have $\tau = 0$.

The characteristics of this model are the following. The population distribution converges over time in a bumpy fashion. The convergent distribution was in a way a combination of the two convergent distributions of the univariate models. The subpopulation of predators searching for prey converged to a peaked distribution. The population distribution of the predators handling prey however wasn't simply a peak, but was shaped more like a ridge. At $\tau = 0$ we saw a peaked distribution which had the same shape as $p_0$. Over time this peaked distribution was transported and compressed to form a narrower distribution at higher values of $s$. In this convergence, the initial distribution at $t = 0$ does not seem to affect this convergent distribution. Also the total population size is unaffected by the changes.

## 5.2 Similarities and differences

Using the properties we reiterated in the previous section, we can now analyze the similarities and differences between the different models. This is combined in table 5.2.1.

| | $\tau$ | $s$ | $\tau$ and $s$ |
|---|---|---|---|
| Number of state types | 2 | 1 | 2 |
| Converges | yes | yes | yes |
| Way of converging | bumpy | smooth | bumpy |
| Total pop. size changes | no | no | no |
| Initial distribution affects convergent distribution | no | no | no |
| Shape of final distribution | uniform | peak | peak and ridge |

Table 5.2.1: Comparison of the uni- and multivariate structured population models based on satiation and handling time.

By inspecting the table it becomes clear that the three proposed models are in many ways very similar. In particular in the way they converge. Mass is relocated in the beginning until a stable distribution arises. The

result of this process is not affected by the initial distribution and also this process does not change the total population size. These similarities can in large be ascribed to the similar assumptions that underlie the models: the prey density is kept constant, no births or deaths occur and we only model predators searching for prey and predators handling prey. These assumption don't just simplify our model and make them easier to compare, they also cause similar behavior in particular in the long run. Similarities are also the result of the fact that the models are all structured population models. Since these kinds of models are constructed in the same way, the mechanics that deal with the assumptions stated before are equivalent.

There are however differences between these models. For one thing, the models don't converge in the same way. While the satiation based model converges rather smoothly, the handling time and combined model show a bumpy convergence behavior. More importantly however, we also see that the models converge to different shaped distributions. The generating components interact in different ways and therefore result in different distributions of p-mass along $\Omega$. We thus obtain both uniform, peaked and ridge shaped convergent distributions.

The convergent distribution of the multivariate model is particularly interesting as we can see characteristics of the convergent distributions of both univariate models. If we consider the shapes of the convergent distribution of the multivariate model, this is mostly the result of characteristics of the satiation based model. Subpopulation $P_0$ converged to a peak similar to the peak we see with the satiation based model. Subpopulation $P_1$ converged to a ridge which is simply a peaked distribution that is somewhat altered over time. The process of convergence is however strongly influenced by the characteristics of the handling time based model. The delay which is caused by having a handling time of $b$ causes the convergence to be rather bumpy and it also takes longer than when the satiation based model.

All in all, modeling predator prey population dynamics with different models has resulted in both strong similarities and differences. All models converged to a predetermined distribution regardless of the initialization. The process of convergence and the resulting distributions themselves where however different among the models.

## 5.3 The added value of using a structured population model

We have now arrived at the pinnacle of this thesis since can now underpin the added value of using a structured population model. Through the specific results of the proposed models we have been able to make a general observation. Namely: by using a structured population model we don't only know *that* dynamics converge in some way, but we know *what* kinds of distribution they converge *to*. This is a major advantage of the use of structured population models. Modeling a population is about understanding how and why populations change and what they change into. By modeling the distribution of a population we obtain information about both the individuals and the entire population.

Imagine if we had used a top down approach and we had just modeled the total population size. Modeling only the total population size would have provided us with very little information, since this did not change over time. If we had even only modeled the subpopulation sizes, then still our information of the convergent population would be limited. In this case we would know how the subpopulations relate, but we would have no information about the structure of the subpopulations or how this structure affects the dynamic itself. This is why structured population models allow for a deeper understanding of the individuals and the population as a whole.

# 6 Discussion

This thesis was centered around the theory behind and examples of structured population models. In essence this thesis has provided two kinds of results.

First of all, we have shown how to construct and use structured population models. We've seen how to set up a structured population model in four steps, what characteristics should be modeled and what general rules such models abide by. These practical tools are provided both explicitly and implicitly. We explicitly proposed our four steps of modeling which were also demonstrated with the the uni- and multivariate structured population models. We implicitly provided you with an understanding of how you could go about simulating and generating such a model by using queues or by using a finite difference type approach. All in all these results give you the tools necessary for constructing, simulating and analyzing a structured population model of your own.

Second of all, we have shown why structured population model are relevant and why they should often be preferred over unstructured population models. Structured population models allow us to utilize characteristics of the individual that affect its population related behavior. These characteristics are the basis of modeling change in a population with a bottom-up approach. With such a basis we acknowledge the complexity of the individual and implement it in our model instead of ignoring it. This provides us with a lot more information about the population, since structured population models generate not only macro scale states like population size, but generate the distribution of individuals as well. This is an important advantage of the use of structured population models.

Because of the benefits of structured population models, our opinion might be perceived as being that structured population models are always the best choice. Obviously top-down approaches of modeling do have their own advantages. For one thing structured population models are often much more computationally demanding. Simulating a distribution requires a lot more computations than simulating only the total population size. Another reason for choosing a top-down approach could be that all the detail a structured population model provides is unnecessary. If for instance the total population size is the only quantity you are interested in, then a structured population model might be unnecessarily complex.

In general however, generating the change of the p-state can provide in-

sights in the dynamic of a population that could have gone unnoticed if a top-down approach had been used. Choosing a structured population model will bring forth an in-depth understanding of what a population was, is and will become. And for an ecologist, that's what its all about.

# References


[1] Heijmans, H.J.A.M.: *Holling's 'hungry mantid' model for the invertebrate functional response considered as a Markov process*. Part III: Stable satiation distribution. Journal of Mathematical Biology 21: 115-143 (1984).

[2] Holling, C.S.: *The Functional Response of Invertebrate Predators to Prey Density*. Mem. Ent. Soc. Canada, (1966).

[3] Levin, S.: *Lecture Notes in Biomathematics, The Dynamics of Physiologically Structured Populations*. Springer-Verlag, Berlin, (1986).

[4] McCauley, E.: Wilson, G.W. and de Roos, A.D.: Dynamics of Age-structured and Spatially Structured Predator-Prey Interactions: Individual-based Models and Population-level Formulations. *The Americal Naturalist*, Vol. 142, No. 3. pp. 412-442 (1993).

[5] Odum, E.P.: *Fundamentals of Ecology. Second Edition*. W.B. Saunders Co. Philadelphia and London, (1959).

[6] Voit, E.O.: Martens, H.A. and Omholt, S.W.: 150 Years of the Mass Action Law *PLOS Computational Biology*, Vol. 11, No. 1, (2015).

# A Appendices

## A.1 Quantifying the appearance term of the general p-equation

The quantity of appearing p-mass is for a large part determined by the quantity of disappearing p-mass. We will therefore first quantify the disappearance of p-mass due to internal jumps and we will use this to quantify the appearance of p-mass afterwards. This section is based on theory from chapter three in Levin's book [3].

Let $\gamma(x, y)\Pi dy_i$ be the probability per unit of time that an individual with i-state $x$ will jump to $\Pi(y_i, y_i + dy_i)$. If we integrate over all $y \in \Omega$, we can determine the chance of an individual jumping, since we combine the chances of jumping to all possible destinations. This means that the contribution of $\alpha_j$ to $\alpha$ is:

$$\alpha_j(x) = \int_\Omega \gamma(x, y)\Pi dy_i \tag{A.1.1}$$

We can rewrite this equation. Notice that $\gamma(x, y)$ acts as a density function. We can obtain this density function by using the fact that $\gamma(x, y)$ must be equal to the chance of jumping times the chance of jumping to a specific $y$. Or:

$$\gamma(x, y) = \alpha_j(x)p(x, y) \tag{A.1.2}$$

where $p(x, \cdot)$ is the probability density of "touch down" for jumps starting at $x$. We will use this relation to quantify the appearance of mass in section 2.3.4.

We can use this density function to quantify the reappearance of p-mass within the interior of $\Omega$. Such increases of p-mass are a result of jumps like births, internal jumps and jumps from the boundary. In general we will consider all such arrivals to be births since this simplifies our model. The probability function we just described therefore accounts for both jumps and actual births. Sometimes, we have more than one touchdown as the result of a birth with multiple children. The average number of touchdowns at $y$ is therefore quantified with $r(y)$.

We now have enough information to formulate the first contribution to $\delta_{s+}n(x)$ using a mass action law argument. If we separate partitions of $\Omega$

and denote these as $\Omega_i$, then we have:

$$\text{Contribution 1:} \sum_i \{ \int_{\Omega_i} r(y)p(y,x)\alpha_J(y)n(y)dY \} \tag{A.1.3}$$

The integral describes the expected number of arrivals at $x$ resulting from jumps originating from $\Omega_i$. This is the product of the expected number of offspring, chance of jumping to $x$, the chance of jumping at all and the number of individuals. If we summarize the expected number of departures to $x$ over all area's in $\Omega$, we obtain the contribution to the appearance of p-mass because of (strictly) internal births.

So far we have ignored jumps from the boundary of $\Omega$ which are allowed in some structured population models. These follow a similar reasoning as before, with the difference being that we use the adjusted flux at the boundary which quantifies the amount of p-mass flowing through the boundary. Using $r(y)$ and $p(y,x)$, we arrive at the second contribution:

$$\text{Contribution 2:} \sum_i \{ \int_{\partial_- \Omega_i} r(y)p(y,x)\phi(y)\nu(y)dY \} \tag{A.1.4}$$

Now we can add up these contributions to arrive at the contribution of p-mass from within the interior of $\Omega$. We obtain:

$$b(x) = \sum_i \{ \int_{\Omega} r(y)p(y,x)\alpha_J(y)n(y)dY + \int_{\partial_- \Omega_i} r(y)p(y,x)\phi(y) \cdot \nu(y)d\sigma \} \tag{A.1.5}$$

## A.2 The programming behind the various structured population models

In this appendix all of the coding behind the simulations is provided. Note that often some boundary conditions are not incorporated in the model. This is justified because many boundary conditions don't influence the simulation in a noticeable way. By omitting them the simulations often become much simpler.

### A.2.1 Univariate handling time based model coding

The simulation of section 3.2 is programmed in Mathematica as follows:

```
With[{d\[Tau] = 0.01, dt = 0.01, a = 0.09, x = 10.},
 ClearAll[f];
 f[0] = {50.,
   d\[Tau]*Table[
     25. - 5. (i - 0.5*d\[Tau]), {i, d\[Tau], 1., d\[Tau]}]};
 f[t_] :=
  f[t] = {f[t - 1][[1]] + Last[f[t - 1][[2]]] - a x dt f[t - 1][[1]],
    Delete[Prepend[f[t - 1][[2]], a x dt f[t - 1][[1]]], -1]};
 ]
```

### A.2.2 Univariate satiation based model coding

The simulation of section 3.3 is programmed in Mathematica as follows:

```
With[{dt = 0.01, a = 0.5, x = 10., ds = 0.01, w = 0.1},
 ClearAll[f, g, p];
 f[s_] := -a s;
 g[s_] :=
  Piecewise[{{0, s < 0}, {1. - 1.25 s, 0 <= s < 0.8}, {0, s >= 0.8}}];
 p[0] = Table[{i, ds 30/4 (i^2 - i^4)}, {i, 0.5*ds, 1. - 0.5*ds, ds}];
 p[t_] := p[t] =
   Block[{s = #1[[1]], \[Rho] = #1[[2]], pminw},
      pminw = SelectFirst[p[t - 1], #[[1]] == s - w &, {0, 0}][[2]];
      {s,
       Max[\[Rho] +
         dt (-f[s]*((#2[[2]] - \[Rho])/ds) - \[Rho]*f'[s] -
```

```
            x g[s] \[Rho] + x g[s - w] pminw), 0]}] & @@@
    Partition[p[t - 1]~Join~{{1 + 0.5 ds, 0}}, 2, 1]
 ];
```

The approximation of $\overline{p(s)}$ which is used for determining the convergent distribution is programmed in Mathematica as follows:

```
With[{a = 0.5, x = 10., ds = 0.001, w = 0.1, windex = 100},
 ClearAll[p, f, g];
 f[s_] := -a s;
 g[s_] :=
  Piecewise[{{0, s < 0}, {1. - 1.25 s, 0 <= s < 0.8}, {0, s >= 0.8}}];
 p[100] = 10^-9;
 Set[p[#], 0] & /@ Table[-i, {i, 0, 110}];

 p[sindex_] := p[sindex] =
   Block[{s = sindex*ds - 0.5*ds},
    If[sindex > windex,
     p[sindex - 1] +
      ds*(1/f[s] (-f'[s] p[sindex - 1] - x g[s] p[sindex - 1] +
           x g[s - w] p[(sindex - windex - 1)])),
     p[sindex + 1] -
      ds*(1/f[s] (-f'[s] p[sindex + 1] - x g[s] p[sindex + 1] +
           x g[s - w] p[(sindex - windex + 1)]))
     ]
    ]
 ]
```

### A.2.3 Handling time and satiation based model coding

The simulation of chapter 4 is programmed in Mathematica as follows:

```
With[{dt = 0.01, a = 0.5, x = 10., ds = 0.01, w = 0.1, d\[Tau] = 0.01,
    taue = 0.5, u = 0.5},
  ClearAll[g, p0, p1, f0, f1];

  (*Functions*)
  f0[s_] := -a s;
  f1[s_] := -a s + u;
```

```
g[s_] :=
 Piecewise[{{0, s < 0}, {1. - 1.25 s, 0 <= s < 0.8}, {0, s >= 0.8}}];

(*Initialization*)
p0[0] = Table[{i, ds 2}, {i, 0.5*ds, 1. - 0.5*ds, ds}];
p1[0] =
 Table[{s, 2 ds}, {s, 0.5*ds, 1. - 0.5*ds, ds}, {\[Tau],
   0.5*d\[Tau], 1. - 0.5*d\[Tau], d\[Tau]}];

(*Recursive functions*)
p0[t_] := p0[t] =
  Block[{s = #1[[1]], \[Rho] = #1[[2]]},
     {s,
      Max[\[Rho] +
        dt (-f0[s]*((#2[[2]] - \[Rho])/ds) - \[Rho]*f0'[s] -
           x g[s] \[Rho]) +
        p1[t - 1][[Round[(s + ds/2)/ds], -1, 2]]
       , 0]}] & @@@
   Partition[p0[t - 1]~Join~{{1 + 0.5 ds, 0}}, 2, 1];
p1[t_] := p1[t] =
  Map[Block[{s = Part[#, 1, 1]}, Delete[Prepend[#,
       {s, dt x g[s] p0[t - 1][[Round[(s + ds/2)/ds], 2]]}], -1]] &,
   MapIndexed[
    Block[{sindex = #2[[1]], \[Tau]index = #2[[2]],
       s = #1[[1]], \[Tau] = #2[[2]]*d\[Tau] - d\[Tau]/2, ps,
       p = #1[[2]]},
      ps =
       If[sindex > 1, p1[t - 1][[sindex - 1, \[Tau]index, 2]], 0];
      {#1[[1]], #1[[2]] + dt (-f1[s]*((p - ps)/ds) - p*f1'[s])}] &,
    p1[t - 1], {2}]
   ]
];
```

The approximation of $\overline{p_0(s)}$ and $\overline{p_1(s,\tau)}$ which are used for determining the convergent distribution is programmed in Mathematica as follows:

```
With[{a = 0.5, x = 10., ds = 0.01, w = 0.1, u = 0.5, d\[Tau] = 0.01},
 ClearAll[g, p0, p1, f0, f1, dp0];
 g[s_] :=
```

```
 Piecewise[{{0, s < 0}, {1. - 1.25 s, 0 <= s < 0.8}, {0, s >= 0.8}}];
f0[s_] := -a s;
f1[s_] := -a s + u;
p1[0] = Table[
  d\[Tau]*x*g[i*ds - 0.5*ds]*ToExpression["v" <> ToString[i]], {i, 1,
    1/ds}];
p1[\[Tau]_] :=
 p1[\[Tau]] =
  Simplify /@
   MapIndexed[#1[[2]] +
      d\[Tau] (-f1[First[#2]*ds - 0.5*ds]*((#1[[2]] - #1[[1]])/ds) -
         #1[[2]]*f1'[First[#2]*ds - 0.5*ds]) &,
    Partition[{0}~Join~p1[\[Tau] - 1], 2, 1]];
dp0[s_, v_] := 1/f0[s] (-f0'[s] v - x g[s] v);
p0 = Table[
  ToExpression["v" <> ToString[i]] +
   ds*dp0[i*ds - 0.5*ds, ToExpression["v" <> ToString[i]]] +
   1/f0[i*ds - 0.5*ds]*p1[99][[i]], {i, 1, 1/ds}];
solutions =
 Solve[ReplacePart[
    MapIndexed[ToExpression["v" <> ToString[First[#2] + 1]] == #1 &,
     p0], -1 -> v1 == 10^-12],
   Table[ToExpression["v" <> ToString[i]], {i, 1, 1/ds}]][[1, All,
   2]];
]
```