How turtles find their way home

H. Remmen

On the migration of loggerhead hatchlings:
Using continuous and discrete methods to model their behaviour and assess recent survival rates.

Cover image: [28]

TU Delft

How turtles find their way home

by

H. Remmen

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
defended publicly on Tuesday June 17, 2025 at 15:00.

Student number:	5751152	
Project duration:	February 25, 2025 – June 17, 2025	
Thesis committee:	Dr. H. Yoldaş	TU Delft
	Dr. habil. J. Spandaw	TU Delft
Supervisors:	Dr. H. Yoldaş	TU Delft
	Dr. V. Freingruber	TU Delft

TU Delft

Acknowledgements

It is now at the end of writing this Bachelor's thesis that I am finally able to thank those people who have made this work possible.

Most importantly, my gratitude goes out to my supervisors: dr. Havva Yoldaş and dr. Viktoria Freingruber. Without your continued guidance, excellent reviews and endless patience, this paper could never have ended up as well as it has. Thank you for making this possible.

I would like to say a special thank you to Félice Dierick for his comments on my work. I could not be more truthful when I say that you are the reason I have studied Mathematics, and I feel honoured to have you by my side.

Finally, I want to thank my family and partner for accommodating my full commitment to this project. The freedom you have granted me is just as valuable as the support you otherwise provide:
Het is gelukt, en daar zijn jullie onmisbaar in geweest. Bedankt Edith, Raymond en Frey.

Hazel Remmen
Delft, June 9th 2025

Abstract

We investigate the migratory behaviour of juvenile loggerhead sea turtle *(Caretta caretta)* hatchlings within a region of the North Atlantic Gyre. To do so, we develop and compare an individual based (IB) model and a partial differential equation (PDE) model, specifically an advection-diffusion equation derived from a position-jump process. We compare their behaviour and show that they yield similar results, but that there are still some differences between the two. Using the IB model we assess survival probabilities of hatchlings in recent years (2016-2023), revealing that survival rates are statistically significantly (p-value $= 2.62 \cdot 10^{-6}$) not constant over time in this region. However, there could be global events that influence the survival probability, as years 2019 and 2023 have a large deviation in survival probability.

Contents

1 Introduction

Sea turtles are part of the group of animals who undertake vast migrations across the oceans. Among them, the young loggerhead sea turtle undertakes a challenging migration across the North Atlantic Gyre: a large system of ocean currents that circulates between Europe, Africa and the Americas. In the north-eastern part of the gyre near Europe, this stream splits into a northbound stream headed towards the cooler waters of Ireland and a southbound stream, towards the warmer waters of Azores, the latter providing favourable survival conditions for this species [5]. Although work has been done on the topic before, much still remains unclear about the navigation of these hatchlings inside the Gyre.

Our goals in this thesis are to

1. implement two models that both describe the migratory behaviour of the loggerhead hatchlings.
2. use the simulations to determine whether the probability of successfully traversing a region of the Gyre has changed in recent years.

A large part of this work is dedicated to the derivation and implementation of these models, namely an *individual-based (IB)* model and a *partial differential equation (PDE)* model. The IB model treats each turtle as an individual, tracing its path step-by-step, whereas the PDE model captures the movement of turtle population densities, describing how their collective presence spreads through the ocean over time.

These models are not limited to the study of migrating sea turtles, or even to that of aquatic migration. Variants of the PDE model we use in this thesis can be used to predict the spread of pollutants through groundwater, the flow of heat in materials or spread of bacteria [10, 32, 33]. Variants of the IB model can be used to understand phenomena such as particle movement in fluids and a diverse range of animal movement models [1, 4, 17].

Previous work

The phenomenon we model is inspired by the work [24], but our approach is different. In [24], the authors use a velocity-jump process (VJP) to derive a PDE known as the FAAD[1] Equation. In a VJP, the movement of an individual is described by discrete changes in velocity: the individual moves at a constant speed in a given direction for a random period of time, and then jumps to a new *velocity* and a new direction.

In contrast, we use a position-jump process (PJP), where movement is represented as a series of discrete, instantaneous *position* jumps on a grid. In our specific PJP, every time an individual makes a step, it moves to an adjacent location on a grid according to probabilities describing the likelihood of moving in a certain direction.

In [24], the VJP and resulting FAAD equation are numerically implemented as models, and applied to the sea turtle migration phenomenon using ocean stream data [16] from 2016 and 2017. Specifically, the models start on 01-01-2016 and runs for 500 days. This ocean stream data is still publicly available,

[1] Fully anisotropic advection-diffusion

so we compare our results in the year 2016 with [24] to validate our models: if they are visually similar, we are satisfied with the model.

Thomas Hillen and Kevin J. Painter, the authors of [24], have (co-)authored numerous works in the context of animal migration models such as [14, 15, 19]. Turtle migration is a recurring topic in their research. For instance, in [25] they develop and apply models to the homing behaviour of green turtles migrating to Ascension Island, a common nesting ground for green turtles. In [24] they present the derivation of PDE models from position-jump and velocity-jump processes and present applications to loggerhead hatchling navigation (the topic also explored in this thesis). Finally, in [26], Painter revisits the travel of green sea turtles to Ascension Island, this time incorporating additional navigational cues into his model.

Considering [24] was published in 2018, and ocean stream data is now available[2] until 2024, after comparing our models, we aim to determine whether in recent years the probability of successfully traversing this region has remained constant.

Our contribution to the topic lies in the derivation and custom implementation of different models (the IB model and the PDE model) using Python, and an overview of the steps to reach that point as well as how to use the models. We also utilize more recent ocean stream data to predict whether, using the proposed survival metric, we can reasonably expect the survival rate to have changed in this region.

Chapter Overview

In Chapter 2 we introduce a position jump model and show the derivation of a position jump model to an advection-diffusion model.

In Chapter 3 we globally present the models we will be implementing, and explain some factors that relate to this such as the simulation region, boundary conditions and initial conditions.

In Chapter 4 we discuss the implementation-related details of both models and provide directions for those trying to reproduce results using the resources we have created.

In Chapter 5 we provide an overview of results our models have created. We compare our work to [24], and determine whether the survivability of the region has changed recently.

In Chapter 6 we discuss the implications of our work, and bring some matters to light that require discussion. We also point out where our work could be improved or continued upon, and conclude the thesis.

[2] We can simulate up to and including 2023, as each simulation takes 500 days. The flow data between 01-01-2024 and 15-05-2025 (500 days) is not currently available

2

Derivation: From random walks to the advection-diffusion equation

In this chapter we show how from a position jump process, under certain limits, a PDE can be derived. We also demonstrate that this derivation works with probabilities depending on space and time. For some solutions to simple cases of the PDE we find, we refer to Appendix D

2.1. The two-dimensional case

For a similar derivation as the one we present here, but in one dimension, we would like to refer to [4]. They provide resources not only on derivations of position (and velocity-) jump processes, but also on more specific types of random walks than in this thesis.

In the one-dimensional random walk (RW), generally a probability of moving left l and moving right r are defined, for an individual walking on $\delta\mathbb{Z} = \{\dots, -2\delta, -\delta, 0, \delta, 2\delta, \dots\}$. Extending this concept to two dimensions, we introduce the probabilities l, r, u and d to denote the probability of an agent moving left, right, up or down on $\delta\mathbb{Z}^2$. We also write $H = l + r$ and $V = u + d$, for the probability of a horizontal or vertical step respectively, and $l + r + u + d = 1$: Standing still has probability 0.

These probabilities are currently not allowed to depend on space or time. Therefore, any agents behaviour will probabilistically be the same in each location, at each moment in time. We start with the derivation assuming this non-dependence, and in Section 2.2 we extend to probabilities that depend on space and time.

Permit $p(x, y, t)$ to denote the probability of an agent on this grid being at position (x, y) at time t. We can derive in 2 dimensions, in a similar fashion to the steps in [24], the probability of moving towards position $p(x, y, t + \tau)$ after having come from adjacent locations

$$\begin{aligned} p(x, y, t + \tau) = {} & lp(x + \delta, y, t) + rp(x - \delta, y, t) \\ & + up(x, y - \delta, t) + dp(x, y + \delta, t). \end{aligned} \tag{2.1}$$

Taking the Taylor expansion about t on the left-hand-side of Equation 2.1 and about x on the right-hand-side, denoting by $\mathcal{O}(x^n)$ those functions which vanish at least as fast as x^n near zero, writing p

without its arguments for brevity, and contracting notation $\frac{\partial^n}{\partial x^n} := \partial_x^n$, or simply $\frac{\partial}{\partial x}p = p_x$, we get

$$\begin{aligned} p + \tau p_t + \mathcal{O}(\tau^2) = l\left(p + \delta p_x + \frac{\delta^2}{2}p_{xx} + \mathcal{O}(\delta^3)\right) \\ + r\left(p - \delta p_x + \frac{\delta^2}{2}p_{xx} - \mathcal{O}(\delta^3)\right) \\ + u\left(p - \delta p_y + \frac{\delta^2}{2}p_{yy} - \mathcal{O}(\delta^3)\right) \\ + d\left(p + \delta p_y + \frac{\delta^2}{2}p_{yy} + \mathcal{O}(\delta^3)\right). \end{aligned}$$

Using $l + r + u + d = 1$ to cancel all the terms containing p, collecting like terms in partial derivatives of p, and moving all the $\mathcal{O}$-terms to the right and contracting the $\mathcal{O}$-notation, we get

$$p_t = \frac{\delta}{\tau}\big((l-r)p_x + (d-u)p_y\big) + \frac{\delta^2}{2\tau}\big(Hp_{xx} + Vp_{yy}\big) + \mathcal{O}\left(\tau, \frac{\delta^3}{\tau}\right). \tag{2.2}$$

Equation 2.2 is our first important result. When taking appropriate limits for the spatial- and temporal discretization, this equation yields an advection-diffusion equation. Letting l, r, u and d be arbitrary and valid probabilities, we can define some new terms that will help us in the next steps.

1. The horizontal (rightward) advection bias $\lim_{\delta,\tau\to 0}\frac{\delta}{\tau}(r-l) = a_H$
2. The vertical (upward) advection bias $\lim_{\delta,\tau\to 0}\frac{\delta}{\tau}(u-d) = a_V$
3. The diffusion intensity $\lim_{\delta,\tau\to 0}\frac{\delta^2}{2\tau} = c$

Note that these limits can all coexist at the same time, so long as $(r-l) \to 0$ and $(u-d) \to 0$ at the same speed as δ, which we assume.

Now we reintroduce Equation 2.2,

$$p_t = \frac{\delta}{\tau}\big((l-r)p_x + (d-u)p_y\big) + \frac{\delta^2}{2\tau}\big(Hp_{xx} + Vp_{yy}\big) + \mathcal{O}\left(\tau, \frac{\delta^3}{\tau}\right),$$

and take the limit of δ and τ to zero such that the three limits we just defined exist. This means that $\lim_{\delta,\tau\to 0}\frac{\delta^3}{\tau} = 0$, as $\lim_{\delta,\tau\to 0}\frac{\delta^2}{2\tau} = c$ is a constant. Therefore the terms inside $\mathcal{O}\left(\tau, \frac{\delta^3}{\tau}\right)$ all go to zero:

$$\begin{aligned} p_t &= -a_H p_x - a_V p_y + cHp_{xx} + cVp_{yy} \\ &= -\partial_x(a_H p) - \partial_y(a_V p) + \partial_{xx}(cHp) + \partial_{yy}(cVp). \end{aligned} \tag{2.3}$$

To move towards a nicer form of Equation 2.3, we use the Nabla operator, $\nabla = \begin{pmatrix}\partial_x \\ \partial_y\end{pmatrix}$. We can introduce it by

$$\begin{aligned} p_t &= \begin{pmatrix}\partial_x \\ \partial_y\end{pmatrix} \cdot \begin{pmatrix}-a_H p \\ -a_V p\end{pmatrix} + \begin{pmatrix}\partial_x \\ \partial_y\end{pmatrix} \cdot \begin{pmatrix}\partial_x cHp \\ \partial_y cVp\end{pmatrix} \\ &= \nabla \cdot \begin{pmatrix}-a_H \\ -a_V\end{pmatrix} p + \nabla \cdot \begin{pmatrix}cH & 0 \\ 0 & cV\end{pmatrix}\begin{pmatrix}\partial_x p \\ \partial_y p\end{pmatrix} \\ &= \nabla \cdot \begin{pmatrix}-a_H \\ -a_V\end{pmatrix} p + \nabla \cdot \begin{pmatrix}cH & 0 \\ 0 & cV\end{pmatrix}\nabla p. \end{aligned} \tag{2.4}$$

Writing $A = \begin{pmatrix}a_H \\ a_V\end{pmatrix}$ and $D = \begin{pmatrix}cH & 0 \\ 0 & cV\end{pmatrix}$ then we can further modify 2.4 to acquire our PDE:

$$\begin{aligned} p_t &= \nabla \cdot (-Ap) + \nabla \cdot D\nabla p \\ &= \nabla \cdot (-Ap + D\nabla p). \end{aligned} \tag{2.5}$$

What if τ and δ scale differently?

One fair question to ask is when τ and δ don't scale precisely as desired. These cases yield slightly different equations to Equation 2.5.

One such case is when $\lim_{\delta,\tau\to 0} \frac{\delta}{\tau}$ is a constant. This means that, as $\delta \to 0$, the diffusion intensity $\frac{\delta^2}{2\tau}$ goes to zero. This means we are left with a model which is purely advective, so using a similar derivation as to acquire Equation 2.5, we acquire

$$p_t = \nabla \cdot (-Ap). \tag{2.6}$$

Alternatively, consider the case when $l = r$ and $u = d$, and $\lim_{\delta,\tau\to 0} \frac{\delta^2}{2\tau}$ exists. Now the advective terms a_H and a_V become zero, while the diffusive term remains existent:

$$p_t = \nabla \cdot (D\nabla p) \tag{2.7}$$

Now assuming that $\delta, \tau \to 0$, we see we have exhausted our options. We have considered the cases where $\frac{\delta^2}{2\tau}$, $\frac{\delta}{\tau}$ (or combinations) converge to a constant while others go to zero. The last case, where both of these terms go to zero, is not of great interest. There could be higher order limits worth considering, when the Taylor expansions about x and t include even higher order terms, but in this thesis we do not investigate these possibilities.

2.2. Spatial dependence of probability

Later on in this paper, we will see that the probabilities of moving, l, r, u, d are also going to be functions of space and time. This is because our agent is acting in a vector field represented by ocean stream data. This data depends on space and time, so the probabilities will become functions of space and time, too. To justify the consistency of the model we have presented when taking this spatial and temporal dependence, we would like to also provide a derivation on the one-dimensional case, but where $l(x,t)$ and $r(x,t)$ are the x- and t-dependent probabilities of moving left and right.

Again we start from

$$p(x, t+\tau) = l(x+\delta, t)p(x+\delta, t) + r(x-\delta, t)p(x-\delta, t)$$

by taking a Taylor expansion, now also for $l(x+\delta, t)$ and $r(x-\delta, t)$ about x, and using the $\mathcal{O}$-notation to collect high-order terms,

$$\begin{aligned} p + \tau p_t + \mathcal{O}(\tau^2) &= \left(l + \delta l_x + \frac{\delta^2}{2} l_{xx} + \mathcal{O}(\delta^3)\right)\left(p + \delta p_x + \frac{\delta^2}{2} p_{xx} + \mathcal{O}(\delta^3)\right) \\ &\quad + \left(r - \delta r_x + \frac{\delta^2}{2} r_{xx} + \mathcal{O}(\delta^3)\right)\left(p - \delta p_x + \frac{\delta^2}{2} p_{xx} + \mathcal{O}(\delta^3)\right) \\ &= lp + \delta l p_x + \frac{\delta^2}{2} l p_{xx} + \delta l_x p + \frac{\delta^2}{2} l_{xx} p + \mathcal{O}(\delta^3) \\ &\quad + rp - \delta r p_x + \frac{\delta^2}{2} r p_{xx} - \delta r_x p + \frac{\delta^2}{2} r_{xx} p + \mathcal{O}(\delta^3). \end{aligned}$$

Note that $l(x,t) + r(x,t) = 1$ always when evaluated at the same point. This also means that $l_x + r_x = 0$, and $l_{xx} + r_{xx} = 0$. Using these, we can write

$$\begin{aligned} p + \tau p_t + \mathcal{O}(\tau^2) &= (l+r)p + \delta\left((l-r)p_x + (l_x - r_x)p\right) \\ &\quad + \frac{\delta^2}{2}\left((l+r)p_{xx} + (l_{xx} + r_{xx})p\right) + \mathcal{O}(\delta^3) \\ p_t &= \partial_x\left(\frac{\delta}{\tau}(l-r)p\right) + \partial_x^2\left(\frac{\delta^2}{2\tau}p\right) + \mathcal{O}\left(\frac{\delta^3}{\tau}, \tau\right). \end{aligned} \tag{2.8}$$

Here we can finally see that the spatial dependency of the advective and diffusive terms does not yield a different result to Equation 2.2. So long as the limits in δ and τ work as we required earlier, we find that the spatial or temporal dependence of $l(x,t)$ and $r(x,t)$ do not matter.

To informally point at the similarity between this result and the above Equation 2.5, we can suggestively replace the remaining derivatives in Equation 2.8 by the 'one-dimensional' ∇-operator, $\nabla = \frac{d}{dx}$ and appropriate in-product operators. This very liberal symbol-pushing approach yields (when taking the limits of δ and τ, and again using $A(x,t)$ and D defined similarly in one dimension)

$$\begin{aligned} p_t &= \nabla \cdot (-A(x,t)p) + \nabla \cdot \nabla Dp \\ &= \nabla \cdot (-A(x,t)p + D\nabla p) \end{aligned} \tag{2.9}$$

for a spatially and temporally dependent $A(x,t)$ and a constant D. In this thesis we heuristically extend this concept to our application in two dimensions.

3

Model Descriptions

As mentioned earlier, the aim of this thesis is to create models which can describe loggerhead migration in a certain region of the North Atlantic Gyre, and to use these models to investigate whether the probability for loggerhead hatchlings to remain in this Gyre has changed in the period 2016-2024.

Because it is quite a big step to immediately discuss implementation details after the mathematical derivation we presented in Chapter 2, we provide a relatively high-level introduction of both the models here.

We employ two modelling approaches: an IB model using Equation 2.1 and a PDE model using Equation 2.9 derived by taking appropriate limits, both with time- and space-dependence. While both models aim to describe the same behaviour, their implementation and interpretation differ. We first introduce some physical concepts common between the models, after which we discuss approaches specific to each model.

Shared concepts

The movement of loggerhead hatchlings we model is the result of three components:

1. **Active undirected movement (diffusion)**: Turtles display locally random, wandering behaviour. This is modelled as a diffusive process, and captures the variations in individual swimming paths.
2. **Active directed movement (advection)**: Turtles also exhibit directional or biased swimming. We assume this bias to be constant throughout all simulations and model it as an advective process.
3. **Passive movement due to ocean currents:** In addition to their own swimming, turtles are carried around by the ocean's flow. This is not a result of the turtles' own effort, and thus allows them to travel large distances. This movement is also modelled as an advective process.

3.1. Individual based model

To implement Equation 2.1,

$$\begin{aligned} p(x,y,t+\tau) = lp(x+\delta,y,t) + rp(x-\delta,y,t) \\ + up(x,y-\delta,t) + dp(x,y+\delta,t), \end{aligned}$$

we use an IB model, in which each turtles position is updated in discrete time steps based on active and passive movements.

The active movement factors are incorporated into the model through a set of constant probabilities $\{\widehat{p_l}, \widehat{p_r}, \widehat{p_u}, \widehat{p_d}\}$ that determine the direction a turtle moves in each time step. How these $\widehat{p_i}$ are found is shown in Section 4.3. The passive movement is incorporated by linearly modifying these probabilities with components of the flow $\{f_l, f_r, f_u, f_d\}$ (which are assumed to be constant during a step, but the flow *field* changes in time and space) and a constant unit-conversion factor γ by

$$p_i = \frac{\widehat{p_i} + \gamma f_i}{1 + \gamma \sum_{j=l,r,u,d} f_j}.$$

Note that these $p_i, i \in \{l, r, u, d\}$ are the probabilities $l(x,y,t), r(x,y,t), u(x,y,t), d(x,y,t)$ in Equation 2.1 (so for example the probability of moving left $l(x,y,t)$ becomes p_l). The $\widehat{p_i}$'s represent the *constant* movement probabilities. We rename them for brevity in writing, as the function $p(x,y,t)$ is not needed to represent populations any more. The flow dataset is updated every 24 hours, but the simulation advances with smaller time steps within each day. It will be discussed in Section 4.3 how γ and the $\widehat{p_i}$ are found.

3.2. PDE-based model

In Chapter 2, we have shown that taking appropriate limits, the RW (Equation 2.1) leads to a PDE model. The model we acquire is a partial differential equation in $u(x,y,t)$, with u representing a population density of turtles[1].

$$u_t = \nabla \cdot (-A(x,y,t)u + D\nabla u)$$

with parameters $A(x,y,t) = a + \alpha f(x,y,t)$ the vector containing the advection terms and D the diffusion matrix being related to the probabilities in the IB model. The relations for the constant terms are found in Chapter 2 as:

$$a = \underbrace{\begin{pmatrix} a_H \\ a_V \end{pmatrix}}_{\text{PDE}} = \underbrace{\frac{\delta}{\tau}\begin{pmatrix} \widehat{p_r} - \widehat{p_l} \\ \widehat{p_u} - \widehat{p_d} \end{pmatrix}}_{\text{RW}},$$

$$D = \underbrace{\begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix}}_{\text{PDE}} = \underbrace{\frac{\delta^2}{2\tau}\begin{pmatrix} \widehat{p_l} + \widehat{p_r} & 0 \\ 0 & \widehat{p_u} + \widehat{p_d} \end{pmatrix}}_{\text{RW}}.$$

These equations show that the advective terms are proportional to the **bias** or **difference** in the movement probabilities, whereas the diffusion terms depend on the **total movement activity** or **sum** of the probabilities in each direction.

Including the flow term

$$f(x,y,t) = \begin{pmatrix} f_H(x,y,t) \\ f_V(x,y,t) \end{pmatrix}$$

with a unit conversion factor α, the model thus reads

$$u_t = \nabla \cdot \left(-\left[\begin{pmatrix} a_H \\ a_V \end{pmatrix} + \alpha \begin{pmatrix} f_H(x,y,t) \\ f_V(x,y,t) \end{pmatrix} \right] u + \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} \nabla u \right).$$

3.3. Simulation Domain and Units

Here we describe the space on which we simulate per model.

For the IB model, the space- and time step size δ and τ are assumed to be constant over the entire domain. We set $\delta = 0.02°$, and although this is quite an idealized estimate, we assume $1° \equiv 100$ km everywhere, so that the model makes steps of 2 km. The variable τ is set to be 1 day. In the absence of a vector field this means each day 2 km is traversed. This 2 km of active swimming per day is a modelling assumption taken from [24]. The amount of steps that must be taken each day with the addition of a vector field is a point of discussion, see Appendix A.

The simulation for a turtle ends when it crosses the horizontal boundaries at $[42.5°, 46.5°]$N, or the vertical boundaries at $[-29°, -11°]$E. As will become apparent in Chapter 5, most of the turtles cross a horizontal boundary.

For the PDE model, the variable x represents a **longitude** in the simulation, and is measured in degrees East. Unfortunately, the datasets do not all utilize the same range for the longitudinal variable, so x can fall within the range $[-29°, -11°]$E, or equivalently $[331°, 349°]$E. This is a width chosen such that (as will become apparent in Chapter 5) only very small densities reach it.

[1] In reality fractional turtles do not exist, but the relation between u the density and the realized positions of turtles from the RW model will *resemble* that of a probability density.

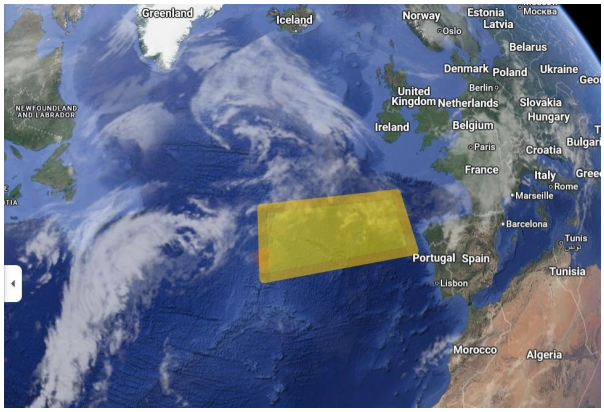

Figure 3.1: A rough sketch of the simulation region. Exiting southbound is considered a success, while exiting northbound is considered a failure.

Similarly y represents a **latitude** in the simulation, which is measured in degrees North. We simulate on the range $y \in [42°, 47°]$, but in the figures in Chapter 5 we only show the density for $y \in [42.5°, 46.5°]$. This difference between simulation and visualization ranges is to minimize the effect the simulation boundaries can have on the solution.

Both x and y are discretised by taking steps of $dx = dy = 0.04°$. This means that the PDE model's simulation grid is a rectangle, subdivided into squares.

Finally, t represents the simulation time. In this thesis, time is usually measured in *days*. Which time ranges are used varies from run-to-run, but certainly to acquire the results in Chapter 5 (unless specified otherwise), we have used t such that[2] the simulation range starts ($t = 0$) at 01-01-2016, and ends when $t = 500$ days. Time progresses in steps of $dt = 0.025$ days, which is equal to steps of 36 minutes. Turtles are introduced to the system for 365 days, the extra time is to let most turtles finish their traversal of the region.

The boundary conditions we used for the PDE model are homogeneous Neumann type for the vertical boundaries, and absorbing type for the horizontal boundaries. In the IBM these boundaries are not present: each agent keeps track of its own location, and when it exceeds either the top or bottom boundary, we assume it to never return to the simulation region.

Regarding initial conditions, to each system we introduce a mass of 2 turtles at the start of each simulation day, at the location $(-25°\text{E}, 44.5°\text{N})$. In the IBM this means we place two agents there, in the PDE model this means we place a density of $\frac{2}{dxdy}$ inside the cell containing $(-25°\text{E}, 44.5°\text{N})$ each day.

[2]This is slightly more nuanced: the timestamps in the vector field data are measured in hours since 01-01-2000. To ease implementation, the time t is implicitly a day count from 01-01-2016, but a bonus time variable is used to track the hours passed since 01-01-2000.

4

Implementation and computation

In this chapter, we will take a look at how the mathematical model proposed in Chapter 2 and further discussed in Chapter 3 are translated to problems computers can solve. We also explain how we approached problems related to implementation, optimization in time and discuss some pitfalls.

As an important note, all the source code to this thesis can be found at: `https://github.com/Hephaestois/Thesis_HRemmen_2025`. We intend for this chapter to reasonably serve as explanation, but the research was never intended to be accompanied by a ready-to-run program, or something necessarily easily adaptable. Therefore, it will not take the form of a step-by-step description with pseudocode, and be more like a justification for the concepts written into the code. Where to find the simulation data, and how this is handled in the programs is also discussed in this chapter. We end the chapter with a short 'handbook' on interacting with the code.

4.1. About the models

As the implementation of the model is a large part of this research, (and just plugging in numbers and discussing results is not interesting without discussing how those results were found), we start by giving an overview of which 'phases' the models have gone through. Only the final versions will be highlighted in more detail, with more detailed explanations about the specific approach, and why some of the choices that were made have been made. By doing this, we hope the reader can appreciate the factors the model accounts for.

For each IB model, a 'repeated' version has also been implemented. These are functionally very similar, but instead of simulating just one walk, they simulate multiple in one go. These will be used for determining similarity between the continuous and discrete model, by means of visual comparison: Does the density created by the PDE model reasonably represent the sample population generated by the IB model? These 'repeated' variants will not be treated separately, as there they are only small variations from the non-repeated models.

4.2. Discrete models, overview

The 'simple' random walk (*RandomWalk.py*)

As a first model, we implement a RW on a 2d grid, with the possibility to specify the probability of moving left and right, and up and down.

To start, we create the class Walker. This will be the class describing the behaviour of the sea turtles. In this first version, it takes a set of probabilities `initProbs`, describing the probability of moving in each direction. The class function `positionJumpProcess(`n`)` has the `Walker` move n steps, according to the defined probabilities, and returns the set of locations which the `Walker` visited. For a sample of a single RW, see Figure 4.1, left. For an impression of how the repeated RW would look, see Figure 4.1, right. Here 500 walkers each started a RW in $x_0 = (-25, 44.5)$. It should be

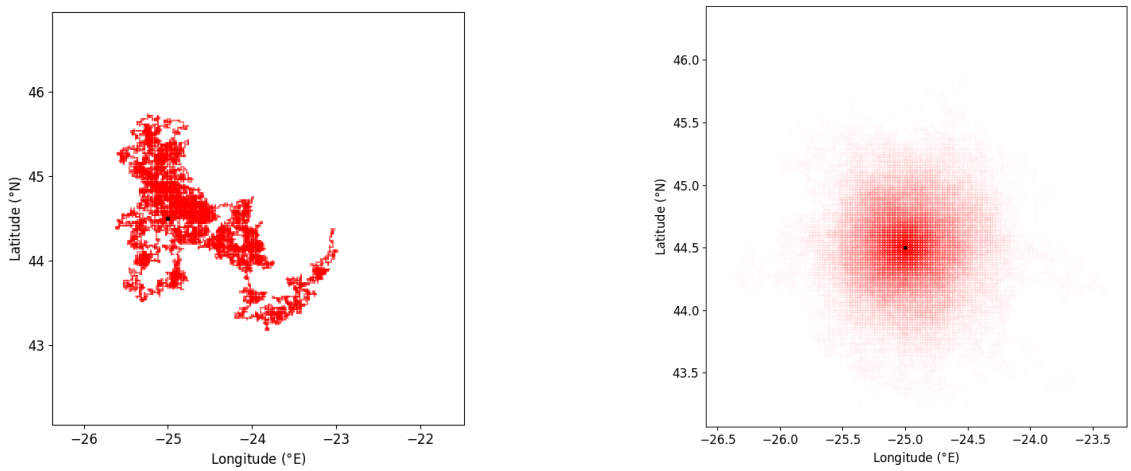

Figure 4.1: Left: A RW starting from $(-25°, 44.5°)$ (marked with a black dot), making $20\,000$ horizontal or vertical steps of $0.02°$, with equal probabilities for each direction. Right: 500 RWs starting from $(-25°, 44.5°)$, each making 1000 steps of $0.02°$, with equal probabilities.

noted that the apparent white-coloured grid in Figure 4.1, right is due to aliasing, and has no further mathematical implications.

The random walk in a time-constant vector field (*RWinCVF.py*)

Because we wish to simulate a RW in ocean streams, we need some way to influence the behaviour of a walker with a vector field. We import our dataset from HYCOM [16] using the netCDF4 package. Then the walker class is adapted with two notable functions, `getRWBiasInField(vectorfield)` and `traverseVectorField(vectorfield,n)`, and given a new set of probabilities, `probs`, which describe the local movement probability due to the vector field.

In Figure 4.2, top and bottom, we can see examples of this RW in a vector field. Note that, in the top image, the probability of moving in a specific direction is now dictated by the vector field alone, but that this is *still a probability*. This is why not all the walks head in the same direction, but do generally all follow the 'flow' of the vector field. In Figure 4.2, bottom, where the agents themself have agency, we can see how this introduces a light downwards bias (there is less exploration at the top), but that the effects of the vector field are clearly dominant.

The random walk in a time-dependent vector field (*RWinVF.py*)

The previous model we described is still missing one part, and it is the vector field changing over time. In the spirit of [24], we want to simulate over a period of 500 days, introducing turtles for the first 365. The changes between this and the time-constant vector field approach are mostly interesting for those following the implementation, and are discussed in the details. Technically speaking, the only change made is that every 24 hours of simulation time, the vector field which is used (as in the time-constant case) is changed to a different field. Otherwise, it is functionally very similar.

Notably for those just trying to use the program, it is at this stage that we decided the runtime exceeded a certain usability bound, and split the python program into one which creates data, and one which plots data. All plotting scripts are under the `/plotting` directory, and in 4.6 we specify how to use them.

4.3. Discrete model, details

Which factors contribute to the direction a turtle moves?

To model the movement of a turtle in the stream, we consider a set of probabilities (p_r, p_r, p_u, p_d), and write components p_i for $i \in \{l, r, u, d\}$. Each of these describes a probability to move left, right, up and down, and are recalculated each time step. The influence of the walker's own probability of movement and the vector field's influence, are both encoded into these four probabilities. The vector field acts in the direction of the flow, and is affected by the magnitude of the flow. If f_i denotes the local components

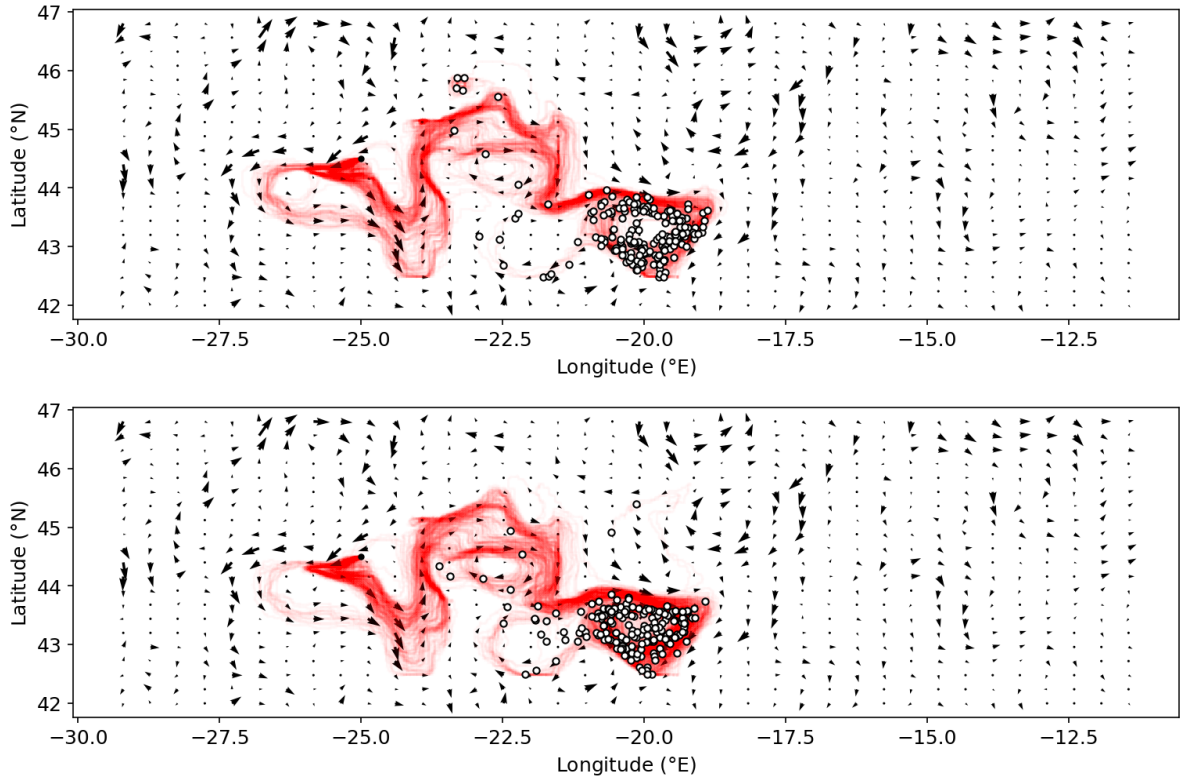

Figure 4.2: 200 RWs starting at $(-25°, 44.5°)$, each moving in steps of $0.02°$. The vector field is sampled at a resolution of $0.08° \times 0.04°$ (lon×lat) at 11-04-2015 (133893 hours from 01-01-2000, selected on visual complexity) and is kept constant throughout the simulation. Top: all the movement is caused by the vector field; the turtles do not move. 1000 steps are made. Bottom: the turtles' swimming behaviour contributes to the movement. 100 days of swimming are simulated, the amount of steps is determined by methods in Appendix A. The final position of the turtles is shown as black-and-white dots.

of the flow in an analogous (f_l, f_r, f_u, f_d) format (where either f_l or f_r is zero, f_u or f_d is zero, all are ≥ 0), and $\hat{p_i}$ the walkers constant bias, then we could write

$$p_i = \frac{\hat{p_i} + \gamma \cdot f_i}{1 + \gamma \cdot \sum_{j=l,r,u,d} f_j} \tag{4.1}$$

for a certain weight $\gamma \in \mathbb{R}$ for the vector fields' influence. This weight needs to be introduced to account for a mismatch in units. To begin with, the units of the flow field are given in m/s (metres per second), so we convert it to °/day by 1 m/s $= 0.864$ °/day. The probabilities $\widehat{p_i}$ represent 2 km/day $= 0.02°$/day, so γ should be divided by this intensity: $\gamma = \frac{0.864}{0.02} = 43.2$.

When a turtle has determined its set of probabilities (p_l, p_r, p_u, p_d) and needs to choose a direction to take a step in, we use the Python `random.random()` to generate a random number in $[0, 1]$. This random number is compared to the cumulative sum of the probabilities.

How are the parameters calibrated?

The IB model simulation uses the units degrees (°) and days (d). We assume that the turtles can do a daily active swim of 2 km, and idealistically assume 2 km$= 0.02°$.

To figure out the direction of the active advective and diffusive behaviour, we assume the turtle finds itself in a zero-flow field. This means that each day, the direction to move in is sampled only from the constant biases $\widehat{p_i}$ (Equation 4.1 with $f_i = 0$). We propose that the von Mises distribution utilized in the velocity-jump process in [24][1] can be integrated over appropriate domains to acquire the probabilities

[1]The von Mises distribution can be thought of as a circular Gaussian distribution. [24] use this to determine the direction a turtle moves in the VJP model. Their parameters in the PDE are statistical results of the von Mises.

in our position-jump process. For example, the probability of moving right would be

$$\widehat{p_r} = \int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} f(x|\mu,\kappa)dx,$$

the integral of the right quarter of a circle. The other probabilities are found by integrating over their respective $\frac{\pi}{2}$ circle arc. The variable x represent the angle at which the Von-Mises is sampled. Doing this operation with the parameters $\mu = 5.0233$ and $\kappa = 0.875$ proposed by [24] using the data collected in [18], we acquire

$$\begin{cases} \widehat{p_l} = 0.1745 \\ \widehat{p_r} = 0.2817 \\ \widehat{p_u} = 0.0994 \\ \widehat{p_d} = 0.4444, \end{cases} \tag{4.2}$$

whose bias $\begin{pmatrix} \widehat{p_r} - \widehat{p_l} \\ \widehat{p_u} - \widehat{p_d} \end{pmatrix} = \begin{pmatrix} 0.108 \\ -0.345 \end{pmatrix}$ indeed points in a south-south-easterly direction (and whose sum $\sum_{i=l,r,u,d,} p_i = 1$ indeed). This is the set of constant probabilities that every turtle uses.

How to incorporate a flow field into the simulation?

The goal is to simulate a walker acting inside a vector field. There is a subtle problem here, related to the distance a walker could traverse with and without the influence of a vector field. As this requires a rather mathematical explanation, we have opted to include this in Appendix A.

Using the method described in Appendix A, it only remains to integrate the flow data into the model. This is done by a weighted sum as described in Section 4.3.

Some notes about acquiring ocean flow data

The data for the time-changing vector field is collected by the HYCOM consortium [16] (HYbrid Coordinate Ocean Model), and is acquired from `https://tds.hycom.org/thredds/catalog.html`. Specifically, the data used in this paper is from the GOFS 3.1: 41-layer HYCOM + NCODA Global 1/12° Analysis (NRL). This set contains data for the time period July 1st 2014 through to September 4th 2024, broken into periods of approx. 1.5 years. The data is used only at a depth level of 0 meters, as the hatchling turtles can not dive deep enough to consider their depth displacement impactful, see [8].

It is also appropriate to notify future researchers that the 'time' variable in the datasets we used does not always have a consistent step size. It is measured in hours since 01-01-2000, and usually has a step size of 3 hours. However, some data points are missing and have been simply removed. This sometimes leaves a 6-hour gap, so letting the time variable progress by steps of 8 data points ($8 \cdot 3 = 24$ hours) will not always result in time steps of 24 hours. In this thesis this is solved by keeping track of an independent time variable, moving in multiples of 24 hours. Then if at any simulation step this data point happens to have been deleted, we select the most recent, earlier data point. There is a point of consideration here that has to do with the tides, and it will be discussed in Chapter 6.

What if a turtle exceeds the simulation bounds?

Recall that exceeding the simulation bounds is precisely what we are after. In [24], a 'success measure' is introduced to analyse how 'well' the turtles have navigated the region. This measure is stated as

$$p_{\text{success}} \text{ at time T} = \frac{\text{\#Turtles who reached success boundary by time T}}{\text{\#Turtles who reached success/ failure boundary by time T}} \tag{4.3}$$

Or phrased differently: Of the turtles (density) who swim outside of the range 42.5°-46.5°N, which fraction exits on the 42.5°-boundary or "continue in the NA Gyre?". When any boundary is reached, the IB model updates a global counter for overflow through the top or bottom, depending on where the turtle left. That turtle's simulation then stops. After introducing turtles to the system for 365 days, we continue simulating until day 500, as most turtles have finished simulation by then.

Some notes about optimization

The throughput of data required for the time-changing vector field is not spectacular (only a few megabyte per iteration of the vector field), and is therefore not an interesting constraint for the programming side

of this thesis. One factor that deserves mention is that, until changing from a time-constant to a time-dependent vector field, it has been possible to simulate one turtle through all its steps one at a time.

When the vector field in which the turtles are simulated changes every day, it makes more sense to only progress through each day only once, to reduce the amount of times a new flow dataset needs to be loaded. This means all the turtles need to be simulated at the same time, so that they can work with the same data for the vector field. If the dataset is streamed remotely (so there exists no copy of the data on the machine the simulations are being ran on), this makes all the more sense, due to server response times and bandwidth limitations. This remote approach has proven itself feasible: in our programs, the flow datasets do not need to exist locally to utilize the flow data.

4.4. PDE model, overview

We have seen the implementation of the IB model based on a position jump process, and wish to compare results between that model and a PDE based model, we derived from it in Chapter 2. Therefore we also implement a PDE based model. As is more common for use in PDE's, we replace the variable $p(x, y, t)$ in Equation 2.9 by the variable $u(x, y, t)$, representing the turtle density.

The approach in this paper is by the Finite Volume Method (FVM) [29]. To achieve this, we discretize our space with small surfaces $dx \times dy$ in the 2d domain and approximate the function $u(x, y, t)$ with a matrix of $u_{i,j}(t)$ evaluated on these surfaces, so that we can use numerical methods to solve it.

The advective and diffusive terms are handled in three separate parts. This is possible as the equation we are solving u for,

$$\begin{aligned} u_t &= \nabla \cdot (-A(x, y, t)u + \nabla Du) \\ &= \nabla \cdot (-a - \alpha f(x, y, t))\, u + \nabla \cdot \nabla Du \\ &= \nabla \cdot (-au) + \nabla \cdot (-\alpha f(x, y, t)u) + \nabla \cdot \nabla Du \end{aligned}$$

allows solving the advective and diffusive terms separately of each other and combining them. Moreover, there is a space- and time constant advection and diffusion caused by the behaviour of the turtle, a and D, and a space- and time dependent advection caused by the influence of the ocean stream, $f(x, y, t)$ with a unit-conversion α. So together, we solve for three different components: the constant advective and diffusive terms, and one non-constant advective term.

4.5. PDE model, details

How is the time progression handled?

The time variable has been discretised in steps $dt = 0.025$. To model the progression of time, we use Forward Euler. As stated before, the density function u is approximated by a matrix (NumPy array). This is slightly more nuanced in implementation, as we define NumPy arrays `u_old` and `u_new`, and when applying Forward Euler in time, calculate `u_new←u_old+dt*flux(u_old)`. After a day of simulation time passes we assign `u_old←u_new`, and this process repeats. This process allows us to calculate the fluxes one by one from `u_old`, add them up into `u_new`, and step time in one go. This has a positive impact on the possibility to optimize the code, as only the flow field $f(x, y, t)$ is a non-constant influence on u, and can be calculated independently of the influence of D and of the constant advection a.

How is the ocean flow vector field handled?

After we acquire the data from HYCOM [16] each simulation day (see section 4.3), we create two matrices in our simulation, f_H and f_V, the horizontal and vertical components of the flow field. It is important to mention again that the density function $u(x, y, t)$ is approximated by a matrix $(u_{i,j}(t))$.

When we do this discretisation, we need to rethink in what sense f_H and f_V act on u. For the sake of example, consider in just one dimension the pure advection equation we derived in 2.2 with an advective term depending on space $a(x)$. This represents the vector field.

$$u_t(x, t) = \partial_x\,(-a(x)u(x, t))$$

In the coming subsections we explain that we will use an upwind scheme (and what that means). For simplicity let $a > 0$, so we just use forward differences here. That means that the derivate should be

evaluated numerically by

$$\frac{u(x,t+dt)-u(x,t)}{dt} = -\frac{a(x+dx)u(x+dx,t)-a(x)u(x,t)}{dx}$$

So a works on u as a scalar, evaluated at the same location. When having approximated u and a into matrices, this means that a and u should be multiplicated element-wise.

Returning to our original question, considering this element-wise multiplication requires f_H, f_V and u are matrices of the same shape, we sample f_H and f_V at the locations where u is defined. This means that corresponding indices in the matrices f_H and f_V and u represent the same physical location in our simulation space.

Because the ocean flow dataset is not aligned to our simulation grid, we must do some form of interpolation. For performance and simplicity, we use a nearest-neighbour approach. The step size in the data is $0.08° \times 0.04°$ (lon×lat or $dx \times dy$), and our (PDE) simulation never use resolutions for the space discretisation finer than $dx \times dy = 0.04° \times 0.04°$. By choosing this resolution, longitudinally each point is (over)sampled twice, and longitudinally sampled just once. A finer resolution would start over-sampling heavily, using certain data points more often, while a courser resolution would not capture all the information provided by the flow field. Because the derivation in Chapter 2 requires the spatial step size δ is constant and the same in both directions (so $dx = dy$), we choose $dx \times dy = 0.04° \times 0.04°$.

How are the parameters calibrated?

Another point of interest is how the entries for a and D, representing the active advection and diffusion terms in the PDE model, which are both constants, are found. First we determine the constant advection a. From Section 2.1 and Section 4.3 we know that the elements in $a = \binom{a_H}{a_V}$ should be $a_H = \frac{\delta}{\tau}(\widehat{p_r} - \widehat{p_l}) = \frac{0.02}{1}(0.2817 - 0.1745) = \frac{2}{100}(0.108)$ and $a_V = \frac{\delta}{\tau}(\widehat{p_u} - \widehat{p_d}) = \frac{2}{100}(-0.345)$:

$$a = \frac{2}{100}\begin{pmatrix} 0.108 \\ -0.345 \end{pmatrix}$$

Using the same method we can also find the elements of $D = \begin{pmatrix} c_H & 0 \\ 0 & c_V \end{pmatrix}$ with $c = \frac{\delta^2}{2\tau}$ and $H = \widehat{p_l} + \widehat{p_r} = 0.4562$, $V = \widehat{p_u} + \widehat{p_d} = 0.5438$ to be

$$D = 10^{-5}\begin{pmatrix} 9.124 & 0 \\ 0 & 10.876 \end{pmatrix}$$

Now including the strength of the vector field to the equation, we need to convert m/s to °/day, for which we introduce conversion factor α. This conversion factor should be 0.864, as 1 m/s $= 0.864$ °/day. This means that, with all our parameters filled in, the PDE we solve is

$$u_t = \nabla \cdot \left(-\left[\frac{2}{100}\begin{pmatrix} 0.108 \\ -0.345 \end{pmatrix} + 0.864 \begin{pmatrix} f_H(x,y,t) \\ f_V(x,y,t) \end{pmatrix} \right] u + 10^{-5}\begin{pmatrix} 9.124 & 0 \\ 0 & 10.876 \end{pmatrix} \nabla u \right) \tag{4.4}$$

How are the numerical schemes implemented?

The different parts in Equation 4.4 should be solved with different numerical methods. It is often not entirely clear how these numerical methods are able to translate difference operators into operations computers are able to do, and we feel this line of reason is sometimes obfuscated behind mathematical notation in most literature. It is for this purpose that we have added a tangent on the topic into Appendix B, which we hope can provide some insight into the more computer-oriented side of the story.

How are the boundary conditions handled?

The vertical homogeneous Neumann boundaries at $-11°$E and $-29°$E are handled using ghost nodes on the boundaries (see for example [30]). The absorbing boundaries are implemented at $42°$N and $47°$N, where mass can flow out of our simulation region, but not back in. In Chapter 5, the figures show the solution in the range $[42.5°, 46.5°]$ to minimize the effect the boundaries have in these.

Due to the choice of absorbing boundaries on the horizontal sides, we have not been able to determine how to *also* measure the out-flow in the PDE model. Calculating the downward flow of density

over the horizontal $42.5°$N, and subtracting the upward back-flow is maybe viable, but the scope of this paper has not permitted this being implemented.

4.6. Some comments about using the code

The more experimentally inclined reader might want to either re-run our models, or to just play around with them. This is a great attitude, and we encourage interaction with the code-side of this thesis, which can be found at `https://github.com/Hephaestois/Thesis_HRemmen_2025`. For those daring people, this section exists.

The file structure

The project is structured so that different topics reside in different directories. That is, there are directories for: The IB model (`/discrete_model`), the PDE model (`/pde_model`), a globally shared library (`/library`), a shared data reading/writing location (`/data`), files related to plotting images (`/plotting`) and some directories to capture output from the plotting files.

All the data in the programs flows through `/data`: there is no (python) file that runs a model *and* plots the related figures; they need to be called separately. This seems tedious, but permits greater levels of automation and flexibility when handled properly. It also permits collecting data once, and (re)making figures many times over.

Part of the need for separation is how the programs handles different years: we never implemented a shared function that takes a year and day and returns the appropriate flow dataset for that day (which would be a great tool for a future researcher to create!). This non-existence means that each year gets its own file, with the appropriate dataset calls hardcoded[2] in. This yearly separation unfortunately also means that some tweaks (different parameters or just a bugfix) have to be made in a separate file for each year. All the files share a library `/library`, which contains structures that are common between the years, such as the IB models' `walker`, or the PDE models' `grid`, which is where the FVM is implemented. The file `functions.py` contains some useful functions for saving and loading data, as well as the highly-recommended-for-use function `progressBar.py`.

Running python files

As mentioned, the file structure and separation *does* permit a relatively streamlined workflow, provided you do not need tweaks to the python files. Each python file gets a header so that they can find data and library files. Also using `sys.argv`, the scripts can be called from the terminal with the following signatures (for a specific year 2016-2023):

```
PDE : $ python pde_model/FVM<year>.py <sim_days> <dx> <dy> <dt>
IB  : $ python discrete_model/RWinVF-<year>.py <sim_days> <n_per_day>
```

with 'sim_days' the amount of days to simulate for (usually 500) and 'n_per_day' the amount of agents to release per day in the IB model.

Doing this, the appropriate model starts to run, and saves the data it generates to a location in `/data`.

Automation

Because the models take on the order of tens of minutes to run, multiple runs should be automated (to preserve sanity). This can be done by making a Bash script (see [12]), and running the models sequentially. Because all data streams through the `/data` directory, such a workflow is possible. An example of a plotting routine with newly generated data might be:

```
#!/bin/bash
ndays=500
dx=0.04
dy=0.04
dt=0.025
nperday=2
```

[2] Hardcoding is the act of setting parameters directly in code, instead of leaving them dependent/variable by a user. It simplifies interacting with the program, but customizability suffers for it.

```
year=2016
frame=100
mode=both
offset=0

python "discrete_model/RWinVF-${year}.py" $ndays $nperday
python "pde_model/FVM${year}.py" $ndays $dx $dy $dt

python plotting/plot_image_fine.py $year $ndays $dx $dy $dt $nperday \
    $offset $frame $mode
python plotting/plot_video_fine.py $year $ndays $dx $dy $dt $nperday \
    $offset $mode

exit 0
```

There are some variables that warrant clarification:

1. `frame` is the day on which a plot should be made.
2. `mode` tells the plotting program which data to include. It can be 'rw', 'pde' or 'both'. When only the results from the IB model are plotted, trails are added to better visualise the steps made. In other visual modes these trails are too distracting.
3. `offset` is a value used to convert longitudes. Because some years use °E in $[-180°, 180°]$, others use $[0°, 360°]$. This offset makes sure that the plotting program can handle either of those. As a rule of thumb, 2016 needs offset=0, the other years need offset=360.
4. The use of `'fine'`: this is another pitfall of hardcoding. Due to the output of the PDE model being total value inside a surface area, not value per surface area, taking finer resolutions requires different graphical options for the density map. The two resolutions that are supported by default are $0.1° \times 0.1°$ and $0.04° \times 0.04°$, the latter one making use of the 'fine' plotting files. The values of dx and dy could be used to determine these graphical options, but we didn't spend the time to figure them out.

The truly daring might even use the `&` operator in their automation routine to parallelize the processes. This works great for plotting, but has not been tested on running the models, as these stream their data over the internet.

We hope that this small handbook will aid any researchers trying to validate our research. There are a few directions in which we could point development of these files to further ease usage, but it might be debated how much time will be saved by improving the code compared to the time invested to modify it. Regardless, this will be discussed in Chapter 6

5 Results

In this thesis, we have so far implemented two models, in an attempt to reproduce the results in [24] with different techniques. In the first two sections of this chapter, we would like to discuss the results from our models on their own, and then move on to our original goals:

1. Can we say our modelling approach provides mutually consistent results?
2. Can we conclude something about loggerhead hatchlings' survival rate in recent years based on new ocean stream data?

We also compare our results with [24], to show similarity. Before showing the result of the model with all factors considered, we wish to show some intermediate states of each model. All the relevant figures are placed at the end of this chapter. The effects of the flow fields are seen clearest when looking at a time progression of these images. For this purpose, videos have been posted to `https://www.youtube.com/playlist?list=PLdvQXQjD1hC0QMlMZs_tKMkSwqwe30PDN`

5.1. The individual based model

For the IB model, before introducing the vector field's contribution, we show the effect of the constant advective and diffusive terms.

In Figure 5.1, we can see three results. Each figure is the snapshot at the end of day 100, with the paths the agents took to get to that position. The top figure shows only constant diffusion, the middle figure shows only constant advection and the bottom figure shows the combined effect of these two. It should be noted that, because the active movement is calibrated to always contribute 2 km, separating the effects of advection and diffusion yields a stronger diffusion and advection: they would normally work against one another, whereas now in the separate figures, their effects are amplified. When they are combined, their relative contributions balance out to their preferred values. The probabilities used in Figure 5.1 are special, and described in the caption. In all the other figures in Chapter 5, we use the probabilities we found in Equation 4.2. Figure 5.1, bottom, is most representative of the behaviour we expect to see in still standing water, with shorter, somewhat spread out walks. When we add a flow term, we expect that the distance travelled per day changes drastically, as flowing along a stream does not take 'effort' from the turtles.

So, including the vector field, a sudden explosion of complexity occurs. In Figure 5.2 we see three figures showing a sample[1] of the IB model. We can see complex behaviour arise from the rules the turtles have to obey. It's important to mention here that, although the turtles still act on a grid, due to the way the amount of steps in a day is determined their day-to-day movement does to be appear off-grid: their paths (which are drawn between the start and end of a day) may run from non-neighbouring cells, thus creating lines between non-neighbouring vertices in the grid.

[1] Because the IB model is based off of probabilities, each run yields a different population. The IB model is ran *once* for all three figures.

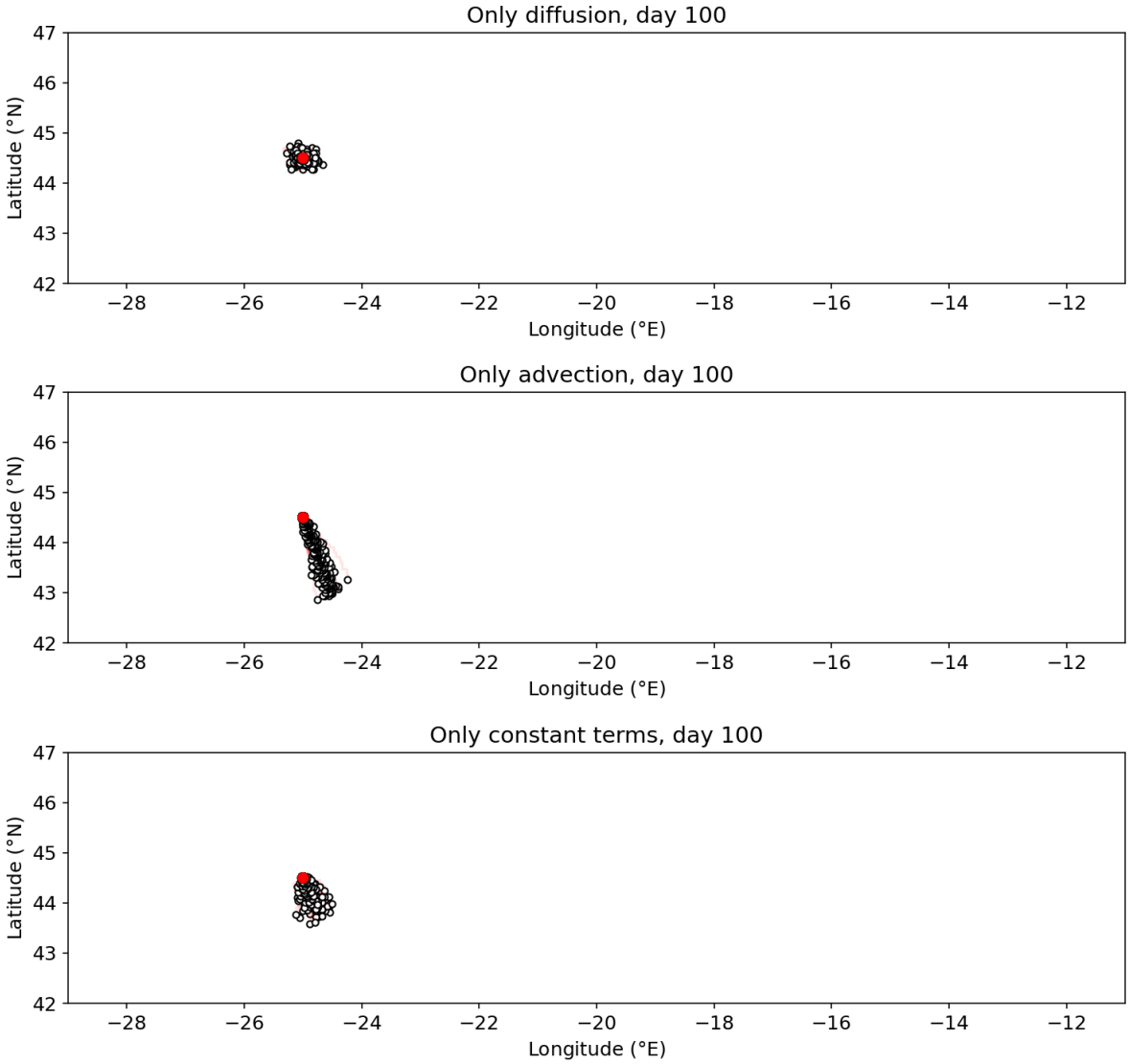

Figure 5.1: Three samples (each after 100 days of simulation) of one run of the IB model. In each sub-figure the step size is $0.02°$ in both directions, and each day two turtles are placed at $(-25°\text{E}, 44.5°\text{N})$, which is marked in red. Top: only the diffusive terms contribute to movement: $(\widehat{p_l}, \widehat{p_r}, \widehat{p_u}, \widehat{p_d}) = (0.2281, 0.2281, 0.2719, 0.2719)$, where $\widehat{p_l}$ is the probability of moving left, etc. Middle: only the active advective terms contribute to movement: $(\widehat{p_l}, \widehat{p_r}, \widehat{p_u}, \widehat{p_d}) = (0, 0.2384, 0, 0.7616)$. Bottom: Both the active and diffusive terms contribute to movement: $(\widehat{p_l}, \widehat{p_r}, \widehat{p_u}, \widehat{p_d}) = (0.1744, 0.2817, 0.0994, 0.4444)$. Each image is captured at the end of simulation day 100. The black-and-white dots indicate the position of turtles at this time, the thin red lines indicate the paths they traversed.

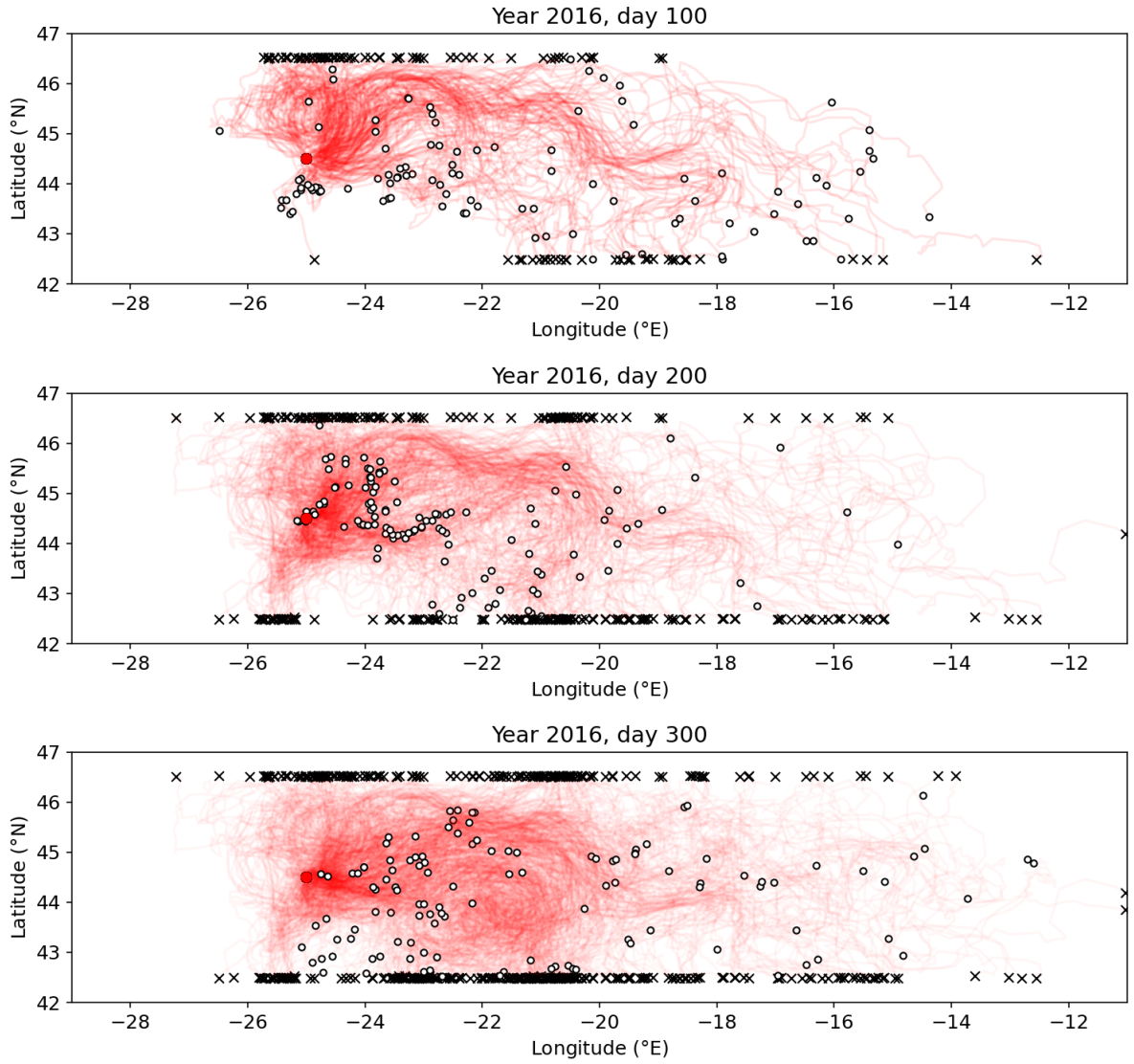

Figure 5.2: Three snapshots (day 100, 200, 300) of one continued run of the IB model starting on 01-01-2016. In each sub-figure the step size is 0.02° in both directions and the constant probabilities for active movement are $(\widehat{p_l}, \widehat{p_r}, \widehat{p_u}, \widehat{p_d}) = (0.1744, 0.2817, 0.0994, 0.4444)$. Each image is captured at the end of the simulation day. The black-and-white dots indicate active (moving) turtles, the crosses indicate those turtles who have reached a simulation bound and whose simulation has therefore stopped. The path each turtle traversed is drawn in red.

5.2. The PDE model

For the PDE model, we can start with a similar approach, by including first only the constant diffusion or only the constant advection, combining their effects, and then showing the full picture with also the influence of the vector field.

In Figure 5.3 we can see the PDE model with these constant factors singled out: only the active diffusion, active advection, or both. As opposed to the IB model, the influence of the diffusive and advective operators is separated naturally, instead of implicitly in the probabilities. This means that the diffusion portrays the behaviour we would expect from the diffusive parameters more accurately: note how the high-density part is a lot smaller than in Figure 5.1, top. The spread of density above 0.001 is relatively large, but diffusion spreads a low amount of mass out very fast, while the massive peak is mostly stationary. The active advection is seen in Figure 5.3, middle. This figure looks very sharp, due to the nature of advection: it does not permit flow against the stream, so the source is the furthest left and up our population can move. This is fine, as the against-the-bias flow is handled by the diffusion. The third figure shows their effects combined, and as we hope, shows a very similar picture as in Figure 5.1, bottom.

In Figure 5.4 we can see three snapshots of the PDE model with the flow field included. For these images, we also overlay the flow field itself. Each image shows the density at a certain moment in time, and the density is cut-off below a value of 0.001, for better visibility of the resulting data.

5.3. Comparison between the models

With these two models, we can compare their performance, and determine whether they provide similar results. To do so, we gather data from both models and combine their outputs into singular figures. Since the PDE model yeilds population density functions which evolve over time, while the IB model creates individual walks which should be a sample from this population, we compare snapshots of the density functions overlaid with the positions of walkers at corresponding time steps. If these visually coincide, we consider the models to be producing similar results.

We first turn our attention to the results in 2016. Running our model on a spatial resolution of $0.04^\circ \times 0.04^\circ$, and a time step size $\tau = 0.025$ day, we acquire the results in Figure 5.5. Here we can see a good correlation between the spread of swimmers (white dots from the IB model) and population density (PDE model) shown as a colour map. While the models are not in perfect agreement, as the highly dense region in the top figure and the swimmer population seem misaligned, there is clear visual similarity.

We are now also in a position to compare our models with those created and implemented in [24], who modelled the same phenomenon. Their PDE model appears to capture more intricate details of the flow fields, and our models appear to move upwards more strongly. In their IB model the agents remain more closely together, especially near the source or high-density regions, where our agents are usually spread out further. These observations are discussed in Chapter 6.

Turning attention to the survival rates we find, in 2016 with 2 km of active swimming per day, our IB model (with 5 turtles introduced per day) yields a survival probability of 0.5505. This is significantly lower than the 0.78 in Figure 11 of [24] (left sub-figure, red dashed line at '2 km/day'). Despite this difference, the relative values and trends over time remain valuable for statistical analysis.

Notably, it seems that there is a difference in how [24] and we acquire our flow data, as they are visually dissimilar. We further discuss this in Chapter 6

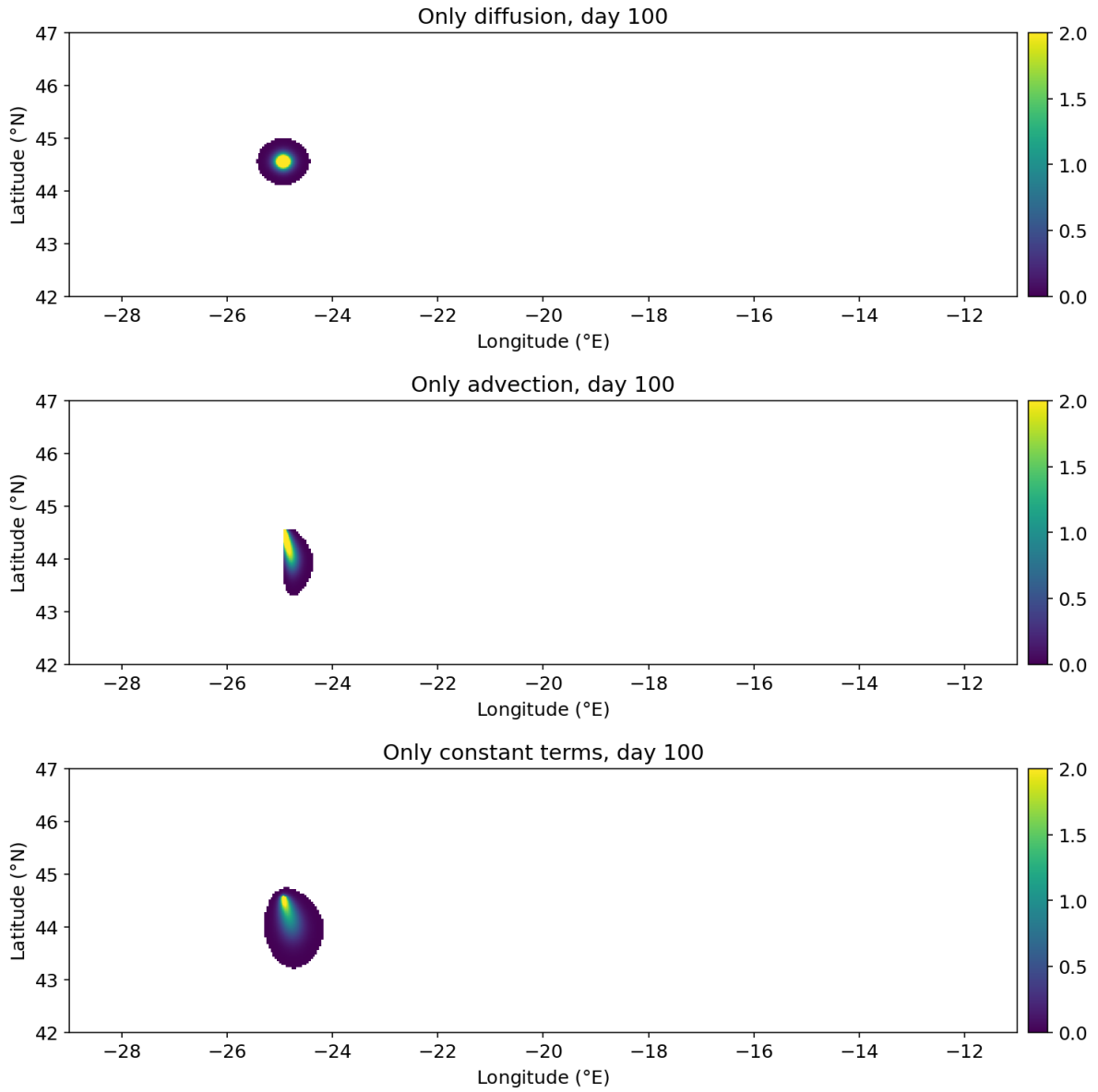

Figure 5.3: Three samples (each after 100 days of simulation) of the PDE model. In each sub-figure, $dx = dy = 0.04°, dt = 0.025$ days. Top: only diffusion affects the simulation. Middle: only advection due to active turtle movement. The hard edges are due to the advection acting by itself: there will movement against the direction of the flow, so from the source term we only see south-easterly movement of the population. Bottom: The combined effect of constant advection and diffusion. In each figure, the values used are as in Section 4.5, and the advection due to a flow field is *not* present. The density is only shown for values larger than 0.001 per cell for improved readability.

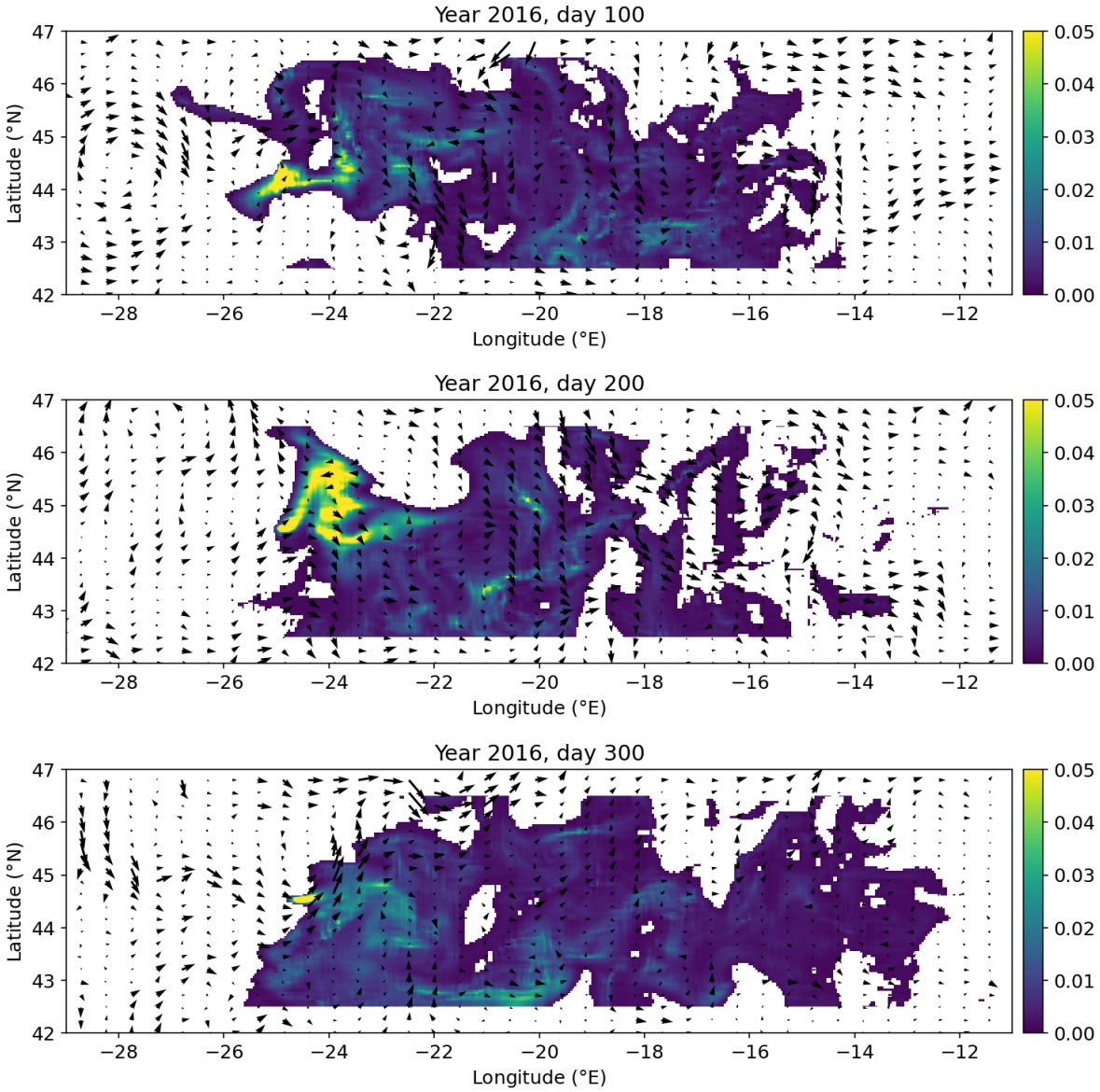

Figure 5.4: Three samples (day 100, 200, 300) of one run of the PDE model starting on 01-01-2016. In each sub-figure $dx = dy = 0.04^{\circ}, dt = 0.025$ days. The advective and diffusive terms are as in Section 4.5, with a flow field influencing the advective term. Each image is captured at the end of the simulation day. The density is only shown for values larger than 0.001 per cell. In this figure, the flow field we are working in is also included.

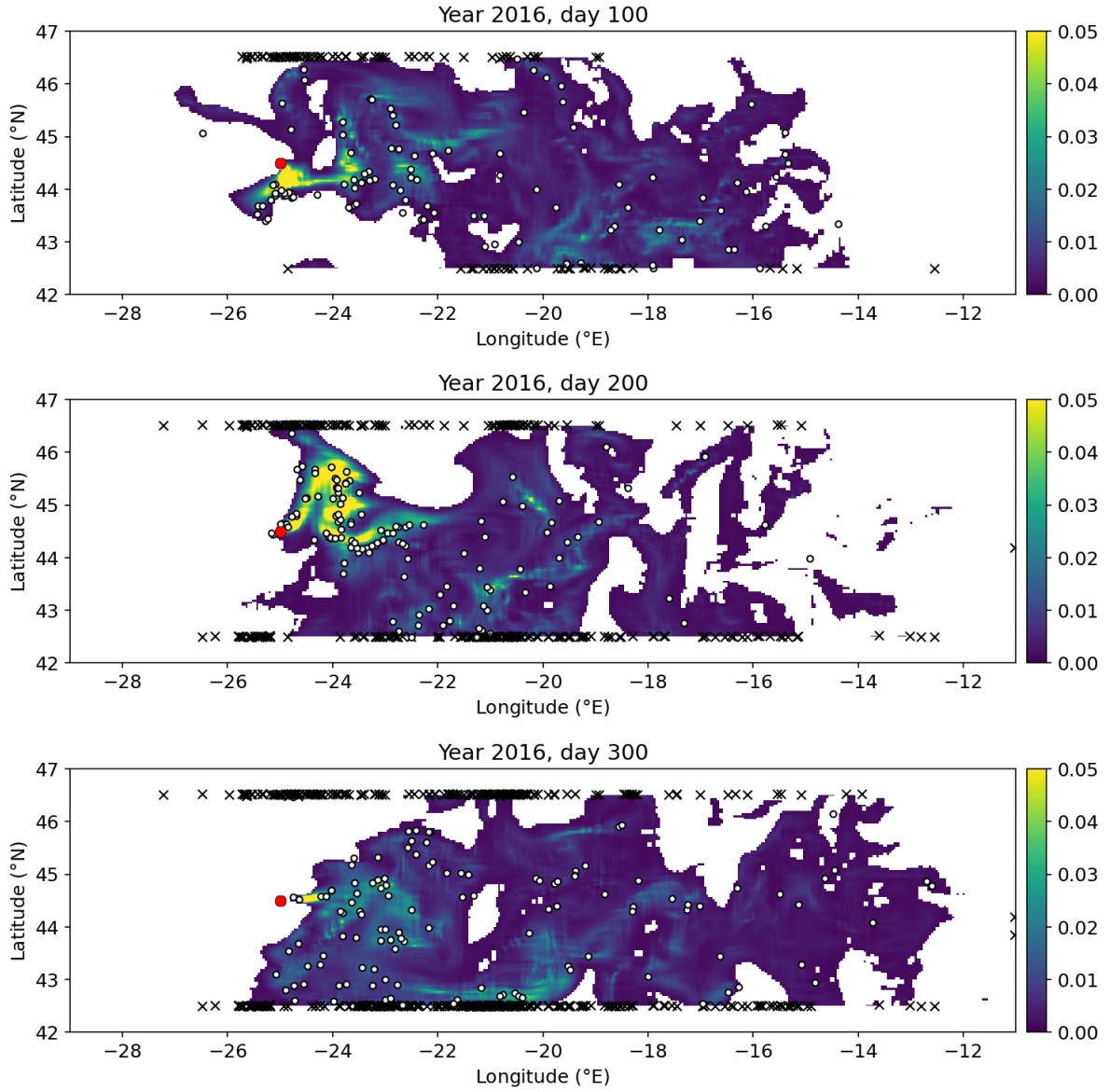

Figure 5.5: Three samples (day 100, 200, 300) of one run of the IB and PDE model starting on 01-01-2016, combined into three sub-figures. In each sub-figure for the PDE model $dx = dy = 0.04°, dt = 0.025$ days, for the IB model the step size was $0.02°$, with constant probabilities as in 4.3. The parameters for the PDE are as in 4.5, and both models are affected by the same flow field. Each image is captured at the end of the simulation day. The density is only shown for values larger than 0.001 per cell.

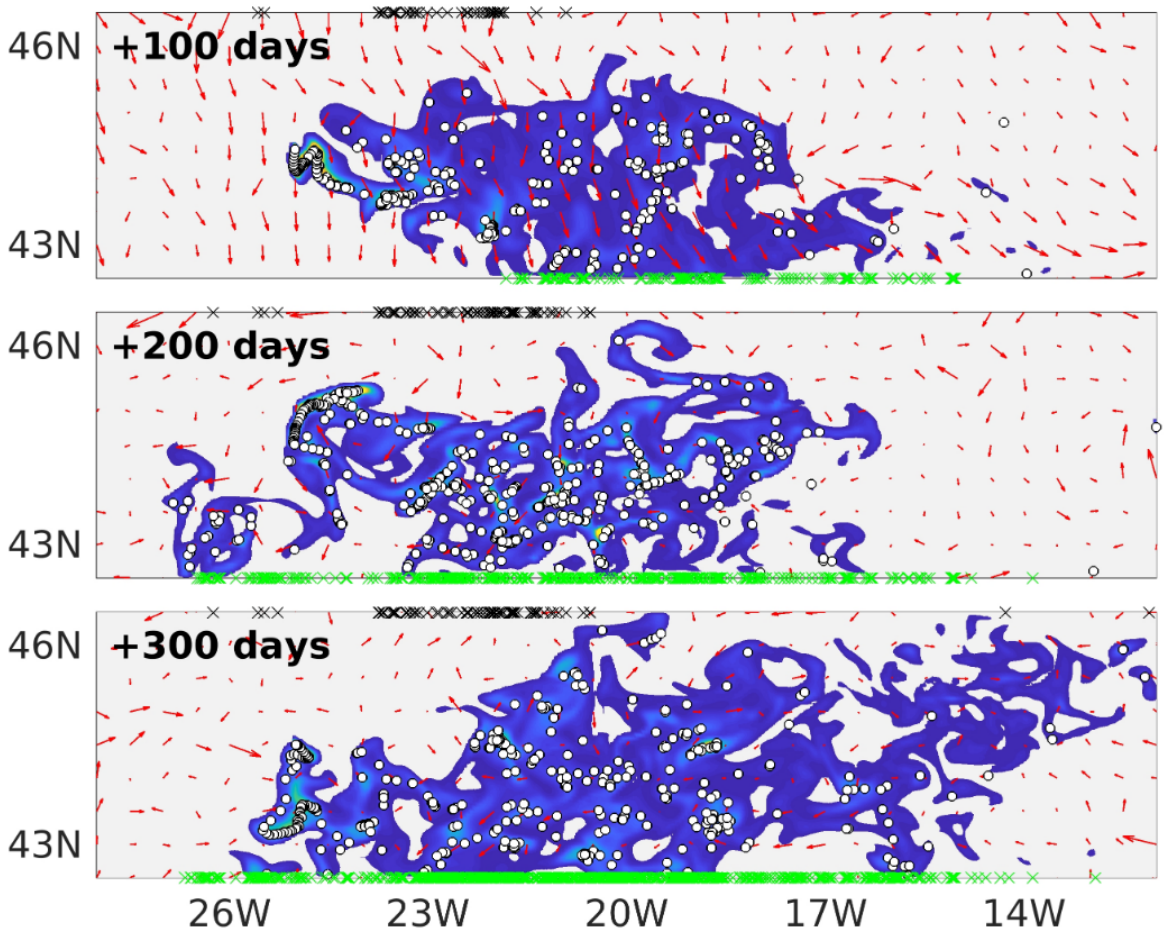

Figure 5.6: Three sample of the models [24] created to model the same experiment as in Figure 5.5. This image is a copy of Figure 9 in [24], and has been included for comparison purposes. They do not provide a legend for the population density

5.4. What will happen to the loggerheads?

Considering the impact climate change can have on the ocean streams [21], we hypothesise that in recent years, the survival probability p_{success} might have changed.

To determine this, we use a success function inspired by [24]: the amount of turtles exceeding the bottom boundary of the simulation zone divided by the amount of turtles exceeding any horizontal boundary of the simulation zone, as generated by the IB model[2]. Formulated it reads

$$p_{\text{success}} = \frac{\text{Amount of turtles subceeding } 42.5^{\circ}\text{N}}{\text{Amount of turtles exceeding } 46.5 \text{ or subceeding } 42.5^{\circ}\text{N}}. \tag{5.1}$$

For the year 2016, utilizing the IB model to determine these probabilities, we acquire a success probability[3] of $p_{\text{success}} = 0.5505$. Continuing to run our model for subsequent years, we find the results in Figure 5.7. Here, year 2017 does not have a data measurement.[4]

There are two trends here that deserve further investigation: the valleys seen in 2019 and 2023, and the apparent upwards trend from 2016 to 2022.

The valleys in 2019 and 2023 could have a number of causes, which we discuss in Chapter 6. Because they appear to be statistical outliers (2019 deviates 203 turtles from the average of other years, 2023 deviates by 335 turtles), or at least influenced by some other events, we ignore them for determining p_{success}. Doing this exception, we see an upwards trend in the data in years 2016, 2018, 2020, 2021 an 2022, and we would like to determine whether we may statistically speak of a trend, or whether the data lies within expectable variation.

We assume the data is binomially Bin$(N = 1825, p_{\text{success}})$ distributed. That is, on a yearly basis, each of the 1825 turtles crossing the region has an independent probability of successfully reaching the bottom p_{success}. We can formulate a null hypothesis H_0, the statement we are trying to show to be true or false [2]:

$$H_0 : p_{\text{success}} \text{ is a constant value.}$$

With this H_0, we can apply some statistical methods described in Appendix C. In short, for each possible probability p_{success} we determine a p-value for each measured outcome of the binomial process. That is, we determine for each observation *how probable* it is that we would make observations as extreme as in Figure 5.7, under the assumption these results are drawn from a Bin(1825, p_{success})-distribution. We then combine these p-values using Fisher's method [11] into a single overall p-value.

In this way, we find that our null hypothesis H_0 has a p-value of $2.62 \cdot 10^{-6}$, and we can therefore reject (using most conventional levels of significance) the notion that the survival probability is binomially distributed by a constant probability of success.

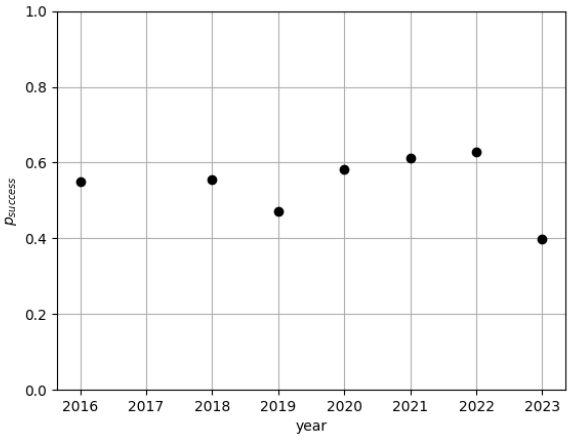

Figure 5.7: A plot of the sample values for p_{success}, as a function of the year in which it is being sampled. The anomalies in 2019 and 2023 are discussed in Chapter 6, and largely considered outliers for the statistical analysis; they are excluded from the data.

[2] Of course, it would be best to use both the IB and the PDE models' results, but there are some limitations to using the PDE model's density that this thesis has not been able to overcome. These are discussed in Chapter 6

[3] Note that the turtles who exceed the vertical borders or never reach a horizontal border do not throw off the statistic: the measured p_{success} is calculated precisely as in Equation 5.1. We assume the turtles that didn't finish simulation would have contributed to the probability in the same way.

[4] In the year 2017, longitudinal measurement swapped from the range $[-180°, 180°]$ to $[0°, 360°]$. Although the flow data is therefore technically available, we did not expect this switch between 'units' to happen. It is for technical limitations of our implementation that we had to exclude 2017 from the simulations.

6

Final thoughts

In this chapter, we cover three main topics: what differences we can see between our models and between our models and the literature, how we can improve the models themselves, and how we can extend the work presented in this thesis overall. We end by concluding the thesis.

6.1. Discussion about results

Before noting where our models differ, we should mention that we believe to have succeeded at our goal of implementing two models that agree with each other, as well as with previous attempts in the literature to model this phenomenon. We still think it valuable to discuss discrepancies between the models and the literature, and what might have caused them.

Most notably, we want to talk about the observation of walkers generated by the IB model being *close to* the densities generated by the PDE model (see Figure 5.5), but not quite overlapping: the PDE density and the IB population seem *close*, but not quite convincingly so. There is always the possibility of off-by-one errors in which day is plotted, but assuming the implementation is correct, a likely reason for the difference may be a mathematical problem to do with determining the amount of steps a walker makes. As discussed in Appendix A, the IB model likely permits the walkers to travel more distance due to swimming when the influence of a flow field is added to the model.

Another difference we noticed between our and [24]'s outcome is that our models show an upward bias compared to theirs, with our density often crossing the $46.5°$N boundary–where theirs rarely reaches it. The IB model appears to have the same bias, as it agrees with the PDE, and produces a substantially lower survival rate. We suspect this bias is due to the discretization of our space, together with the assumption that $1° \equiv 100$ km. This is further discussed in Section 6.2.

As a last observation, where in [24] turtles are very close together in areas of high density, our turtles are more scattered. We think this is a symptom of using a PJP: where a VJP can move in a precise direction, the PJP can only ever move along the axes. So to move in an off-axis direction, our PJP needs to combine steps in two directions. We hypothesize that this necessity creates a sufficient amount of noise in the movement pattern to reduce the likelihood of clusters occurring.

What happened in 2019 and 2023?

The survival rates in 2019 and 2023 seem to follow the pattern in the data poorly, so we look for some causes that might explain these outliers: what happened in or during 2019 and 2023 that didn't influence other years? Unfortunately, thoroughly researching the causes to this observation is far outside the scope of this research paper, but this is a clear area of continued research. Incorporating older flow data[1] might uncover larger-scale patterns related to successful migration and provide the statistical power to link global weather events with the observed probabilities.

Speculating about the causes to the observation, [31] supports that the ocean streams in the North Atlantic are changing in strength over time. This aligns with our result, as the probability of traversal

[1] HYCOM not only provides until 2024 now, but also back to 1994. Not using this older data was also a decision made in the context of scope

should remain constant with a non-changing gyre, which it isn't. Additionally, according to the Copernicus Climate Change Service [6], 2023 saw a sea surface temperature anomaly of $+1.36^\circ$ above average. Although 2016-2022 were comparatively unremarkable years, this deviation could also have had an indirect impact on the ocean streams; while temperature is not a direct factor our model considers, it may have an influence of the direction of ocean flow, thus survival probability. Considering this topic is so broad, we recommend this for further research to better understand the environmental cues influencing turtle migration success.

6.2. Discussion about our methods

Throughout the thesis, some points of interest have been marked for further discussion. We go through those topics here, as well as provide thoughts on further improvement on this work.

Should we use a better resolution?

The highest spatial resolution we utilize in this thesis is $0.04^\circ \times 0.04^\circ$. This is for a reason, and it has to do with the flow data. As mentioned before, the resolution of the flow data is $0.08^\circ \times 0.04^\circ$ (lon×lat). This means that, with our current method of nearest-neighbour sampling, we are doubly sampling in longitudinal direction, and match the latitudinal axis. To avoid even higher oversampling rates, we avoid a higher resolution. Essentially, we have used all the information the flow field could provide at this resolution. We could choose to use an even finer scale and switch to linear (or higher!) interpolation sampling, but the implementation of this interpolation has proven to exceed the scope of this thesis.

Improving the resolution of the time variable t might prove worthwhile. We later discuss whether the influence of the tides as a point of interest, but stated simply: the flow data is provided at intervals of three hours, but we use it with an interval of 24 hours. This means we could get an eightfold improvement, but this also means that 8 times as much data would need to be sent around. Considering the retrieval time of the data is already a considerable factor in the runtime of the models (and sometimes a network failure causes tens of minutes of simulation time to become useless, a problem that can only become worse when the total data volume grows), we have opted to remain at the 24-hour scale.

Should we use a better conversion between degrees and kilometres?

Throughout the thesis, the assumption has been made that $1^\circ = 100$ km everywhere. It should however be mentioned that this approximation is not great: in latitudinal (vertical) direction, degrees are equidistant, but $1^\circ \equiv 111$ km. In longitudinal (horizontal) direction, their distance varies dependent on the latitude. Our region runs from 42.5°N (where $1^\circ \equiv 82$ km) to 46.5°N (where $1^\circ \equiv 76$ km). The choice to not include this varying term is due to the difficulty this variation introduces to the derivations: all the derivations have been done with the assumption the spatial step size δ is the same across the simulation space and the same in x- as y-direction. Correcting the δ to be more accurate across the axes, or accurate across every location in the space would invalidate the current derivations, and the scope of this thesis has not permitted the inclusion of this factor. So we chose a middle ground of $1^\circ \equiv 100$ km.

Is the flow sampled at the correct time?

Another point of interest is the observed difference between the vector fields shown in Figures 5.4 and 5.6. To highlight one difference: at day 100 their field seems to point globally south, while ours shows generally rightward and locally rotational behaviour. Further comparison only reveals more differences in the vector field. Based on the similarity of the solutions our and their models yield, this is likely a problem to do with keeping track of time, or even a difference in phase at which we sample our 24-hour intervals of the flow field.

Should the tides have an influence on our flow simulation?

According to the [22], during each 24 hours and 50 minutes coastal areas see the tide rise twice and fall twice. This means that peak-to-peak, the tide has a period of 12 hours and 25 minutes. There are two factors in the model that might suffer for this pattern: the fact datasets are swapped day-by-day, and the way in which missing data is handled.

The first problem, the day-by-day swapping is not so subtle: A day lasts 24 hours and two tides last 24 hours and 50 minutes, which means the calendar days and tidal period run 50 minutes out of

phase every 24 hours. As a result, the influence of the tides could be seen with a phase of 28.8 days, but running in reverse. This effect can be minimized by reducing the period between swapping out the dataset: if we would use each 3-hour snapshot, we would capture the effect of the tide more accurately. Considering the limiting factor of the runtime of the model is already swapping the flow data out, we have chosen not to do this.

Recall the modelling approach for a missing data point is to utilize the most recent past existent point. This is generally the one 3 hours earlier, as the time variable comes in steps of three. This is where the concern related to the tides comes from: the tide at a 3-hour difference is very different. Taking the snapshot 12 hours ago provides a far more similar tidal reference, but has the setback of being more distant in time for the flow itself. We chose to use the most recent data point, as the tides have not been a consideration throughout the report in general.

6.3. Further work

While we have met our original research goals, many avenues for further development and exploration remain open.

On the mathematical side of these, the first that comes to mind is introducing a spatially varying step size to the derivations in Chapter 2, or simply one that is different along the axes. This would allow us to move beyond the simplifying assumption that $1^\circ = 100$ km, and thus hopefully improve the accuracy of the model. In the same mathematical spirit, we have proven some matters in one dimension in this paper, and assumed them to be true in two dimensions (Section 2.2, Section 4.5). These are not complete proofs, and are thus areas where our work could be strengthened: Finalizing these proofs would be valuable to the validity of our methods.

In a more computational context, there is much that can be done. First, creating functions that download the flow data and save it locally would prove very valuable to the runtime of our models; network and server latency are the primary bottlenecks in runtime. It would also make it possible to run the models without an internet connection. Furthermore, functionality that takes a time as its argument and returns a flow dataset would vastly simplify interaction with the program, as the current approach of one-file-per-year is not friendly to customization. Furthermore, more sophisticated sampling methods for the vector field than nearest-neighbour might open the possibility for higher spatial resolutions. Swapping the vector field every 3 hours instead of 24 hours might introduce stronger tidal effects, and generally improve the accuracy of the streams' influence.

In a broader scientific context, one obvious next step would be tracing real loggerheads as they migrate the region. While such a study [20] has been conducted, it did not focus on our simulation zone. Of course, our study could be applied in this zone too, and then comparisons could be made. Another possible direction to take this thesis is to use earlier stream data: HYCOM provides the ocean flow data starting 01-01-1994, so instead of our 7 data points, there could be 29. This allows for more detailed and more powerful statistical methods, and uncover larger-scale trends.

6.4. Conclusion

The migratory behaviour of loggerhead sea turtle hatchlings within the North Atlantic Gyre is a complex interplay of biological instincts and dynamic ocean currents. We set out to model this phenomenon and explore how it has evolved in recent years. Specifically, we set out to

1. implement an IB model and PDE model to describe the migratory behaviour of loggerhead hatchlings.
2. use the simulations to determine whether the probability of successfully traversing the region of the Gyre has changed in recent years.

Our findings show a good, but not perfect visual consistency between the IB and PDE model. Overlaying the IB model's simulated agent positions with the PDE model's density maps suggests that despite differences in formulation and implementation, these models capture their underlying dynamics in a similar way.

Beyond developing and implementing these models, our analysis has uncovered that the survival probability has changed over recent years ($p = 2.62 \cdot 10^{-6}$), particularly marked by notable deviations in 2019 and 2023. These deviations suggest that external factors may be disrupting the survival rates of loggerhead hatchlings, and highlight the need for further investigation into these yet unknown causes.

Bibliography

[1] Danish A. Ahmed, Joseph D. Bailey, and Michael B. Bonsall. “On random walk models as a baseline for animal movement in three-dimensional space”. In: *Ecological Modelling* 475 (2023), p. 110169. ISSN: 0304-3800. DOI: https://doi.org/10.1016/j.ecolmodel.2022.110169.

[2] Fetsje Bijma, Marianne Jonker, and Aad van der Vaart. *An Introduction to Mathematical Statistics*. 1st ed. Amsterdam University Press, 2017. ISBN: 978-94-6298-510-0.

[3] N. L. Carothers. *Real Analysis*. Cambridge: Cambridge University Press, 2000.

[4] Edward A. Codling, Michael J Plank, and Benhamou Simon. “Random walk models in Biology”. In: (2008). URL: http://doi.org/10.1098/rsif.2008.0014.

[5] Therese A. Conant et al. *Loggerhead Sea Turtle (Caretta caretta) 2009 Status Review Under the U.S. Endangered Species Act*. Tech. rep. Loggerhead Biological Review Team, Aug. 2009.

[6] Copernicus Climate Change Service. *Record-breaking North Atlantic ocean temperatures contribute to extreme marine heatwaves*. Accessed: 2025-06-08. 2023. URL: https://climate.copernicus.eu/record-breaking-north-atlantic-ocean-temperatures-contribute-extreme-marine-heatwaves.

[7] Richard Courant, E. Isaacson, and M. Rees. “On the Solution of Nonlinear Hyperbolic Differential Equations by Finite Differences”. In: *Communications on Pure and Applied Mathematics* 5.3 (1952), pp. 243–255. DOI: 10.1002/cpa.3160050303.

[8] J. Davenport and W. Clough. “Swimming and diving in young loggerhead sea turtles (Caretta caretta L.)” In: *Copeia* (1986).

[9] Albert Einstein. “On the Electrodynamics of Moving Bodies”. In: *Annalen der Physik* 17 (1905). English translation available in *The Principle of Relativity*, Dover Publications, 1952, pp. 891–921.

[10] Ali Ercan. “Self-similarity in fate and transport of contaminants in groundwater”. In: *Science of The Total Environment* 706 (2020), p. 135738. ISSN: 0048-9697. DOI: https://doi.org/10.1016/j.scitotenv.2019.135738.

[11] Ronald A. Fisher. *Statistical Methods for Research Workers*. 1st. Introduced Fisher’s method for combining p-values. Edinburgh: Oliver and Boyd, 1925.

[12] P GNU. *Free Software Foundation. Bash (3.2. 48)[Unix shell program]*. 2007.

[13] Richard Haberman. *Applied Partial Differential Equations with Fourier Series and Boundary Value Problems*. 5th. Boston: Pearson, 2013.

[14] Thomas Hillen and Kevin J. Painter. “A user’s guide to PDE models for chemotaxis”. In: *Journal of Mathematical Biology* 58.1-2 (2009), pp. 183–217. DOI: 10.1007/s00285-008-0201-3.

[15] Thomas Hillen and Kevin J. Painter. “Transport and Anisotropic Diffusion Models for Movement in Oriented Habitats”. In: *Dispersal, Individual Movement and Spatial Ecology: A Mathematical Perspective*. Ed. by Mark A. Lewis, Philip K. Maini, and Sergei V. Petrovskii. Vol. 2071. Lecture Notes in Mathematics. Springer, Berlin, Heidelberg, 2013, pp. 177–222. DOI: 10.1007/978-3-642-35497-7_7.

[16] HYCOM Consortium for Data-Assimilative Ocean Modeling. *GOFS 3.1: 41-layer HYCOM + NCODA Global 1/12° Analysis (NRL)*. https://tds.hycom.org/thredds/catalog.html. Accessed: 2025. 2023.

[17] Mark Kac. “Random Walk and the Theory of Brownian Motion”. In: *The American Mathematical Monthly* 54.7P1 (1947), pp. 369–391. DOI: 10.1080/00029890.1947.11990189.

[18] Kenneth J. Lohmann, Nathan F. Putman, and Catherine M.F. Lohmann. “The magnetic map of hatchling loggerhead sea turtles”. In: *Current Opinion in Neurobiology* 22.2 (2012), pp. 336–342. DOI: 10.1016/j.conb.2011.11.005.

[19] N. Loy, T. Hillen, and K. J. Painter. “Direction-dependent turning leads to anisotropic diffusion and persistence”. In: *European Journal of Applied Mathematics* 33.4 (2022), pp. 729–765. DOI: 10.1017/S0956792521000206.

[20] K. L. Mansfield, V. S. Saba, J. A. Keinath, et al. “Satellite tracking reveals a dichotomy in migration strategies among juvenile loggerhead turtles in the Northwest Atlantic”. In: *Marine Biology* 156.12 (2009), pp. 2555–2570. DOI: 10.1007/s00227-009-1279-x.

[21] NASA Science Editorial Team. *Slowdown of the Motion of the Ocean*. Accessed: 2025-05-31. 2023. URL: https://science.nasa.gov/earth/earth-atmosphere/slowdown-of-the-motion-of-the-ocean/.

[22] National Oceanic and Atmospheric Administration. *Frequency of tides – The lunar day*. Accessed: 2025-05-30. n.d. URL: https://oceanservice.noaa.gov/education/tutorial_tides/tides05_lunarday.html.

[23] Fritz Oberhettinger. *Tables of Fourier Transforms and Fourier Transforms of Distributions*. Berlin: Springer, 1990.

[24] Kevin J. Painter and Thomas Hillen. “From Random Walks to Fully Anisotropic Diffusion Models for Cell and Animal Movement”. In: *Cell Movement: Modeling and Applications*. Cham: Springer International Publishing, 2018, pp. 103–141. DOI: 10.1007/978-3-319-96842-1_5.

[25] Kevin J. Painter and Thomas Hillen. “Navigating the flow: individual and continuum models for homing in flowing environments”. In: *Journal of the Royal Society Interface* 12.112 (2015), p. 20150647. DOI: 10.1098/rsif.2015.0647.

[26] Kevin J. Painter and A. Z. Plochocka. “Efficiency of island homing by sea turtles under multi-modal navigating strategies”. In: *Ecological Modelling* 391 (2018), pp. 40–52. DOI: 10.1016/j.ecolmodel.2018.10.025.

[27] Suhas V. Patankar. *Numerical Heat Transfer and Fluid Flow*. New York: Taylor & Francis, 1980. ISBN: 978-0891165224.

[28] Ukanda. *Loggerhead sea turtle*. https://commons.wikimedia.org/wiki/File:Loggerhead_sea_turtle.jpg. 2006.

[29] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Harlow, Essex, England: Longman Scientific & Technical, 1995. ISBN: 9780582218840.

[30] Kees Vuik et al. *Numerical Methods for Ordinary Differential Equations*. Delft Institute of Applied Mathematics, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands, 2023. DOI: 10.5074/t.2023.001.

[31] L. Wallberg et al. “Extremely warm European summers preceded by sub-decadal North Atlantic ocean heat accumulation”. In: *Earth System Dynamics* 15.1 (2024), pp. 1–14. DOI: 10.5194/esd-15-1-2024.

[32] Xueying Wang, Drew Posny, and Jin Wang. “A reaction-convection-diffusion model for cholera spatial dynamics”. In: *Discrete Contin. Dyn. Syst. Ser. B* 21 (2016), pp. 2785–2809.

[33] David Vernon Widder. *The heat equation*. Vol. 67. Academic Press, 1976.

A

How many steps should a walker make?

One of the issues we ran into while implementing the IB and PDE model, is that there is an important difference between how the addition of a flow is incorporated into the model.

In the PDE model, $u_t = \nabla \cdot (-A(x,y,t)u + D\nabla u)$, when the term $A(x,y,t)$ becomes a combination of the active and passive advection, so when $A(x,y,t) = a + \alpha f(x,y,t)$, we can rewrite the model to $u_t = \nabla \cdot (-au + D\nabla u) + \nabla \cdot (-f(x,y,t)u)$. This means that, while influencing the total behaviour, the short-term effects of the flow field can be seen (and calculated!) separately. This idea is formalized in Appendix D.2 for a one-dimensional equation with constant advection: after doing a coordinate transformation that moves along the direction of advection, the apparent behaviour of the turtles is *independent* of the strength and direction of the vector field[1]. This means that the distance traversed by a turtle in a day can't be constant: the distance it traverses through *active* movement should be constant, but an unknown amount of movement will be introduced by the surrounding flow field.

Now we run into a problem: consider a PJP that moves N steps each day inside a zero-flow vector field, that chooses to *only* go downwards. If we were to run the same simulation inside a constant right-ward vector field $f(x,y,t) = \binom{f_r}{0}$, $f_r > 0$, we would expect the following two things to happen:

1. The total downwards movement of the turtle would remain $N_d = N$ steps, as the vector field moves orthogonal to its movement; their influences should not interfere.
2. The turtles' sideways movement should increase from 0 to some amount N_r caused by the rightwards-flowing field $f(x,y,t)$.

With our current approach, we run into a contradiction: a turtle always makes N steps per day, but now it needs to move $N_d + N_r = N + N_r > N$ steps. So if the total amount of steps the turtle makes remains constant, the effects of the vector field and active movement are not fully represented.

To alleviate this issue, consider the method of determining the probability of moving along each direction, $i \in \{l,r,u,d\}$:

$$p_i = \frac{\widehat{p_i} + \gamma f_i}{1 + \gamma \sum_{j=l,r,u,d} f_j}$$

with the constant probabilities $\widehat{p_i}$ and the components of the flow field f_i. We can rewrite the fraction to be

$$p_i = \underbrace{\frac{\widehat{p_i}}{1 + \gamma \sum_{j=l,r,u,d} f_j}}_{\text{Active movement}} + \underbrace{\frac{\gamma f_i}{1 + \gamma \sum_{j=l,r,u,d} f_j}}_{\text{Passive movement}}.$$

Where we have split up the influence of the active movement and passive movement. We can now reason how many steps a turtle should make: If we assume a turtle can actively swim N $\frac{\text{km}}{\text{day}}$, then a simulation *in a zero-flow field* should have a traversed distance of N $\frac{\text{km}}{\text{day}}$. Stated differently, the final probability of movement p_i in a zero-flow field is fully caused by the turtles own behaviour $\widehat{p_i}$, so the

[1] A similar topic has been broadly studied in the context of physical laws, where these 'coordinate transformations along the direction of advection' are commonly referred to as 'moving frames of reference', see for example [9]

active-probability weighted distance is $N \frac{\text{km·prob.}}{\text{day}}$. If we introduce a vector field to the calculation, the contribution of the probability caused by active movement becomes (summed over all directions)

$$\sum_{i=l,r,u,d} \frac{\widehat{p_i}}{1+\gamma \sum_{j=l,r,u,d} f_j} = \frac{\sum_{i=l,r,u,d} \widehat{p_i}}{1+\gamma \sum_{j=l,r,u,d} f_j} = \frac{1}{1+\gamma \sum_{j=l,r,u,d} f_j},$$

which means that a total of

$$\frac{N}{1+\gamma \sum_{j=l,r,u,d} f_j} \frac{\text{km} \cdot \text{prob.}}{\text{day}}$$

distance was influenced by the walkers' own efforts.

Considering we stated the turtle should actively traverse $N \frac{\text{km}}{\text{day}}$, this means inside the vector field we should make S number of steps on any particular day so that

$$\sum_{k=1}^{S} \frac{N}{1+\sum_{j=l,r,u,d} f_j(x_{t_k}, y_{t_k}, t_k)} = N$$

or equivalently

$$\sum_{k=1}^{S} \frac{1}{1+\sum_{j=l,r,u,d} f_j(x_{t_k}, y_{t_k}, t_k)} = 1$$

to acquire an equal amount of effort from the walkers. Doing this, the effect of the flow field is *added onto* the already present effects, instead of wrongly reducing the influence of the active movement.

When we implement this in an arbitrary vector field, we keep track of the influence the turtle has exerted itself, and make sure that it sums to 1. Each day, we let a turtle make steps where each step add $\left(1+\gamma \sum_{j=l,r,u,d} f_j\right)^{-1}$ to a counter, for the local $f_j's$ (which change with the turtle's movement every day). When this counter equals or exceeds one, the turtle's simulation is stopped for the day, and we simulate the next. *This is how the programs currently approach determining the amount of steps to make.*

There is a disadvantages to this approach, namely that it always overshoots the agency a turtle had. To illustrate the problem, imagine that there is a weak flow field. In fact, let the flow field be only a small $\sum_{i=l,r,u,d} f_i = \epsilon > 0$ strong. Now the influence of the walker is scaled back by some amount $\frac{1}{1+\epsilon} < 1$. This means that the walker gets to make another step. But then the final influence the walker had over its own movement is

$$\frac{2}{1+\epsilon} \approx 2$$

If the influence is slightly above or below 1, that is fine; we would expect there to be some variance. However, turtles traversing double the distance in a weak flow field seems unfavourable, and the expected agency exerted will always be 1 or larger, so the expected value of the influence is > 1. It might seem intuitive to then set this constant value of 1 to some other number, but it is not immediately obvious *which* number then to choose[2].

We propose an alternative, but implementing this alternative into the program has not fit the scope of this research. If an agent currently has exerted its own influence equal to $n < 1$, and the next step will add an influence of m such that $n+m > 1$, then it would have exerted more agency than we wanted it to. To solve this, consider that there exists a number, $q \in [0, m]$ such that $n + q = 1$. If we draw a number x from a Uniform$[0, m]$-distribution, and only let the Walker act if $x \leq q$, then the probabilistic balance is restored: the last step adds an expected value of $\frac{q}{m} \cdot m = q$ agency to the count, of which we knew $n + q = 1$. Unfortunately, the scope of the paper has not permitted this idea being implemented.

[2] Heuristically, 0.5 might seem like the better choice. However, reasoning back from the characteristics of the vector field, the average overshoot should be half of the average undershoot to acquire a balanced influence. Choosing 0.5 assumes the overshoot is always 1, while it is always known to be < 1, so would yield an expected under-agency. This problem is likely more nuanced.

B

Formal derivation for difference operators

To solve the different parts in $\nabla \cdot (Au + D\nabla u)$, we need to use two different numerical methods, and here we describe how they can be found and implemented.

The diffusive part is solved using a central difference scheme. To understand how we transform the diffusive part, Equation 2.7, to matrix form (as u remains a matrix for the computer!). Recall

$$\begin{aligned} u_t &= \nabla \cdot D\nabla u \\ &= \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} \cdot \begin{pmatrix} cH & 0 \\ 0 & cV \end{pmatrix} \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} u \\ &= \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} \cdot \begin{pmatrix} cH\partial_x u \\ cV\partial_y u \end{pmatrix} \\ &= cH\partial_{xx}u + cV\partial_{yy}u. \end{aligned}$$

We can now reason about what happens if we discretize the space. Let $u_{i,j} \in \mathbb{R}$ denote the $(i,j)^{\text{th}}$ element in the matrix that represents the evaluation of $u(x,y,t)$ at spatial location (i,j), so that $u(t) \in \mathbb{R}^{n_x \times n_y}$. Thus, $u(t)=\left(u_{i,j}(t)\right)$. Considering our choice of a central difference scheme (and Forward Euler in time!), we want to move towards a form of

$$\begin{aligned} u_{i,j}(t+dt) &= u_{i,j}(t) + dt \cdot (\nabla \cdot D\nabla u_{i,j}(t)) \\ u_{i,j}(t+dt) - u_{i,j}(t) &= dt \cdot \left(\begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} \cdot \begin{pmatrix} cH & 0 \\ 0 & cV \end{pmatrix} \begin{pmatrix} \partial_x u_{i,j}(t) \\ \partial_y u_{i,j}(t) \end{pmatrix} \right) \\ \frac{u_{i,j}(t+dt) - u_{i,j}(t)}{dt} &= cH\partial_{xx}u_{i,j}(t) + cV\partial_{yy}u_{i,j}(t) \end{aligned}$$

for each element in $u(t) = u_{ij}(t)$. Substituting in the central difference scheme,

$$\partial_{xx}u_{i,j}(t) = \frac{u_{i-1,j}(t) - 2u_{i,j}(t) + u_{i+1,j}(t)}{dx^2}$$

we acquire (note the ∂_{yy} acts on the j-index!)

$$\frac{u_{i,j}(t+dt) - u_{i,j}(t)}{dt} = \left(cH\frac{u_{i-1,j}(t) - 2u_{i,j}(t) + u_{i+1,j}(t)}{dx^2} + cV\frac{u_{i,j-1}(t) - 2u_{i,j}(t) + u_{i,j+1}(t)}{dy^2} \right)$$

Ignoring boundary conditions for now, we can find the ∂_{xx} operator as the left-applied $n_x \times n_x$-matrix with -2 on the diagonal, and 1 on the off-diagonals. ∂_{yy} is the similarly defined $n_y \times n_y$-matrix, but

should be right-applied.[1]

$$\partial_{xx} \widehat{=} \frac{1}{dx^2}\begin{pmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & & \vdots \\ 0 & 1 & -2 & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \dots & 0 & 1 & -2 \end{pmatrix}$$

This matrix representation of the operator preserves the size of the axes of u, and is such that we can implement the diffusion part with a matrix-only approach. This is computationally efficient, and permits a simpler implementation by using the NumPy package.

In a similar fashion, we can reason what the advection operator should look like. For the advection, we use a different numerical scheme, as we are looking for a first-order difference. We should however note that for the flow field, the advective term is dependent on space. In 2.2 we have shown this to be valid in 1D, and we heuristically assume this validity holds for 2D as well. For the constant advective part, this derivation is the same. Similar to the diffusion we start at

$$\begin{aligned} u_{i,j}(t+dt) &= u_{i,j}(t) + dt \cdot (\nabla \cdot -A_{i,j} u_{i,j}(t)) \\ &= u_{i,j}(t) - dt \cdot \Big(\begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} \cdot \begin{pmatrix} a_{Hi,j} \\ a_{Vi,j} \end{pmatrix} u_{i,j}(t) \Big) \\ \frac{u_{i,j}(t+dt) - u_{i,j}(t)}{dt} &= -\partial_x (a_H u)_{i,j}(t) - \partial_y (a_V u)_{i,j}(t) \end{aligned}$$

In this last step, we have contracted the subscript-i,j to be over the element-wise multiplication of a_H, a_V and u. This is also precisely what happens computationally: a_H and a_V represent the ocean data-produced flow field. They are matrices of equal size to u, so using $\odot$ to denote the Hadamard product, $(a_H \odot u)_{i,j}$ is the i,j^{th} entry of the element-wise multiplication between a_H and u. This is useful, because we want to use a first-order upwind scheme (introduced in [7] and further discussed in [27]). This means we do first-order forward/backward difference, *in the direction of local advection*.

Again neglecting boundary conditions, the forward difference resp. backward difference in x-direction are

$$\partial_x^+ (a_H \odot u)_{i,j} = \frac{(a_H \odot u)_{i+1,j} - (a_H \odot u)_{i,j}}{dx}, \quad \partial_x^- (a_H \odot u)_{i,j} = \frac{(a_H \odot u)_{i,j} - (a_H \odot u)_{i-1,j}}{dx}$$

then we acquire

$$\partial_x^+ = \frac{1}{dx}\begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & 0 \\ 0 & 0 & -1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}, \quad \partial_x^- = \frac{1}{dx}\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \ddots & 0 \\ 0 & -1 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

and similarly for $\partial_y^+, \partial_y^-$. The application side of this operator is similar to the diffusive operator. The implementation in code is handled (for example in horizontal direction) by determining for which indices $(a_H \odot u)_{i,j}$ is positive/negative. Considering u is a density function, when $(a_H \odot u)_{i,j}$ is positive, the direction of the vector field is positive. We then know to use the forward difference or vice-versa. The matrix $a_H \odot u$ is created using **numpy.multiply**, and the matrix multiplications are handled with **numpy.matmul**. The flux terms then are implemented by

$$\textbf{flux} = -\partial_x^+ \max\left((a_H \odot u), 0\right) - \partial_x^- \min\left((a_H \odot u), 0\right)$$

[1] This left- or right-side application *really* matters! The matrix defining the operator might be symmetric, but left-applying yields rows in u (try writing out the matrix multiplication), keeping j constant for each element. This is exactly what ∂_{xx} requires, and oppositely ∂_{yy} requires columns, meaning it needs to be right-applied.

C

Statistics on success probabilities

To determine whether the Null hypothesis

$$H_0 : p_{\text{success}} := q \text{ is a constant value}$$

can justifiably be rejected, we have to do some statistics [2].

If we model the amount of turtles that make it to the correct side of the region as a binomial variable, then the amount of surviving turtles T is a binomially distributed variable by,

$$T \sim \text{Bin}(1825, q)$$

where we know the sample size $N = 1825 = 365 \cdot 5$, as for our sample we release 5 turtles per day, for 365 days.

Doing this, the question of whether q is constant can be rephrased: What is the probability that a number of samples as extreme as $T_1 \cdots T_m$ or worse are all drawn from the same Binomial process?

We can not simply compare the p-values of each sample we created to a significance level. To determine the significance of drawing the complete sample (so all the probabilities we find), we have to utilize Fishers Method, also discussed in [11], defining a test statistic that follows a χ^2_{2m}-distribution under the Null hypothesis (recall m is the number of samples):

$$x = -2 \sum_{i=1}^{k} \ln p_i.$$

We can now acquire our total p-value by testing the probability of this x or more extreme values arising from a χ^2_{2m} distribution, i.e.:

$$p\text{-value} = 1 - \mathbb{P}(x \geq \chi^2_{2m})$$

This final evaluation is done using SciPy's **chi2.cdf**.

We turn our attention to answering the original research question: can we reject H_0? To answer this, for each probability $q \in [0, 1]$ we plot the associated p-value (*ignoring the data from 2019 and 2023*), to see if any q exist so that we can accept H_0.

Turning our attention to Figure C.1, we can see the result of this experiment. The q with the highest related p-value to our Null hypothesis is $q = 0.583$. Plotting a dashed red line in the probabilities we found per year, we can see 0.583 does indeed seem to be centred in the data. However, it should be noted that the vertical axis for the p-values is at a very small scale, 10^{-6}.

This means that, after already having omitted the data from 2019 and 2023, we find a p-value of $2.62 \cdot 10^{-6}$ at best. So we can conclude that each year, the amount of turtles who make it across the region is almost certainly not a binomial process with a constant probability: if the choice to model it as a binomial is appropriate, then its probability p_{success} must almost surely depend on the year.

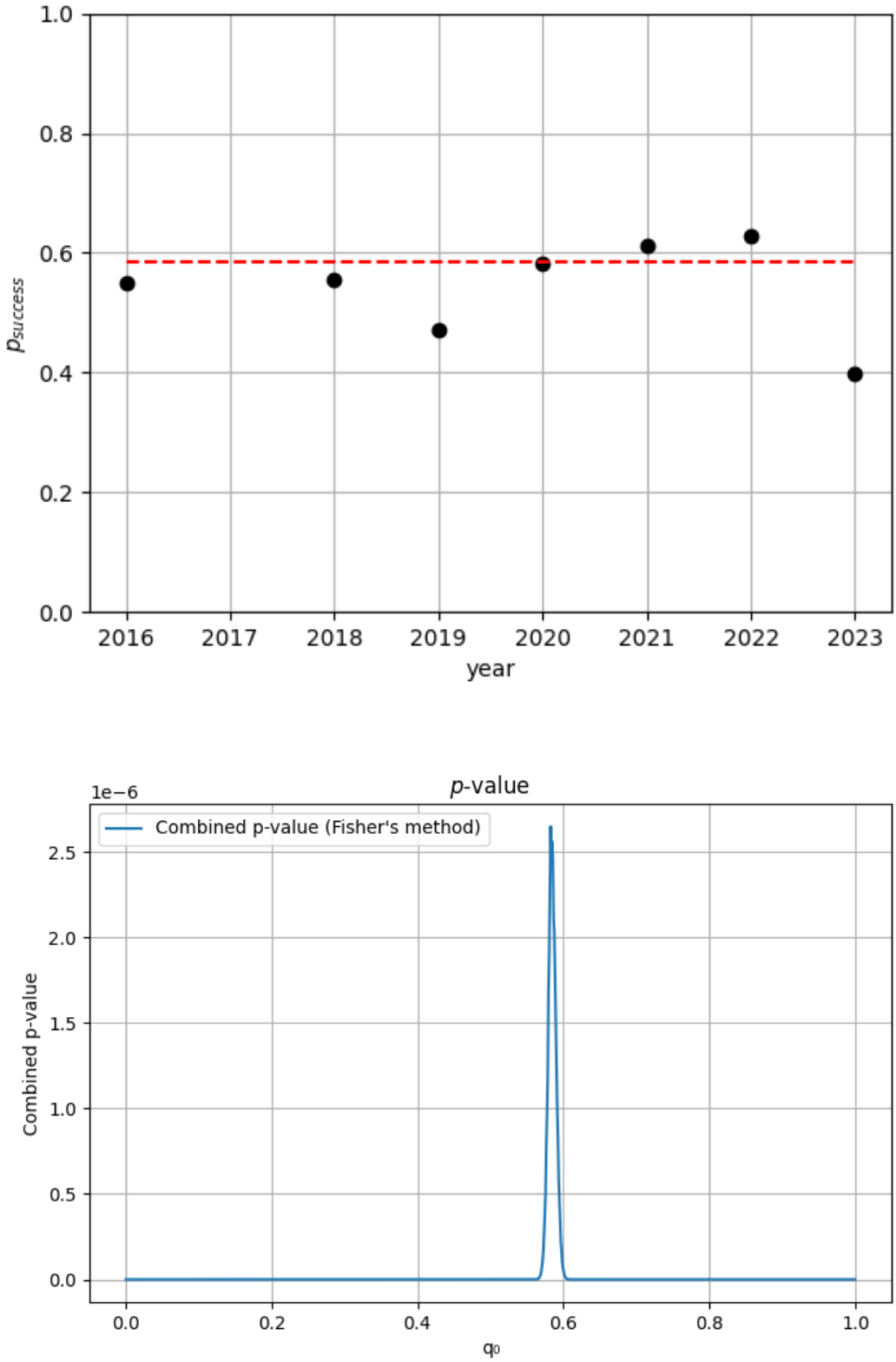

Figure C.1: Top: The probabilities found by running the IB model, per year. A dashed red line is included at the maximum likelihood estimator $q = 0.58518$. Bottom: The p-value related to each q, assuming the data points in 2019 and 2023 are exceptions and should therefore not be included. The maximum p-value lies at $q = 0.5824$. Note the vertical axis scale factor of 10^{-6}

D

Solutions to advection-diffusion equations

In Chapter 2 we have derived continuous PDE models from RWs, that describe densities in population starting from assumptions about the behaviour of the individual agents. In this chapter we show that there exist exact solutions to some simple cases of these PDE's.

D.1. Diffusion equation on an infinite domain

We wish to solve the fully symmetric advection-diffusion equation. This is a PDE, so we need to also determine a domain on which to solve, and an initial condition to solve with. As we are going to model individual turtles, we choose to represent the initial condition as Dirac Delta distribution $\delta(x)$. The resulting solution then describes the population density of the turtles over time.

As for the boundary conditions, to lead into a 'simpler' version of the model, we start with an infinite domain. After this we show the case with Neumann conditions.

Considering the above, we wish to solve

$$\begin{cases} u_t(x,t) = cu_{xx}(x,t) \quad x \in \mathbb{R}, t > 0, c > 0 \\ u(x,0) = \delta(x). \end{cases} \tag{D.1}$$

The usual ansatz for a problem on an unrestricted domain is to use a Fourier transformation. We refer to [13] for additional details on approaching this problem. Let

$$\mathcal{F}[u(x,t)](\omega,t) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-i\omega x}u(x,t)dx := U(\omega,t),$$

$$\mathcal{F}^{-1}[u(\omega,t)](x,t) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{i\omega x}U(\omega,t)d\omega = u(x,t)$$

denote the Fourier transform in x, so $p(x,t) \overset{\mathcal{F}}{\leftrightarrow} P(\omega,t)$ is a Fourier pair. Now applying the Fourier transform to each side of equation D.1,

$$\mathcal{F}[\frac{\partial}{\partial t}u(x,t)](\omega,t) = \frac{\partial}{\partial t}U(\omega,t) = U_t,$$
$$\mathcal{F}[c\frac{\partial^2}{\partial x^2}u(x,t)](\omega,t) = c(i\omega)^2\mathcal{F}[u(x,t)](\omega,t) = -c\omega^2 U$$

yields

$$U_t(\omega,t) = -c\omega^2 U(\omega,t),$$

which has the usual solution of

$$U(\omega,t) = C_1 e^{-c\omega^2 t}, \quad C_1 \in \mathbb{R}. \tag{D.2}$$

To determine C_1 we will need the Fourier Transform of the initial condition, as we essentially want to use $U(\omega, 0)$:

$$U(\omega, 0) = \mathcal{F}[\delta(x)](\omega, 0) = \frac{1}{\sqrt{2\pi}}.$$

Plugging this initial condition into equation D.2, we get

$$C_1 = \frac{1}{\sqrt{2\pi}}.$$

So that it remains to determine $u(x,t)$ by means of Fourier inversion,

$$\begin{aligned} u(x,t) &= \mathcal{F}^{-1}[U(\omega,t)] \\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{ix\omega}U(\omega,t)d\omega \\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{ix\omega}\frac{1}{\sqrt{2\pi}}e^{-c\omega^2 t}d\omega \\ &= \frac{1}{\sqrt{2\pi}}\cdot\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{ix\omega}e^{-c\omega^2 t}d\omega \\ &= \frac{1}{\sqrt{2\pi}}\cdot\frac{1}{\sqrt{2ct}}e^{\frac{-x^2}{4ct}} \\ &= \frac{1}{\sqrt{4\pi ct}}e^{\frac{-x^2}{4ct}}. \end{aligned}$$

Here we use that $\mathcal{F}\left[e^{\frac{-x^2}{4ct}}\right] = \frac{1}{\sqrt{2ct}}e^{-c\omega^2 t}$, with our definition of the Fourier transform. See for example [23] for a collection of Fourier Transforms.

D.2. Advection-diffusion equation, on an infinite domain

To present a slightly more complicated model, we can reintroduce the advective term. So, consider the model

$$\begin{cases} u_t = au_x + cu_{xx} \quad x \in \mathbb{R}, a \in \mathbb{R}, t > 0, c > 0, \\ u(x,0) = \delta(x). \end{cases} \tag{D.3}$$

The term describing an added velocity $a \in \mathbb{R}$ is constant, meaning that in a frame of reference moving with this constant speed of advection a, no movement should be observed. Therefore, permit the transformation $x' = x + at$ and $t' = t$. We can then rewrite the derivatives using the chain rule

$$\begin{cases} \frac{\partial}{\partial x} = \frac{\partial}{\partial x'}\frac{\partial x'}{\partial x} + \frac{\partial}{\partial t'}\frac{\partial t'}{\partial x} = \frac{\partial}{\partial x'} \\ \frac{\partial}{\partial t} = \frac{\partial}{\partial t'}\frac{\partial t'}{\partial t} + \frac{\partial}{\partial x'}\frac{\partial x'}{\partial t} = \frac{\partial}{\partial t'} + a\frac{\partial}{\partial x'}. \end{cases}$$

Using this, we can rewrite the parts in Equation D.3 the above as follows:

$$\begin{aligned} u_t &= \frac{\partial}{\partial t'}u + a\frac{\partial}{\partial x'}u = u_{t'} + au_{x'} \\ u_x &= \frac{\partial}{\partial x'}u = u_{x'} \\ u_{xx} &= \frac{\partial^2}{\partial x'^2}u = u_{x'x'}. \end{aligned}$$

Filling in these new terms in Equation D.3, we find

$$\begin{aligned} u_t &= au_x + cu_{xx} \\ u_{t'} + au_{x'} &= au_{x'} + cu_{x'x'} \\ u_{t'} &= cu_{x'x'}. \end{aligned}$$

Now we see the advective term vanish. This means that using the substitution $x' = x + at$ we have reduced the model to a case which can be solved with the same approach as the previous section, D.1. Therefore, we acquire the solution

$$p(x,t) = \frac{1}{\sqrt{4\pi ct}} e^{\frac{-(x-at)^2}{4ct}}$$

and are thus finished.

D.3. Diffusion equation on a finite domain

There is a stark difference between approaching the solution of this equation when evaluated on a finite or infinite domain. We just saw an infinite domain permits a Fourier transformation. For a finite domain, a more suitable ansatz would be an eigenfunction expansion. As we want to model Neumann boundary conditions, consider

$$\begin{cases} u_t(x,t) = cu_{xx}(x,t) & x \in [0,L], t \geq 0 \\ u(x,0) = \delta(x - \frac{L}{2}) & \\ u_x(0,t) = 0 & t > 0 \\ u_x(L,t) = 0 & t > 0. \end{cases}$$

If we assume that u consist of a multiplication between a spatially dependent and temporally dependent part, so $u(x,t) = X(x)T(t)$, we acquire

$$\begin{aligned} X(x)T'(t) &= cX''(x)T(t) \\ \frac{1}{cT(t)}T'(t) &= \frac{1}{X(x)}X''(x). \end{aligned}$$

We now have an equation where a temporally dependent and spatially dependent part have to equal to each other. This means that these terms must equal some shared constant $\lambda \in \mathbb{R}$,

$$\begin{cases} \frac{1}{cT(t)}T'(t) = \lambda \\ \frac{1}{X(x)}X''(x) = \lambda. \end{cases} \tag{D.4}$$

We have now acquired two ordinary differential equations. To solve these, we need to consider values for λ, first determining possibilities for its sign.

First assume $\lambda = \alpha^2 > 0$. Solving the problem in X, we get

$$X''(x) = \alpha^2 X(x)$$

which has a general solution in terms of $\cosh$ and $\sinh$

$$X(x) = C_1 \cosh(\alpha x) + C_2 \sinh(\alpha x).$$

Applying the boundary conditions in X, so

$$\begin{cases} u_x(0,t) = X'(0)T(t) = 0 \\ u_x(L,t) = X'(L)T(t) = 0, \end{cases}$$

with the requirement that $T \neq 0$ for non-trivial solutions[1], yields that $X'(0) = X'(L) = 0$:

$$\begin{cases} C_1\alpha^2 \cosh(0) + C_2\alpha^2 \sinh(0) = 0 \\ C_1\alpha^2 \cosh(\alpha L) + C_2\alpha^2 \sinh(\alpha L) = 0. \end{cases}$$

Now realizing in the first $\sinh(0) = 0$ implies $C_1 = 0$. But then $C_2\alpha^2 \sinh(\alpha L) = 0$ must have $C_2 = 0$, as $\sinh(\alpha L) = 0$ only when $\lambda = \alpha^2 = 0$ or $L = 0$. We have found $C_1 = C_2 = 0$, so $\lambda > 0$ yields only trivial solutions.

[1] Note that when $T \equiv 0$, $u(x,t) = T(t)X(x) \equiv 0$. These are obviously not interesting solutions, so we avoid them.

Moving on to $\lambda = 0$ we see, again solving from X, that $X''(x) = 0$ implies $X(x) = C_3 x + C_4$. Using the boundary conditions quickly shows $C_3 = C_4 = 0$, which means $\lambda = 0$ only yields trivial solutions.

The last possibility is $\lambda < 0$. For this purpose write $\lambda = -\beta^2 < 0$. Writing out the eigenvalue problem in X again,

$$X''(x) = -\beta^2 X(x),$$

which has solutions

$$X(x) = C_5 \sin(\beta x) + C_6 \cos(\beta x).$$

Now again filling in boundary conditions,

$$\begin{cases} X'(0) = \beta C_5 \cos(0) - \beta C_6 \sin(0) = 0 \\ X'(L) = \beta C_5 \cos(\beta L) - \beta C_6 \sin(\beta L) = 0. \end{cases}$$

In the first of these, we know $\cos(0) = 1$ and $\sin(0) = 0$, so $\beta C_5 = 0$ implies $C_5 = 0$. Now in the second equation, $-\beta C_6 \sin(\beta L) = 0$. Considering $\beta > 0$ (and $C_6 = 0$ would imply only trivial solutions can exist), we divide them out to acquire $\sin(\beta L) = 0$. This means

$$\begin{aligned} \sin(\beta L) &= 0 \\ \beta_n L &= n\pi, \quad n \in \mathbb{Z} \\ \beta_n &= \frac{n\pi}{L}, \quad n \in \mathbb{Z}, \end{aligned}$$

so there are multiple non-trivial solutions to the eigenvalue problem in X, which look like $X_n(x) = C_{n,1} \cos\left(\frac{n\pi}{L} x\right)$ depending on n. Due to the principle of superposition, we ignore this constant for now as it will be combined with the one we find solving for T. Solving the eigenvalue problem for $T_n(t)$ with $\lambda_n = -\beta_n^2 = -\left(\frac{n\pi}{L}\right)^2$, we get

$$\begin{aligned} T_n'(t) &= -\left(\frac{n\pi}{L}\right)^2 c T_n(t) \\ T_n(t) &= C_{n,2} e^{-\frac{n^2\pi^2 c}{L^2} t}. \end{aligned}$$

To re-acquire our solution $p(x,t)$, we use the principle of superposition [13]. Reintroducing the constants of the eigenfunctions as A_n, dependent on n,

$$\begin{aligned} u(x,t) &= \sum_{n=1}^{\infty} A_n X_n(x) T_n(t) \\ &= \sum_{n=1}^{\infty} A_n \cos\left(\frac{n\pi}{L} x\right) e^{-\frac{n^2\pi^2 c}{L^2} t}. \end{aligned}$$

Solving for the initial condition $u(x,0) = \delta(x - \frac{L}{2})$, as to find A_n, consider

$$u(x,0) = \sum_{n=1}^{\infty} A_n \cos\left(\frac{n\pi}{L} x\right) = \delta\left(x - \frac{L}{2}\right).$$

We take the inner product on the left and right side with $\cos\left(\frac{m\pi}{L} x\right)$ for some $m \in \mathbb{Z}$, with the standard inner product $< f, g >= \int_0^L f(x) g(x) dx$.

$$\int_0^L \sum_{n=1}^{\infty} A_n \cos\left(\frac{n\pi}{L} x\right) \cos\left(\frac{m\pi}{L} x\right) dx = \int_0^L \cos\left(\frac{m\pi}{L} x\right) \delta\left(x - \frac{L}{2}\right) dx.$$

Due to Fubini's theorem [3], we are permitted to exchange summation and integration here. This yields

$$\sum_{n=1}^{\infty} \int_0^L A_n \cos\left(\frac{n\pi}{L}x\right) \cos\left(\frac{m\pi}{L}x\right) dx = \int_0^L \cos\left(\frac{m\pi}{L}x\right) \delta\left(x - \frac{L}{2}\right) dx.$$

Now we realize that when $n \neq m$, $\int_0^L A_n \cos\left(\frac{n\pi}{L}x\right) \cos\left(\frac{m\pi}{L}x\right) dx = 0$. Therefore, those terms vanish from the sum and we are left with

$$\int_0^L A_m \cos\left(\frac{m\pi}{L}x\right)^2 dx = \int_0^L \cos\left(\frac{m\pi}{L}x\right) \delta\left(x - \frac{L}{2}\right) dx$$

$$\frac{L}{2} A_m = \cos\left(\frac{m\pi}{L}\frac{L}{2}\right)$$

$$A_m = \frac{2}{L} \cos\left(\frac{m\pi}{2}\right).$$

We now have collected all the components to write down the final solution to our problem.

$$u(x,t) = \frac{2}{L} \sum_{n=1}^{\infty} \cos\left(\frac{n\pi}{2}\right) \cos\left(\frac{n\pi}{2}x\right) e^{-\frac{n^2\pi^2 c}{L^2}t}.$$