

# Quantifying the Golgi

Gert-Jan Both

Supervised by:  
P. Sens  
C. Storm

Technical university of Eindhoven  
January-November 2018

# Abstract

The Golgi apparatus is a key component in intracellular trafficking, maturing and directing proteins essential to the cell. Despite years of research, a model coupling Golgi size and function to the cells' transport properties is lacking. In this thesis we develop such a model, describing the Golgi as an active droplet. New experimental data sheds more insight in the spatial organization of the trafficking and I have also devised two new methods relying on image gradients and neural networks to analyze this data and confront it with our model.

# Contents

## ABSTRACT

i

## 1 INTRODUCTION

1.1	The secretory pathway: biology 101 for physicists . . . . .
1.2	This thesis . . . . .

## 2 MODEL FITTING

2.1	Step 1 - Smoothing and denoising . . . . .
2.2	Step 2 - Derivatives . . . . .
2.3	Step 3 - Segmentation . . . . .
2.4	Step 4 - Fitting . . . . .

## 3 DATA ANALYSIS

3.1	Initial analysis . . . . .
3.2	Analysis of least-squares fit . . . . .
3.3	Conclusion . . . . .

## 4 PHYSICS INFORMED NEURAL NETWORKS

4.1	Neural Networks . . . . .
4.2	Physics Informed Neural Networks . . . . .

## 5 GOLGI AS A PHASE SEPARATED DROPLET

5.1	Phase separation . . . . .
5.2	Effective droplet . . . . .
5.3	Golgi as an active droplet . . . . .

## 6 RESULTS MODEL

6.1	Effective droplet . . . . .
6.2	Free droplet . . . . .
6.3	Droplet stuck to walls . . . . .
6.4	Biological implications . . . . .

## 7 CONCLUSION

## REFERENCES

# 1

## Introduction

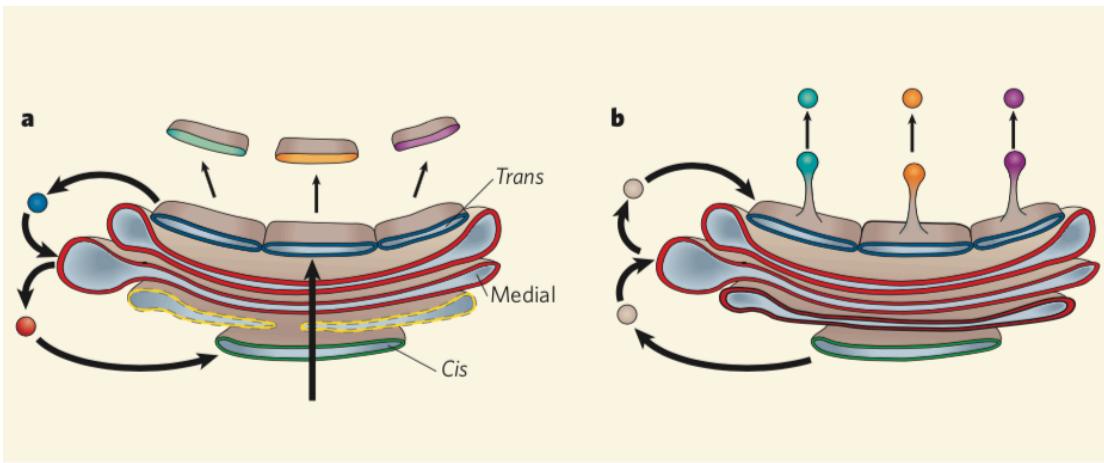
The cell uses thousands of proteins and lipids to function. Many of these are produced in the Endoplasmic Reticulum (ER), an organelle found in eukaryotic cells. Upon exiting the ER, these proteins and lipids are then transported throughout the cell in a process known as intracellular transport. Key component in this intracellular trafficking is the Golgi apparatus, an organelle responsible for biochemically maturing proteins and directing them to the right location. Intense research over the last years has identified key players,<sup>1,2</sup> but an integrated model coupling Golgi size and function to the intracellular transport is lacking<sup>3,4</sup>. In this thesis, we seek to propose such a model.

### 1.1 THE SECRETORY PATHWAY: BIOLOGY 101 FOR PHYSICISTS

Proteins produced in the ER exit the organelle at specific locations referred to as ER Exit Sites - ERES. At these sites, cargo is packaged into a lipid bilayer and this package, known as a vesicle, buds off into the cytoplasm<sup>5</sup>. ERES are located throughout the cell and thus the vesicles need to be transported to their destination: the Golgi apparatus. In general, we can recognise two different trafficking modes: diffusive and directive<sup>6</sup>. In the directive mode, molecular motors pull vesicles along microtubules by hydrolysing ATP. Microtubules (MTs) are long tubular polymers spread throughout the cell and form a network which acts as the backbone for intracellular transport. They are organised around objects known as MicroTubular Organisation Centers (MTOCs). The primary MTOC is the centrosome, an organelle located next to the nucleus, but strong evidence exists that the Golgi apparatus acts a MTOC too<sup>7,8</sup>.

Microtubules are polarised and have two distinct ends, indicated as the (+) and (-). Different molecular motors are utilised for transport towards each end, with dynein being (-)-directed and kinesin (+)-directed<sup>9</sup>. Vesicles are often attached to multiple motors of both types, binding and unbinding constantly, making this active transport a stochastic process which can, for example, be described by a tug-of-war model<sup>10</sup>. Furthermore, cargo can also completely detach from all molecular motors. The vesicle will then move through the cytoplasm in a diffusive way, until it reattaches to a microtubule. Note that diffusive mode is a deceptively simple name, as the cytoplasm is not a simple fluid; it is packed with other cellular components, giving rise to effects such as anomalous diffusion or crowding.

The intracellular trafficking transports the vesicles towards the Golgi, where the cargo undergoes biochemical modification (a process generally referred to as maturation) and is sorted before being sent to their destination. The Golgi thus acts as a sort of post-office of the cell, receiving cargo, repackaging it and sending it to the right destination<sup>11</sup>. Although the function of the Golgi is similar for different cell types, its appearance is strongly dependent on it. In plants for example, the Golgi is distributed throughout the cell in separate but fully functional subunits known as stacks<sup>12</sup>, whereas in mammals all these stacks are localised close to the nucleus in a single organelle known as the *Golgi Ribbon*<sup>7</sup>. A stack consists of a number of stacked compartments of a disk-like shape called *cisternae*. These are membrane enclosed objects containing the enzymes responsible for biochemically altering the proteins, a process generally referred to as maturing. Proteins move through the Golgi in a particular direction and the Golgi thus has distinct entry and exit faces. These are known respectively as the cis and trans face, with the cisternae being labeled analogously. The cisternae in the middle of the stack are referred to as medial compartments.



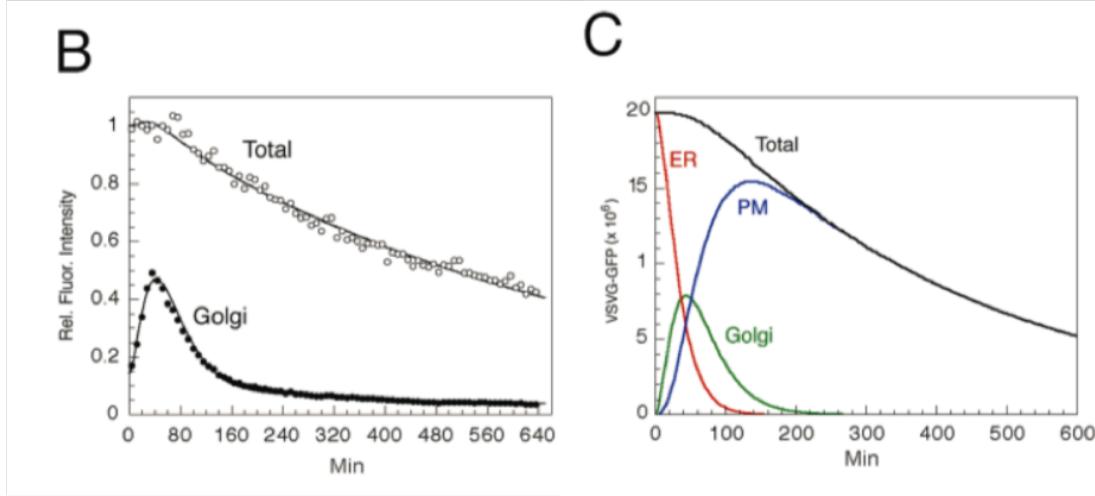
**Figure 1.1:** Left panel: In the cisternal maturation model, compartments mature as a whole and thus change identity. Right panel: In the vesicle transport model, compartments are static objects and cargo is being transported from compartment to compartment by vesicles. Image taken from 13.

At the cis-face vesicles fuse with the Golgi and release their cargo into a compartment, while their lipid bilayer becomes part of the compartment membrane. Exactly how maturation then happens is debated<sup>13,14</sup>. The two main competing explanations are the *cisternal maturation* and *vesicular transport* models. In figure 1.1 we show the structure of the Golgi and a schematic view of each model. In the cisternal maturation model (left panel of 1.1), the compartments mature as a whole, changing identity from cis to medial and finally to trans. Trans compartments are recycled into cis compartments by retrograde vesicular transport. In the vesicular transport model (see right panel of 1.1), the vesicles move in the opposite direction. Rather than constantly changing identity, in this model cisternae are static entities with a defined task and cargo is moved from one compartment to the next by vesicles. The debate could thus be settled by analysing the direction of the vesicles, but so far this has proven elusive. At the trans-face, the cargo is encapsulated again in a lipid bilayer and is transported to its destination, similar to pre-Golgi intracellular transport.

### 1.1.1 QUANTITATIVE MODELS OF THE GOLGI

The Golgi has been intensively studied by biologists for many years, but very few attempts at quantifying the link between the Golgi and the intracellular transport appear to have been made: our research only turned up a single attempt by Hirschberg et al<sup>15</sup>, where the authors present a model for the trafficking of VSVG virus from the ER to the plasma membrane. The secretory pathway is modeled by dividing it into populations connected by a

first order rate equation, i.e.  $d\varphi_2/dt = k_{i \rightarrow 2}\varphi_i$ . Assuming no flowback (i.e.  $k_{i+1 \rightarrow i} = 0$ ) and a population for the ER, Golgi and Plasma Membrane, they find that such a model is sufficient to describe their experimental data, as shown in figure #fig:ratemodel.



**Figure 1.2:** Left panel: First order rate model fitted to experimental data by 15 Right panel: Inferred concentration in ER, Golgi and PM using the fitted parameters from the left panel and their model. Image reprinted from 15.

We reprint their main result in figure 1.2. Although this model describes the experimental data, it is a phenomenological model and reduces the entire system to a few rate parameters  $k_{i \rightarrow i+1}$ . These rate parameters are not coupled to any of the underlying processes and hence this model does not offer any insight into the system. Furthermore, the model lacks any spatial dependence of the concentration.

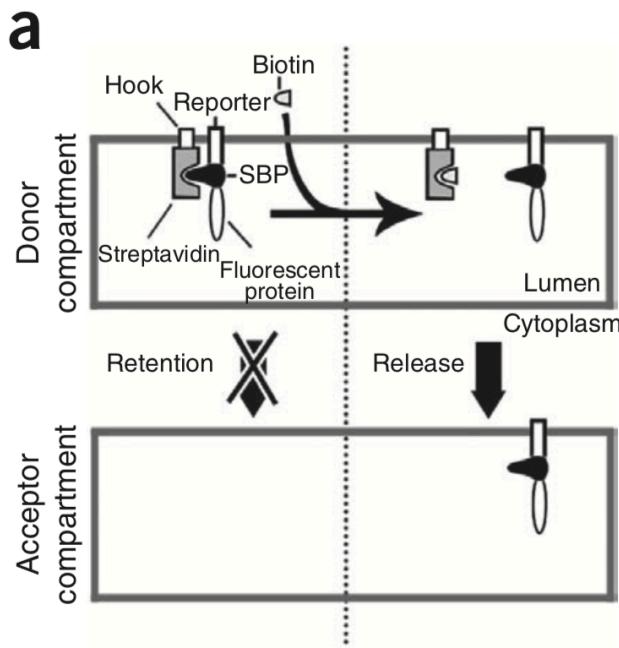
## 1.2 THIS THESIS

Interestingly, studies have shown that the Golgi is able to form *de novo*<sup>8</sup>, meaning that in cells from which the Golgi has been removed, it will automatically reappear. In this thesis we propose a model which couples the intracellular transport to the Golgi in order to explain this by describing the Golgi as an *active, phase separated droplet* and the intracellular transport by an advection-diffusion process. Our model is not restricted to the formation-phase of the Golgi: as we have linked the intracellular transport to the Golgi, it also predicts a spatial distribution of the cargoes being trafficked by intracellular transport when the Golgi is functioning properly. This allows us to confront our model with experimental data gathered by the team of Frank Perez at Institut Curie. This team has developed a new

technique called RUSH<sup>16</sup>, which is used to study the intracellular transport from the ER to the Golgi and beyond using fluorescence microscopy. In the next sections we introduce the experimental data, how we intend to analyse it and justify the description of the Golgi as a phase separated droplet.

#### 1.2.1 EXPERIMENTAL DATA

RUSH (Retention Using Selective Hooks) has recently been developed<sup>16</sup> in the team of Frank Perez at Institut Curie to study intracellular trafficking from the ER to the Golgi and even post-Golgi using fluorescent live-cell imaging.

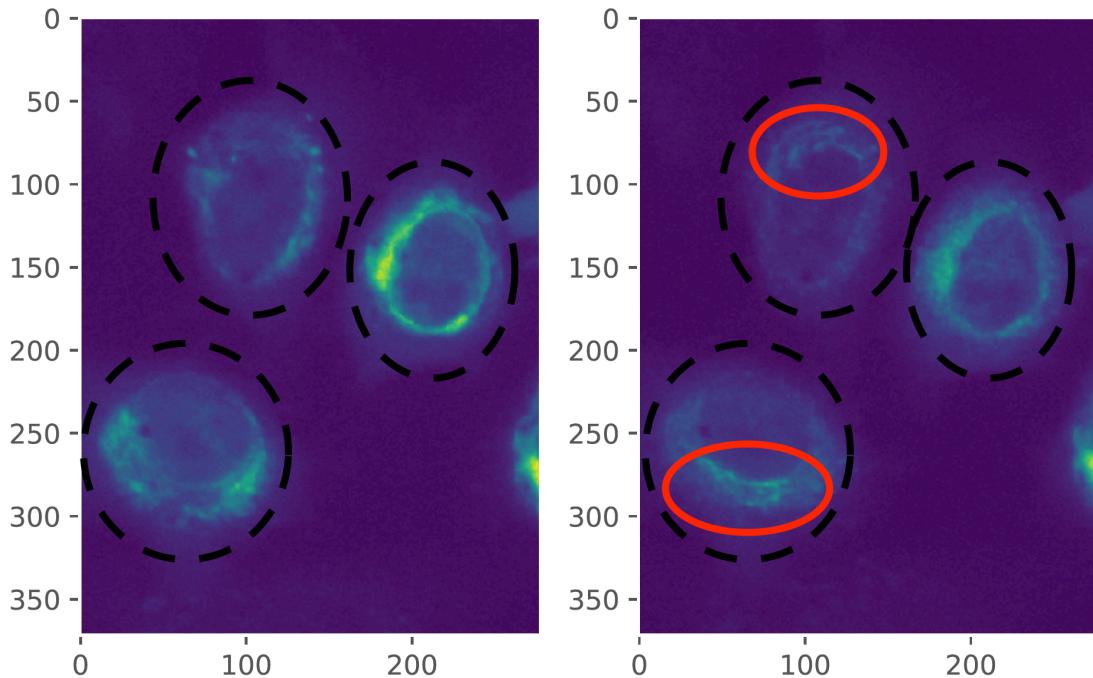


**Figure 1.3:** Schematic overview of the RUSH system. Image taken from 16

Figure 1.3 shows the principle of the RUSH system. Inside the ER, a core streptavidin is fused to the ER using a hook protein. Another protein known as a streptavidin-binding-protein (SBP) binds to streptavidin, but connected to the SBP are also the protein to be transported ('reporter') and a fluorescent protein. Upon the addition of biotin, the SBP

is released from the streptavidin as the biotin binds to the streptavidin. The SBP-reporter-fluorescent complex then exits the ER and can be followed the entire secretory pathway with fluorescence microscopy.

Because the addition of biotin is a nearly instantaneous process, RUSH allows for precise timing of release of the report complex. Another advantage is its versatility, as it can be used for many different proteins. In this thesis we mainly focus on the  $\alpha$ -mannosidase-II, generally referred to as ManII. The ManII protein is retained in the Golgi apparatus after trafficking, meaning that the data we obtain will only contain transport *towards* the Golgi, greatly simplifying the analysis as we will not have to separate pre and post-Golgi trafficking. Figure 1.4 shows two frames containing three cells (denoted by the black dashed line) in a typical RUSH experiment of ManII trafficking. Initially, all the cargo is retained in the ER and the fluorescence should thus be diffuse throughout the cell around the nucleus. This is shown in the left panel: in all cells we observe the fluorescence in a ring around a dark centre. Once the ManII is trafficked, the Golgi will show up as a bright object. This is shown in the right panel, where we have denoted the location of the Golgi by the red circle. However, some of the cargo is either still in the ER or being trafficked, considering the fluorescence outside of the Golgi.



**Figure 1.4:** Two frames of the ManII transport images using the RUSH technique.

### 1.2.2 MODEL

Vesicles exiting the ERES are transported towards the ER over the microtubules. This is a stochastic process with the proteins detaching from and (re-) attaching to the microtubules randomly, while the vesicles move diffusely once detached. Several models have been developed to describe such intracellular transport processes<sup>6, 17</sup>, many in the light of virus trafficking<sup>18, 19, 20</sup>. In general, these models assume a two population model, with one population being cargo attached to a microtubule and another cargo freely diffusing in the cytoplasm. If one assumes that the timescale for attaching and detaching from the microtubules is much smaller than the transport timescale, the two populations can be assumed to be in equilibrium. In this assumption, known as a quasi-steady-state reduction, the two population model reduces to a Fokker-Planck equation. As the Fokker-Planck equation is functionally equivalent to an advection-diffusion equation, we hypothesise that we can model protein transport using an advection-diffusion equation :

$$\partial_t c = \nabla(D\nabla c - \vec{v}c) \quad (1.1)$$

where  $c$  is the concentration of the cargo,  $D$  a diffusion coefficient and  $\vec{v}$  an advection velocity. Equation 1.1 is thus the model we fit our data to. Note that the fluorescence images obtained from the RUSH experiment return an intensity  $I$  and not a concentration  $c$ , and hence we make the assumption  $c \propto I$ .

Many biological processes and reactions require a high concentration of some protein or lipid to occur. This can be achieved by physically separating proteins inside a membrane (consider the lysosome), but the cell contains several membrane-less organelles. These organelles thus require a different means of reaching high concentrations and the prime candidate is liquid-liquid phase separation. In this process a mixture of liquids A and B separates into two phases, one rich in A and one rich in B, due to the interactions between them. Phase separation can thus produce domains with a high concentration without membranes. It has been proposed as a model for early protocells<sup>21</sup> and is able to correctly describe several phenomena such as P-granules<sup>22</sup> and centrosome growth<sup>23</sup>.

We use a similar description for the Golgi, as its biogenesis contains strong clues which point towards phase separation as the process driving the biogenesis. The Golgi is able to form *de novo*, meaning that in cells from which the Golgi has been removed, it will reappear without any specific action. Ronchi et al<sup>8</sup> studied this in detail and found three phases of growth. In the first phase, vesicles are released from the ER, but no larger structures are formed and the vesicles disappear either due to fusion with the ER or degradation. In the second phase larger stack-like structures are formed, while in the third phase all these structures are clustered in a single location; the Golgi Ribbon is formed. Phase two has the mark-

ings of a concentration-dependent phase separation: once a critical concentration of vesicles is reached, the mixed state becomes unstable and the vesicles aggregate and fuse to form a Golgi stack. When such a transition occurs, not all vesicles in the system aggregate: an equilibrium between single vesicles and the aggregated vesicles will exist. Coarse graining such a system yields a dilute phase with a low concentration of vesicles and a dense phase. Interpreting this dense phase as the Golgi and the dilute phase as the cytoplasm, we thus describe the Golgi as a phase separated droplet. Also note that by describing the systems in terms of some coarse-grained vesicle concentration, we can apply phase separation - a theory normally used to describe membrane-less organelles - to a membrane delimited organelle: the Golgi.

The Golgi is highly dynamical organelle, taking up and budding off vesicles constantly and our model needs to account for this. We neglect the precise form of maturation (i.e. cisternal maturation versus vesicular transport) and model the maturation by considering two populations, immature and mature, with the immature population being converted into mature inside the droplet. This makes the droplet *active* and we thus describe the Golgi as an *active phase-separated droplet*.

### 1.2.3 BIOLOGICAL IMAGE ANALYSIS

Image analysis is a lively and ongoing subject in cell biology, with many new methods being developed constantly, especially with quantisation in mind<sup>24, 25, 26, 27</sup>. Techniques for quantifying intracellular transport roughly fall into two categories: single particle tracking (SPT) or correlation spectroscopy. SPT tracks fluorescent proteins or beads moving through the cell on a frame-to-frame basis, so that each particle's trajectory can be reconstructed. These trajectories can then be analysed to obtain information about the transport. The fluorescent movies obtained from the RUSH experiments are not clear enough to accurately localise the vesicles, so that SPT can not be used to analyse the transport. Methods based on correlation spectroscopy<sup>28, 29, 30</sup> rely on a general relationship between the fluctuations and the underlying density of particles and transport properties. These techniques thus require a nearly constant concentration, but the RUSH experiments show highly dynamical concentrations. Thus, none of the techniques we found are directly applicable to the RUSH data.

As stated, we hypothesise that we can describe the intracellular process by an advection-diffusion equation and we wish to confront this with the RUSH data. The question we are thus asking is a rather general one: how do we fit some spatiotemporal ( $nD+1$ ) data to a model? More specifically, since a model is most often presented in the form of a partial differential equation (i.e.  $df/dt = \alpha(x, t)df/dx + \beta(x, t)d^2f/dx^2 + \dots$ ), for what param-

eters  $\alpha(x)$ ,  $\beta(x)$ ... is the temporal evolution of a given dataset best described? We have developed and evaluated two different methods. Our first method approaches the problem rather directly by calculating spatial and temporal derivatives directly from the data using a technique known as image gradients. Our second method is based on a recently developed technique based on neural networks<sup>31</sup>. We will show that by encoding physics into the neural network, we are not only able to infer the optimal parameters (i.e.  $\alpha(x, t) = \alpha_o$ ), but even an optimal parameter *field*  $\alpha(x, t)$ .

#### 1.2.4 STRUCTURE AND MAIN QUESTIONS

The rest of this thesis is divided into two parts. In the first part we show the two model fitting methods we have developed and apply them to the RUSH experimental data. In the second part we connect the model we have for the intracellular transport to that of the Golgi and theoretically investigate it. In a chapter-by-chapter breakdown, we have the following:

- Part I - Model fitting and data analysis
  - Chapter 2 introduces the framework we have developed for model fitting spatiotemporal data using image gradients.
  - Chapter 3 applies the method developed in chapter 2 to experimental data.
  - Chapter 4 shows an alternative method for model fitting based on neural networks.
- Part II - Golgi as an active phase-separated droplet
  - Chapter 5 introduces the Cahn-Hilliard equation, which describes phase separation, an approximation of it known as effective droplet theory and develops our model.
  - Chapter 6 contains the predictions the model developed in chapter 5 and investigates the biological implications.
- Chapter 7 is the concluding chapter and summarizes all the findings from the previous chapters.

# 2

## Model fitting

In this chapter we introduce the method we have developed for fitting a model in the form of a PDE to spatiotemporal data. We start off with the general concept and subsequent section will elaborate on each step.

Assume we have access to experimental data of some process  $f(x, t)$ . Parallelly, we have also developed a model describing this process, but it is in the form of a PDE:

$$\partial_t f(x, t) = \lambda_1 \nabla^2 f(x, t) + \lambda_2 \nabla f(x, t) + \lambda_3 f(x, t) + \lambda_4 \quad (2.1)$$

We now wish to investigate if this model fits the data  $f(x, t)$  and what coefficient values  $\lambda_i$  best describe the dataset. To do so, we consider each term on the right of equation 2.1 in  $f(x, t)$  as some variable  $x_i$  and  $\partial_t f$  as  $y$ , so that we can rewrite it as:

$$y = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 + \lambda_4$$

If we thus can find the variables  $x_i$  and  $y$ , we can perform a fitting procedure such as least squares to obtain the coefficients  $\lambda_i$ . In other words, if we can calculate the spatial and temporal derivatives of our data, we can fit the model. Although the concept seems trivial, its implementation is not. Data is rarely noiseless and obtaining accurate derivatives from noisy data is notoriously hard, but it forms the heart of our method. In the case of biological data, segmentation into sub-areas is often required and the coefficients  $\lambda_i$  might not be constant but space- and time- dependent. The process of fitting the data thus has several steps:

1. Denoising and smoothing

2. Calculating derivatives
3. Segmenting
4. Fitting

In the next sections, we describe each step separately. Note that the method we present here has been developed empirically: there's no theoretical background as to why this particular combination should work. Instead, it's been developed by analysing the data, adapting each step on the go. Nonetheless, the resulting process is general and presents a new tool to quantify fluorescence microscopy. To illuminate the process, we have thus chosen to illustrate the effects of each step with RUSH experimental data instead of synthetic data.

## 2.1 STEP I - SMOOTHING AND DENOISING

The first step is to denoise and smooth the data, which is required for accurately calculating the derivatives. This is a very active area of research (especially in life sciences) and several methods exist<sup>35</sup>. After evaluating several methods, we have settled on the so-called ‘WavIn-POD’ method, introduced in 2016 by the authors of [36]. They show that this method outperforms several other advanced methods<sup>37</sup> by combining two existing methods: Proper Orthogonal Decomposition (POD) with Wavelet filtering (Wav). Both subjects are vast (especially Wavelet transform) and as we are only interested in the result, we only present a short introduction here, adapted from [36].

POD is closely related to Principal Component Analysis (PCA) in statistics and is already used in physics to analyse turbulent flows<sup>38</sup>. Similar to other transformations such as the Fourier transform, we wish to expand a function as the sum over a set of orthogonal functions:

$$f(x, t) = \sum_{n=1}^r E_n \alpha_n(x) \varphi_n(t) \quad (2.2)$$

where  $\alpha_n$  and  $\varphi_n$  are called respectively the spatial and temporal modes and  $E_n$  indicates the relative strength of each mode. However, contrary to a Fourier transform, the orthogonal functions are not some predetermined set, but are determined from the data. More specifically, the data is decomposed into its eigenvectors  $\alpha_n$  and  $\varphi_n$ , with  $E_n$  being the eigenvalues. Modes with high eigenvalues  $E_n$  ‘contribute more’ to the behaviour of  $f(x, t)$  than modes with lower eigenvalues and we can use this observation to denoise data by letting the sum of equation 2.2 only run over the most important modes. In fact, when plotting

the sorted  $\log_{10}$  spectrum of  $E_n$ , a 'knee' is observed: modes above the knee constitute the signal, whereas modes below are essentially noise. We show the eigenvalue spectrum of the ManII data in figure 2.1.

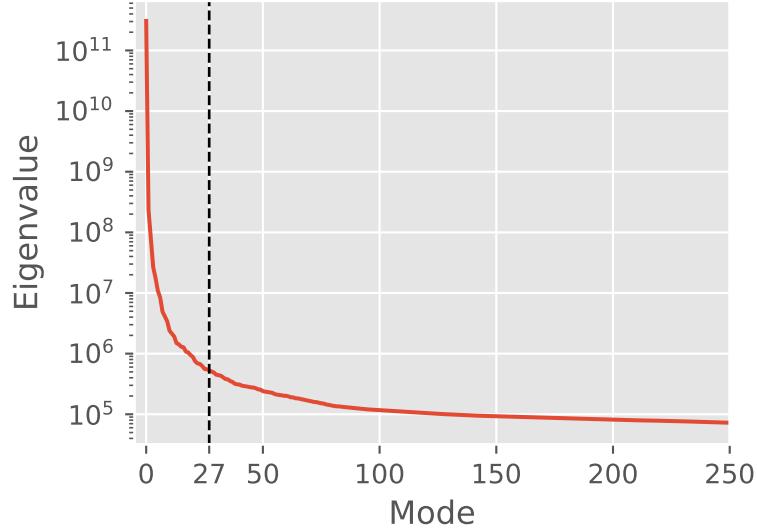


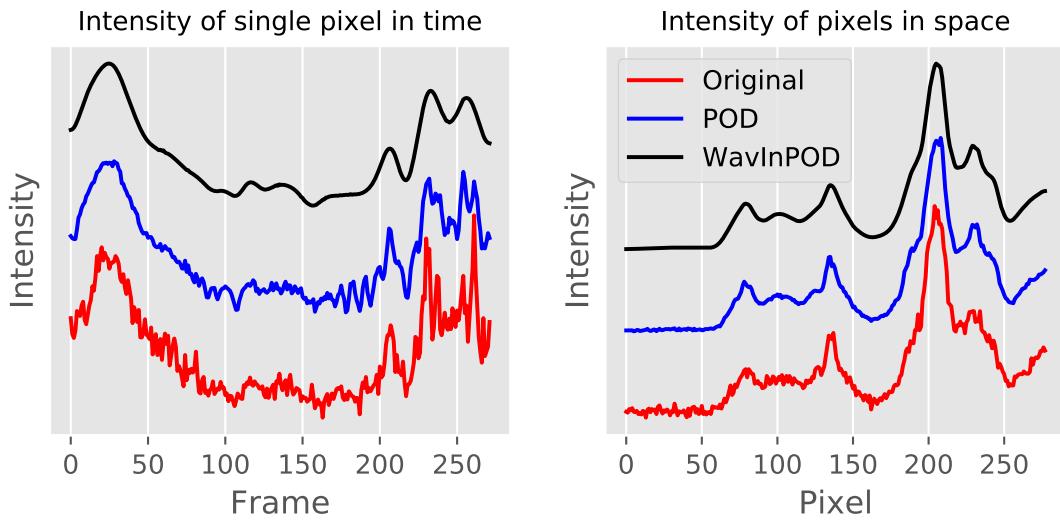
Figure 2.1: Eigenvalue spectrum of the POD of the ManII data. We have placed the cutoff at mode 27.

Note that although a knee is visible, exactly where to place the mode cutoff is not clear. Techniques to consistently determine the cutoff exist<sup>38, 39</sup>, but yielded unsatisfactory performance when applied to the shown spectrum. We thus determined the cutoff by hand by checking if the data after applying the POD still showed the same trend and features.

Consider an oscillation consisting of two frequencies  $f_1$  and  $f_2$ . Initially, the signal is solely composed of  $f_1$ , but after some time the frequency is switched to  $f_2$ . Fourier transforming such a signal would yield two delta peaks at  $f_1$  and  $f_2$ : it returns with infinite precision which frequencies are present in the signal, but not when. In a wavelet transform, this infinite precision in the frequency domain is traded for information in the time domain by the uncertainty theorem: a wavelet transform won't give back the frequencies  $f_1$  and  $f_2$  with infinite precision, but it will state when they are present.

In practice, wavelet transforming returns the signal as an approximation plus a set of details for each datapoint. We can then filter the signal by applying by some sort of cutoff to the details. Note that cutting off the details will only change the signal locally; we thus do not lose any sharpness anywhere else in the data as one would with Fourier filtering. Wav-inPOD combines these two techniques by applying wavelet filtering to the POD modes.

First, the dataset is decomposed into its POD modes and the energy spectrum is analysed to select a cutoff mode. All retained modes are wavelet filtered and are then retransformed to give the denoised and smoothed signal. In figure 2.2 we show the results of the smoothing in the time and spatial domain. In the left panel we show the signal of a single pixel in time, while we plot a line of pixels in a single frame in the right panel. The red lines denote the original (unfiltered) signal, the blue line the effect of just applying POD filtering and the black one the result of the WavInPOD technique. Note that the effect of the wavelet filtering is to smooth the signal significantly and in comparing the original data to the filtered data that we've retained the sharpness of the features whilst obtaining a significantly smoother signal.



**Figure 2.2:** Effect of POD with a cutoff of 27 and wavelet filtering with a level 3 db4 wavelet. Left panel shows the result in the time domain, right panel in the spatial domain. Lines have been offset for clarity.

## 2.2 STEP 2 - DERIVATIVES

After having denoised the images, we calculate the spatial and temporal derivatives. Obtaining correct numerical derivatives is hard and becomes much more so in the presence of noise<sup>40</sup>. Next to a finite-difference scheme, one can for example (locally) fit a polynomial and take its derivative.<sup>41</sup> However, the computational cost of these methods is high and they do not scale well to dimensions higher than one. We thus require an alternative method. In fact, obtaining the gradient of a 2D discrete grid has another subtlety which we need to address.

Naively, one could obtain the gradient of a 2D grid by taking the derivative using a finite difference scheme with respect to the first and second axis. If there are features on the scale of the discretisation ( $\sim$  few pixels), such an operation will lead to artefacts and underestimate the gradient. These issues have long been known and several techniques have been developed to accurately calculate the gradient of an ‘image’. The most-used image-gradient technique is the so-called Sobel operator and we derive its structure here. Consider a basic central finite difference scheme:

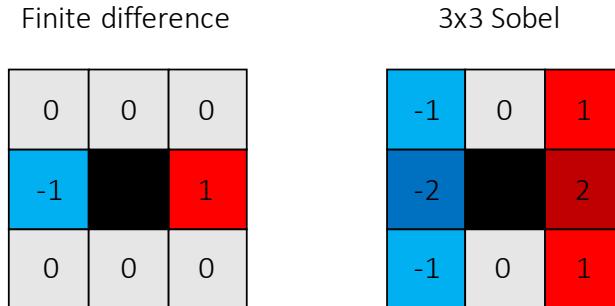
$$\frac{df(x_i)}{dx} \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$$

where  $h$  is defined as  $x_{i+1} - x_i$ . Applying this operation to a set of three pixels thus returns the derivative of the middle pixel. We can rewrite this as a matrix  $S$

$$S = \frac{1}{2} \cdot [-1 \quad 0 \quad 1]$$

which, when applied *element wise* to the three pixels, returns the middle pixels’ derivative. Applying the matrix  $S$  element wise to each point in a dataset is known as an convolution and convolving a matrix  $A$  with the matrix  $S$  yields its derivative:

$$\partial_x A \approx A * \frac{1}{2} [-1 \quad 0 \quad 1]$$



**Figure 2.3:** In the left panel we show how a finite difference operator would be applied to the black pixel. The right panel shows this for the Sobel operator.

As stated, this operation is inaccurate and introduces artefacts. To improve this, we wish to include the pixels on the diagonal of the pixel we’re performing the operation on as well (see figure 2.3). The distance between the diagonal pixels and the center pixel is not 1 but

$\sqrt{2}$  and the diagonal gradient also needs to be decomposed into  $\hat{x}$  and  $\hat{y}$ , introducing another factor  $\sqrt{2}$ . We thus obtain the classic  $3 \times 3$  Sobel filter in the  $\hat{x}$  and  $\hat{y}$  direction:

$$G_x = \frac{1}{8} \cdot \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} G_y = \frac{1}{8} \cdot \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Increasing the size of the Sobel filter increases its accuracy and we've implemented a  $5 \times 5$  operator. The matrix implementation is also beneficial from a computational standpoint, as convolutional operations are very efficient and one can derive a Sobel operator for arbitrary dimensions. Separate methods to calculate second order derivatives exist, we simply apply the Sobel operator twice. We also make use of the derivatives to segment the movie. We show this in the next section.

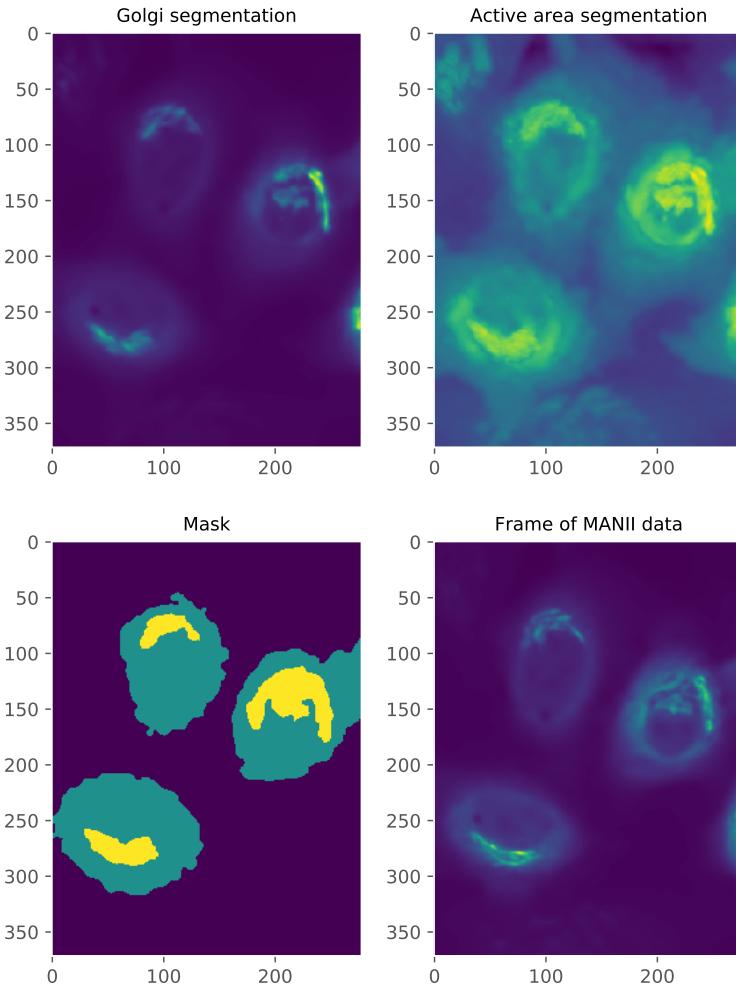
### 2.3 STEP 3 - SEGMENTATION

In the case of the RUSH data, obtained images and movies often contain multiple cells. Each of these cells can be further segmented into two more areas of interest: the cytoplasm, which is where we want to fit our model and the Golgi apparatus. We wish to make a mask which allows us to separate the cells from the background and divide each cell into cytoplasm or Golgi. Figure 1.4 shows two typical frames in the MANII transport cycle. Note that no sharp edges can be observed, especially once the MANII localises in the Golgi. No bright field images were available as well, together making use of techniques such as described in [42] unavailable. We have thus developed our own method which depends on the intensity and its time derivative. It consists of four steps:

1. Normalize the intensity  $I$  and its time derivative between 0 and 1.
2. Sum all the frames over some quantity. To separate the Golgi, we determined  $\sum_n I(x, y, t_n)$ , while we calculated  $\log_{10}(\sum_n I(x, y, t_n) \cdot \partial_t I(x, y, t_n))$  for the cytoplasm.
3. Threshold the image to obtain the mask. This is either done automatically through an Otsu threshold or by manually adjusting the threshold until desired result.
4. The mask is post-processed by filling any potential holes inside the mask.

We show this process in figure 2.4. The upper two panels show the images obtained after performing the summing operation for the Golgi and cytoplasm (also referred to as 'active

area') respectively, while the lower left panel shows the final mask obtained after thresholding these two images. For comparison, we have plotted a frame of the data to compare the mask to.



**Figure 2.4:** Four panels showing the different stages of making the mask. From segmenting the upper two panels we determine the Golgi and active area, leading to the mask in the lower left. This can be compared to the actual data in the lower right frame.

## 2.4 STEP 4 - FITTING

The final step in our method is to fit our model to the data. By having determined both the spatial and temporal derivatives, we have effectively reduced the movie to an  $m$  by  $n$  sized dataset, where  $m$  is the number of datapoints and  $n$  the amount of features calculated. In the case of the RUSH data movie,  $m$  is number of pixels multiplied by the number of frames, while  $n = 5$  as we calculate  $\partial_x I$ ,  $\partial_{xx} I$ ,  $\partial_y I$ ,  $\partial_{yy} I$  and  $\partial_t I$ . The fitting thus becomes a generic problem, which can be solved by virtually any fitting method. We use least-squares, but one could use for example a Bayesian method. Each fitting method assumes that the fitting parameters are constant across a dataset however. Given the nature of the cell, the diffusion constant and advection will not be constant: they will be spatially dependent and it is not unlikely that they will show some temporal dependence as well. On a small enough scale however, we can reasonably approximate these fields as constant and here we can perform a fit. To not lose any spatial resolution and prevent artefacts, we use the sliding-window technique, which is illustrated in figure 2.5. In a small window around a pixel, we perform the fitting procedure, thus yielding the diffusion coefficient and advection velocity for that pixel. We then move the window to the next pixel, thus finding diffusion and advection fields with a similar resolution as the data. In the next chapter we apply this method to the RUSH experimental data.

Sliding window

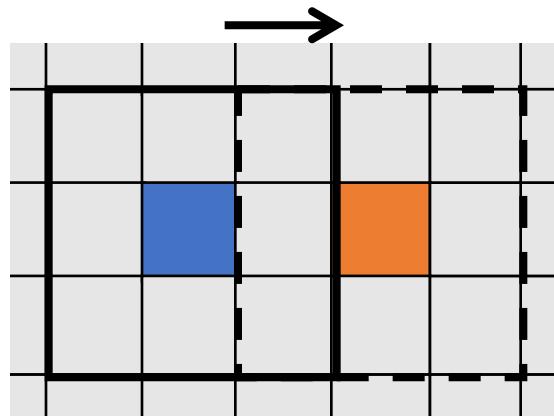


Figure 2.5: Schematic overview of the sliding window technique. The solid black line encompasses an area around its blue coloured central pixel and the fit output is assigned to that pixel. We then move the window (dashed black line) and perform the fit for the orange coloured pixel.

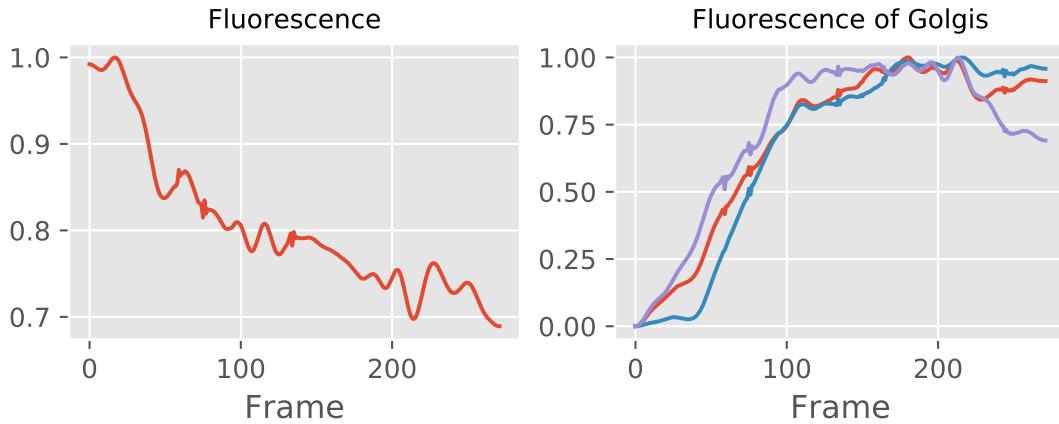
# 3

## Data analysis

In this chapter we apply the method developed in the previous chapter to the ManII experimental data obtained using the RUSH technique. We first present an initial analysis of the data by investigating the fluorescence curves of several areas of interest and study movies' time derivative. We then analyse the results of the least squares fit.

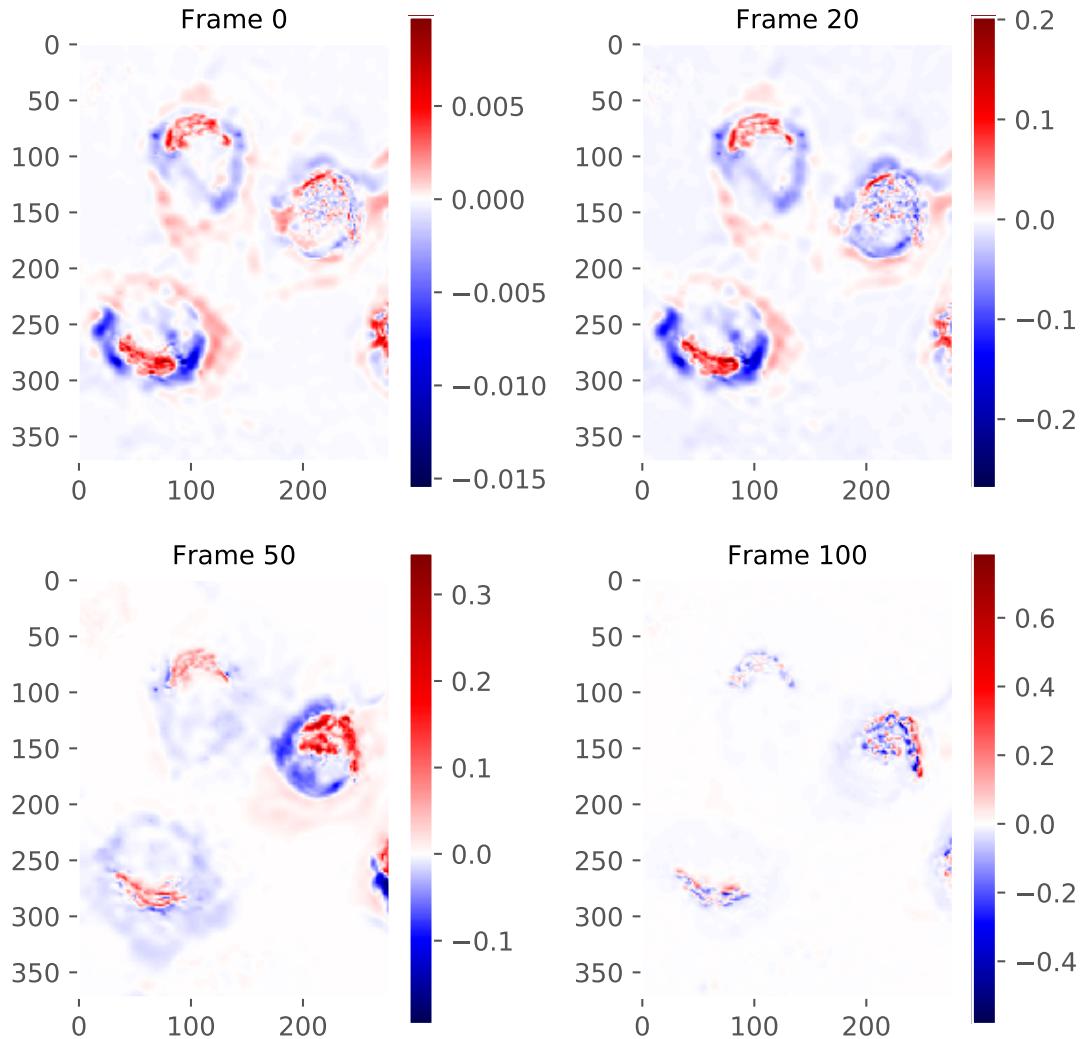
### 3.1 INITIAL ANALYSIS

We stated in the introduction that we assume that the concentration is proportional to the intensity of the fluorescence,  $c \propto I$ . To test this assumption, we plot the time-evolution of the fluorescence of the entire cell. We normalise the fluorescence between 0 and 1 before computing the mean over each frame, to get rid of the background in our statistics. The left panel of frame ?? shows the average fluorescence of each frame, normalised on the maximum average intensity. In the right panel we plot the same quantity but for the fluorescence in each of the three cells' Golgi. Note a significant drop of almost 30% in total fluorescence between the initial and final frame. We recognise two regimes: a strong initial drop up until frame 100 and a slower decay after. The right panel shows a saturation of the fluorescence in the Golgi after frame 100, so we attribute the decrease in total fluorescence after frame 100 to photobleaching. The first regime is more troublesome, as this strong decrease is not explained by photobleaching and thus casts strong doubts on our assumption that  $c \propto I$ . Since this drop is spread out over a hundred frames, on a frame-to-frame basis this effect can be neglected in our model.



**Figure 3.1:** Left panel: Normalised total fluorescence per frame. Normalised fluorescence in each of the three cells' Golgi.

In the right panel we observe that all three curves show a roughly linear increase in fluorescence. The blue line seems to have some sort of delay, but also increases linearly after this delay. The cell indicated by the purple line shows a significant drop at frame 200, but since the ManII protein is retained in the Golgi, this is not caused by any type of intracellular transport and thus not of interest to us. The linear increase and common pattern suggests that the transport properties are not concentration dependent at these concentrations. We now study the time derivative of the fluorescence. We have plotted it at frame 0, 20 50 and 100 in figure 3.2.



**Figure 3.2:** The determined time derivative four different frames of the ManII RUSH experiments.

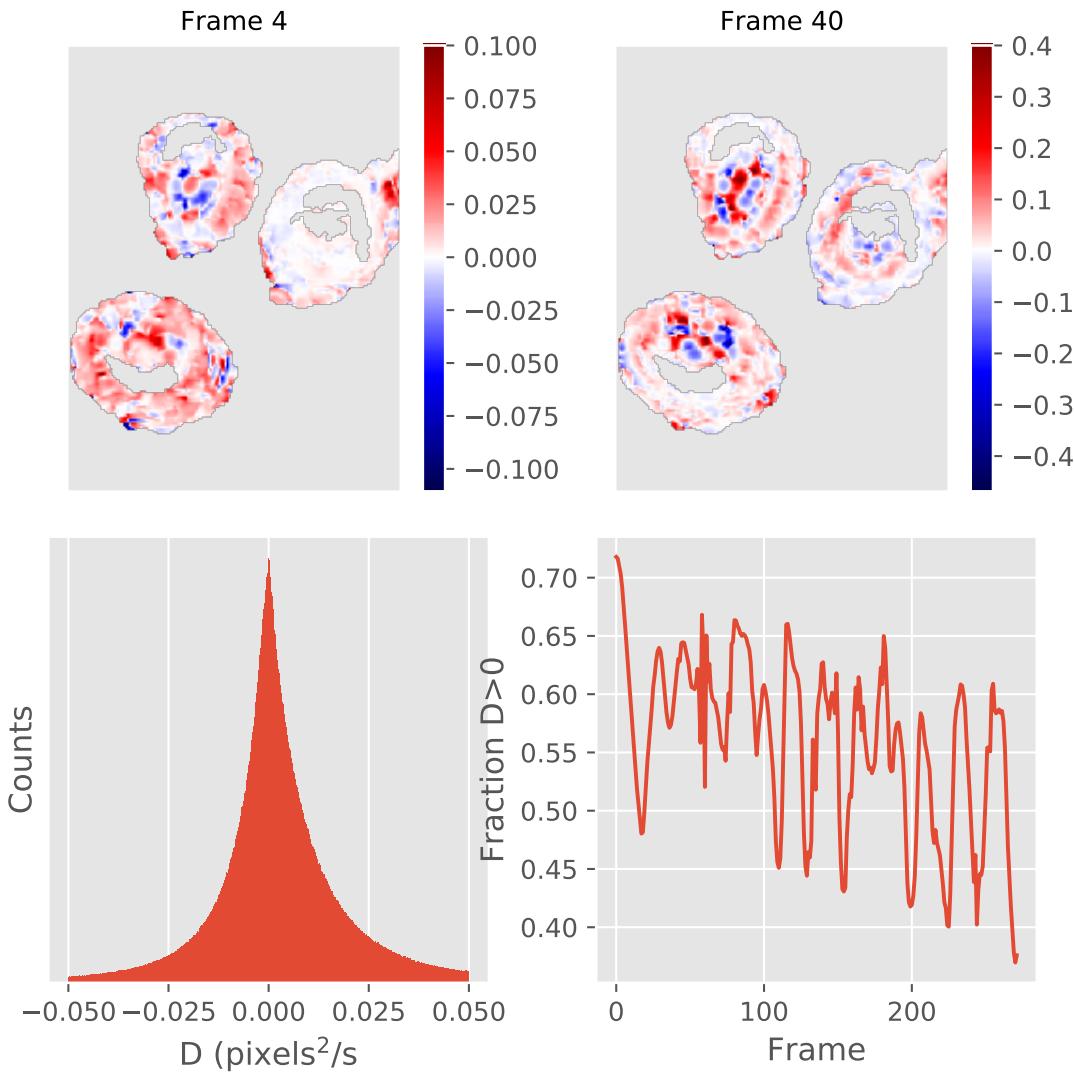
Although figure ?? has shown that  $c \propto I$  is a doubtful assumption, an increase in fluorescence will mean an increase in concentration. Areas where the time derivative is positive thus correspond to a concentration increase, whereas areas with a negative time derivative correspond to a concentration decrease. In figure 3.2, positive areas are shaded, while negative are shaded blue. As expected, the Golgi shows up in each cell as a bright red object. Note however that we also observe red areas towards the edges of the cells. As the concentration close to the Golgi decreases due to trafficking, the blue area moves outwards and slowly takes over the red area. As to the cause of this outer red ring in each cell, we speculate

this is caused by a diffusion: reporters exiting the ER through the ERES initially diffuse, increasing the concentration in some areas.

### 3.2 ANALYSIS OF LEAST-SQUARES FIT

In this section we present the results of our least squares fit. We have used a 7 by 7 pixels area in the spatial domain to perform the sliding window operation and have fitted each frame of the movie independently. The sliding window operation returns diffusion and advection fields the same size as our data. Due to the physical limitations of print media, we only show several frames of the results here. Since a picture is worth more than a thousand words, an animation of 272 frames must be worth a book and we refer the reader to our [Github](#) to find the full results.

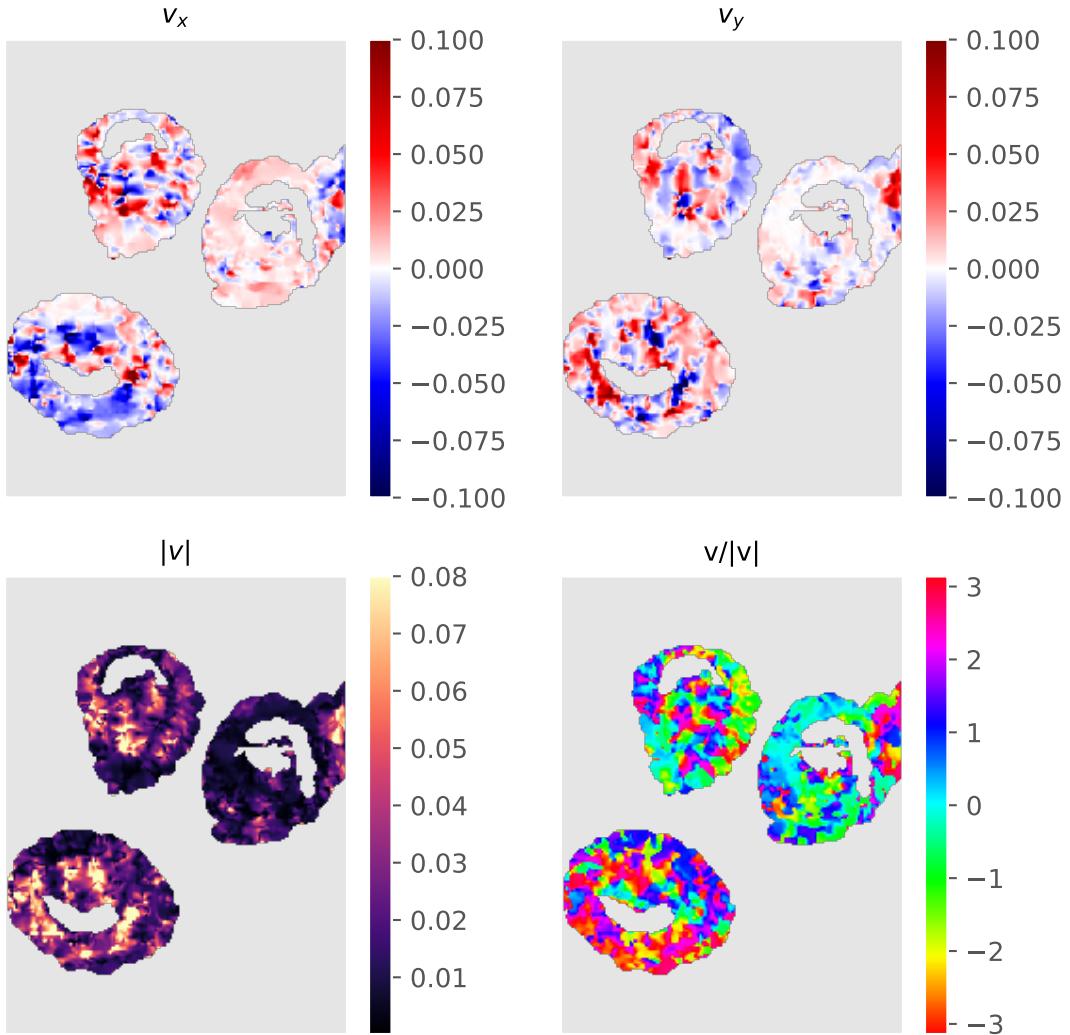
We show the inferred diffusion field at frame 4 and 40 in figure 3.3, together with the distribution of diffusion coefficients throughout the entire movie and the fraction of positive diffusion constants per frame.



**Figure 3.3:** Analysis of the inferred diffusion field. The upper row shows the inferred field at two frames, while the lower row shows the distribution of values and the fraction of physical values as a function of time.

We observe structures larger than the fitting window which, although varying in time, do not differ greatly on a frame-to-frame basis. This means that our fit is capturing at least some of the underlying dynamics. On the other hand, we observe many areas with a negative diffusion coefficient and the inferred diffusion coefficients are on the order of  $10^{-4} \mu\text{m}^2/\text{s}$ -orders of magnitude than expected. In the lower left panel we plot the distribution of values, which shows that roughly 60% of the inferred field has a positive diffusion

coefficient. In the lower right panel we have determined this fraction as a function of time. It shows that, save for a few initial frames, this fraction is not (strongly) time-dependent. Note however that results are skewed due to its strong peak around 0; many coefficients are negative but extremely close to zero (e.g  $-10^{-5}$ ). Negative diffusion coefficients could indicate to clustering, but could also be the result of an incorrect fit. We investigate this in depth after studying the advection profiles, which we show in figure 3.4. In the four panels we show the inferred velocity in the  $\hat{x}$  and  $\hat{y}$  direction in the upper two panels, and the magnitude and angle in the lower two.

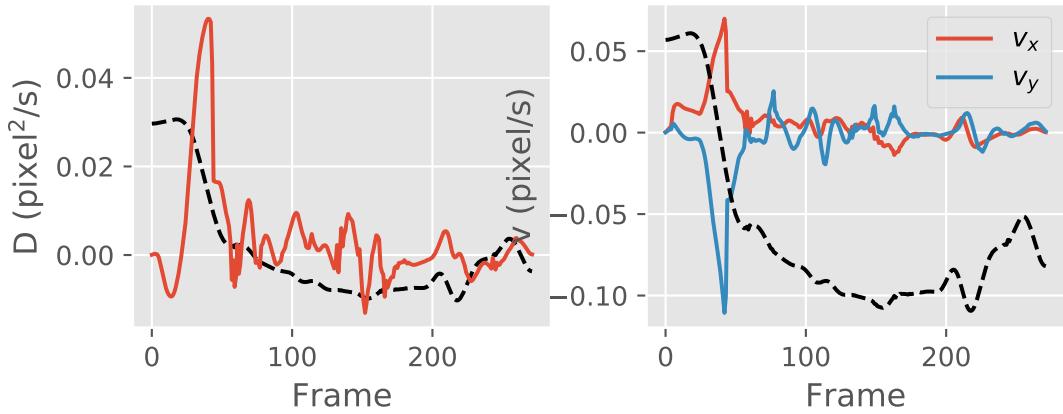


**Figure 3.4:** Analysis of the velocity fields. The upper rows show respectively  $v_x$  and  $v_y$ , while the lower row shows the magnitude and angle.

Similar to the diffusion, we observe patterns both in time and space bigger than our fitting window, meaning that the fit isn't completely random. On the other hand, we are not able to discern any specific flow patterns from the figures in 3.4. For example, we would expect a rainbow-like pattern pointing towards the Golgi in the lower right corner, but no such pattern is observed.

To gain more insight into our fit, we analyse a single pixel in time. Figure 3.5 shows the dif-

fusion constant and advection velocities as fitted at this pixel as a function of time. The scaled and translated intensity of the pixel is plotted by the black dashed line as a reference.



**Figure 3.5:** Diffusion and advection velocities of a single pixel in time. We've plotted the scaled and translated signal as a black dashed line to show the correlation.

The signal of this pixel remains roughly constant for the first 10 frames and then decreases to noise level. Note that during this initial constant phase, the diffusion constant is negative. Once the signal starts decreasing, or, in other words, cargo starts flowing, we see a physical diffusion constant and non-zero velocities. Once the signal returns to around noise-level around frame 50, the inferred velocities and diffusion constant seem to become random around 0. In other words, our method seems to work when the signal is changing but struggling when the signal is either constant or at noise level. We observe similar behaviour in other pixels, so we contribute the unphysical diffusion values to constant and noise-level signal.

Using the Einstein Stokes relation, the diffusion constant of a vesicle with a 50nm radius should be on the order of  $5\mu\text{m}^2/\text{s}$ . Although this is an upper bound, as the vesicle can hardly be considered freely diffusing through the cytoplasm, this is still several orders of magnitude higher than the observed diffusion constants of  $10^{-4}\mu\text{m}^2/\text{s}$ . One possible explanation for this difference is the ‘mixing’ of the transport fluorescence with the fluorescence of the ER. After the addition of biotin, the fluorescent cargo gets released, but still has a finite residence time in the ER. Since the obtained images are projected over an axis, changes in fluorescence we observe can be both due to intracellular transport as well as processes inside the ER. If these processes have different timescales, this can strongly affect the inferred coefficients. Also note that we've assumed that the intensity of a pixel describes a coarse-

grained concentration which we can describe by an advection-diffusion equation. Once the size of the vesicles becomes on the order of the pixel size, this assumption breaks down. In the case of the ManII trafficking, the pixels are roughly two to three times the size of a vesicle, meaning that we are at the limits of our assumption.

### 3.3 CONCLUSION

We've applied the method developed in the previous chapter to the RUSH trafficking data of the ManII protein. Both the diffusion and the advection fields show patterns larger than the fitting window, but no clear structure can be discovered. Furthermore, roughly 40% of the diffusion coefficients were negative, but we also can attribute many of these negative values to either extremely close to zero or a wrong fit due to constant data. Conclusively, we're unable to determine if intracellular transport can be described by an advection-diffusion equation. We present our recommendations to the experimentalists and possible improvements of the fitting method in the conclusion.

# 4

## Physics Informed Neural Networks

In the previous chapters we showed the difficulties in fitting a model in the form of a partial differential equation to spatio-temporal data. The method we developed was a classical numerical approach, separating the problem into several substeps such as denoising, smoothing and numerical differentiation. The main weak points of the developed method are the calculation of numerical derivatives and its rather crude fitting method. In this chapter we present an alternative technique, generally referred to as Physics Informed Neural Networks (PINNs), which solves these issues. Although only recently introduced, it has already shown impressive performance in fitting models and numerically solving equations<sup>31, 43, 44, 45, 46</sup>. Neural networks are a new technique in physics and this chapter also serves as an introduction to neural networks in general. The chapter has the following structure:

- Neural Networks - This part will cover the basics of neural networks: their inner workings, training and other general features.
- Physics Informed Neural networks - In this second part we introduce the concept behind PINNs, use it to solve a toy problem and apply it to our RUSH data.

### 4.1 NEURAL NETWORKS

Normally when programming a computer to perform some task, we break the problem into smaller pieces and write down precise instructions. Often, a model of the underlying

process is also needed to transform some input into an output. The performance of the algorithm is then only as good as the underlying model and when dealing with complex processes, such models often become intractable or oversimplified. Artificial Neural Networks (ANNs) are a different approach to such a problem. Instead of being *programmed*, they are *trained* and hence ‘learn’ an underlying model. In a process known as *supervised learning*, the network is given inputs and the desired outputs for each input. Training the network then consists of adjusting its internal parameters until the predictions match the desired outputs. In the next sections we discuss how to adjust these parameters.

#### 4.1.1 ARCHITECTURE

*An excellent introduction is given by Michael Nielsen in his freely available book “Neural networks and deep learning.” The following section has been strongly inspired by his presentation.*

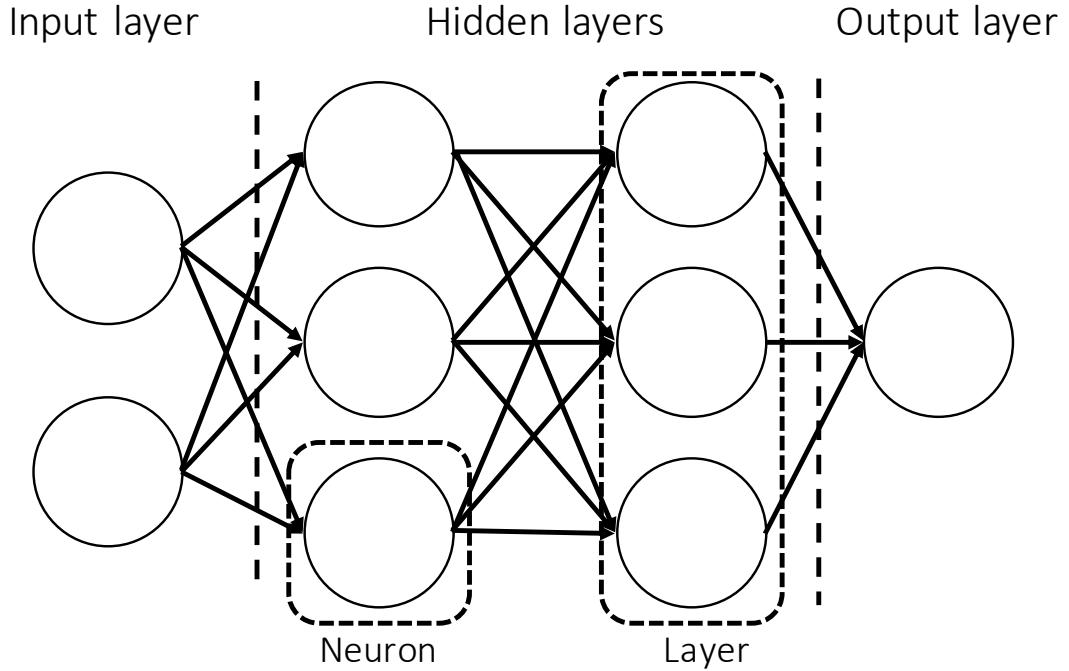
At the basis of each neural network lies the neuron. It transforms several inputs into an output in a two-step process. In the first step, the inputs  $x$  are multiplied with a weight matrix  $w$  and a bias  $b$  is added:

$$z = wx + b$$

$z$  is called the weighted input and is transformed in the second step by the neuronal *activation function*  $\sigma$ . This gives the output of the neuron  $a$ , also known as the activation:

$$a = \sigma(z) = \sigma(wx + b) \quad (4.1)$$

The activation introduces non-linearity into the network and hence is crucial; without it a neural network would merely be several matrix multiplications. The classical activation is a  $\tanh$ , i.e  $\sigma(z) = \tanh(z)$ , but many other forms exist, each having its benefits. Several neurons in parallel constitute a *layer* and several layers can be connected to create a network. The layers in the middle of the network are referred to as *hidden layers*. An example of such a network with two hidden layers is shown in figure 4.1.



**Figure 4.1:** Schematic overview of a neural network. The left layer is known as the input layer, the right layer as the output layer and the layers inbetween are referred to as hidden layers.

#### 4.1.2 TRAINING

Consider again equation 4.1. In a network with multiple layers, it is useful to express the activation  $a^l$  of layer  $l$  in terms of the activation of layer  $l-1$ , so that 4.1 becomes :

$$a^l = \sigma(z^l) = \sigma(w^l a^{l-1} + b^l) \quad (4.2)$$

where  $w^l$  and  $b^l$  are respectively the weight matrix and bias of layer  $l$ . As stated, training a neural network means adjusting the weights  $w^l$  and biases of each layer  $b^l$  until the output of the neural network  $a^L$  - the activation of the last layer  $L$  - matches the desired output  $y_i$ . We thus require a metric to define the difference between the prediction and the desired output. This metric is known as the *cost function*  $\mathcal{L}$  and one of the most commonly used cost functions is the mean squared error:

$$\mathcal{L} = \frac{1}{2n} \sum_i |y_i - a_i^L|^2 \quad (4.3)$$

where  $n$  is the number of samples,  $y_i$  the desired output of sample  $i$  and the prediction of the network given the inputs of sample  $i$ . As the cost function is a measure of the difference between the prediction and desired outputs, training a neural network comes down to minimizing the cost function. Such minimization problems are solved by gradient descent techniques.

Gradient descent techniques are based on the fact that given some position, the minimum from that position can be reached by following the steepest descent. Thus, given a function  $f(x)$  to minimize with respect to  $x$ , we guess an initial position  $x_n$  and iteratively update it until it converges:

$$x_{n+1} = x_n - \gamma \nabla f(x_n) \quad (4.4)$$

$\gamma$  is known as the learning rate and sets the ‘stepsize’. Although this is an iterative technique, if the minimization problem is convex (i.e. no local minima), gradient descent is guaranteed to converge to it. Note that gradient descent requires calculation of the derivative with respect to the variables of the function to be minimized. In other words, one needs to know the derivative of the cost function with respect to each of the weights and biases in the network. A naive finite difference scheme would quickly grow computationally untractable, even for networks with just two hidden layers. Alternatives to gradient descent exist, but all require calculation of the derivatives. In the next section we present an algorithm which is able to efficiently calculate these derivatives.

## BACK PROPAGATION AND AUTOMATIC DIFFERENTIATION

In this section we introduce the so-called *backpropagation* algorithm. The backpropagation algorithm allows for the efficient calculation of the cost function derivatives in a neural network. For simplicity, we move away from a vector notation and introduce  $w_{jk}^l$ , the weight of the  $k$ -th neuron in layer  $l-1$  to neuron  $j$  in layer  $l$  and  $b_j^l$ , the bias of the neuron  $j$  in the  $l$ -th layer. We introduce the error of neuron  $j$  in layer  $l$  as:

$$\delta_j^l = \frac{\partial C}{\partial z_j^l}$$

We can rewrite this using the chain rule as:

$$\delta_j^l = \sum_k \frac{\partial C}{\partial a_k^l} \frac{\partial a_k^l}{\partial z_j^l}$$

The second term on the right is always zero except when  $j = k$ , so the summation can be dropped. Given equation 4.1, we note that  $\partial a_j^l / \partial z_j^l = \sigma'(z_j^l)$ . For the last layer  $l = L$ , we can directly calculate the derivative, resulting in:

$$\delta_j^L = |a_j^L - y_j| \sigma'(z_j^L) \quad (4.5)$$

Equation 4.5 relates the error in the output layer to its activation and weighted input.

Again using the chain rule, we can express the error in a layer  $l$ ,  $\delta_j^l$ , in terms of the error in the next layer,  $\delta_j^{l+1}$ :

$$\delta_j^l = \sum_k \frac{\partial C}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial z_j^l} = \sum_k \delta_k^{l+1} \frac{\partial z_k^{l+1}}{\partial z_j^l}$$

Using equation 4.2, we obtain after substitution:

$$\delta_j^l = \sum_k \delta_k^{l+1} w_{kj}^{l+1} \sigma'(z_j^l) \quad (4.6)$$

Equation 4.5 gives us the error in the final layer, while equation 4.6 allows us to propagate the error back through the network - hence the algorithm is named backpropagation. Two more expressions are needed to relate the error in each neuron  $\delta_j^l$  to the derivatives with respect to the weights and biases. Making use yet again of the chain rule gives the last two backpropagation relations:

$$\frac{\partial C}{\partial b_j^l} \frac{\partial b_j^l}{\partial z_j^l} = \frac{\partial C}{\partial b_j^l} = \delta_j^l \quad (4.7)$$

$$\delta_j^l = \sum_k \frac{\partial C}{\partial w_{jk}^l} \frac{\partial w_{jk}^l}{\partial z_j^l} \rightarrow \frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l \quad (4.8)$$

Given the four fundamental backpropagation relations, we state the algorithm. It consists of four steps:

1. Complete a forward pass, i.e., calculate  $a_j^L$ .
2. Calculate the error in the final layer using 4.5 and propagate it backwards using 4.6 to obtain the error in each neuron. Using 4.7 and 4.8, calculate the derivatives required for the minimizer.
3. Perform a minimization step (e.g. equation 4.4) and update the weights and biases.

4. Return to step one until the minimization algorithm in step three converges.

Mathematically, back propagation is a version of a more general technique known as automatic differentiation. Suppose we want to calculate the derivative of some data  $f(x)$ . Symbolic differentiation would give the most precise answer, but often the function  $f$  is not known. Furthermore, even if  $f$  would be known, it quickly becomes too hard to calculate a symbolic derivative of a complex function  $f$ . One could then turn to numerical differentiation using some finite difference scheme or locally fitting a polynomial whose derivative is then calculated. All these methods require relative closely spaced data and are very sensitive to noise. Automatic differentiation is a third type of differentiation which allows for the precise calculation of derivatives. At its fundamental level, any computational operation, no matter how complex, is a long string of elementary operations whose derivative is easily determined. Using the chain rule, we can then calculate the derivative of any computation in terms of these smaller elementary operations. To see this, consider a function  $f(x) = a + bx$ . Writing this in terms of elementary operations gives:

$$f(x) = a + bx = w_1 + w_2 w_3 = w_1 + w_4 = w_5$$

The derivative of each subexpression  $w_i$  is easily calculated:

$$w'_1 = 0, w'_2 = 0, w'_3 = 1, w'_4 = w'_2 w_3 + w_2 w'_3, w'_5 = w'_4 + w'_1$$

The derivative of  $f$  is then:

$$f' = w'_5 = w'_4 + w'_1 = (w'_2 w_3 + w_2 w'_3) + w'_1 \quad (4.9)$$

We have thus expressed the derivative of  $f$  in quantities we know and indeed, after filling in the remaining derivatives we obtain  $f' = w_2 = b$ . Note the similarity to backpropagation; in automatic differentiation we are only interested in the final expression on the right of equation 4.9, whereas in backpropagation we wish to know the intermediate derivatives (i.e.  $w'_5, w'_4$ ) too. Back propagation is thus a version of automatic differentiation in which the intermediate values are calculated too. In the next section we show that automatic differentiation enables easy encoding of physics into a neural network, leading to a so-called Physics Informed Neural Network (PINN).

## 4.2 PHYSICS INFORMED NEURAL NETWORKS

In this section we introduce Physics Informed Neural Networks (PINNs), a recently developed technique<sup>31, 46</sup> which merges physical models and neural networks. We first introduce how PINNs encode physical laws and models in neural networks and discuss why

this yields such a powerful technique. This is illustrated by applying it to a simple diffusive process and show that even in the presence of noise, PINNs can accurately infer a (spatially-varying) diffusion constant. We then apply a PINN to the RUSH data and end the chapter with our conclusions.

#### 4.2.1 THE CONCEPT

Consider a set of spatiotemporal experimental data,  $u(x, t)$  and a model which describes the temporal evolution of this dataset:

$$\partial_t u = \lambda_1 + \lambda_2 u + \lambda_3 \nabla u + \lambda_4 \nabla^2 u = f(I, u, u_x, \dots) \quad (4.10)$$

We now wish to know which value for the parameters  $\lambda_i$  best describes the dataset  $u(x, t)$ . Naively, one could train a neural network on a training set created by numerically solving 4.10 for different values  $\lambda_i$  and then feed this network the experimental data  $u(x, t)$ . Although theoretically this yields the correct result, for complex processes such as a Navier-Stokes flow or the Schrodinger equation this quickly grows intractable due to the massive amount of training data required for an accurate prediction.

PINNs circumvent this issue by directly encoding physical laws and models such as 4.10 into the neural network. We can write any PDE as:

$$g = -\partial_t u + f(I, u, u_x, u_{xx}, u^2, \dots) \quad (4.11)$$

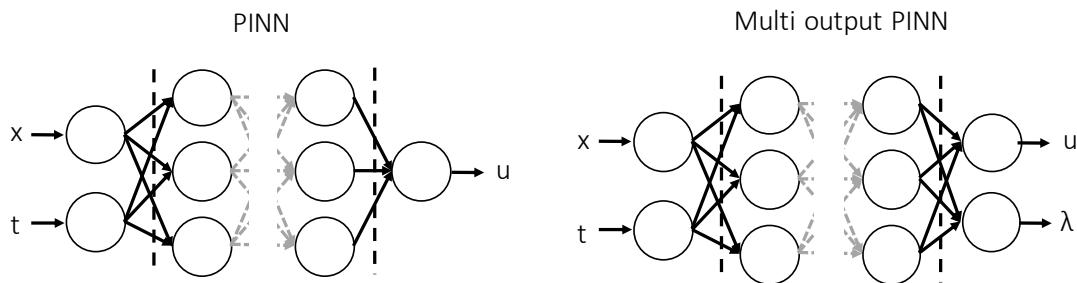
This function  $g$  can be added to the cost function, because to satisfy the PDE,  $g \rightarrow 0$ :

$$\mathcal{L} = \frac{1}{2n} \sum_i |u_i - d_i^L|^2 + \frac{l}{n} \sum_i |g_i|^2$$

where  $l$  sets the effective strength of the two terms. By adding  $g$  to the cost function, it acts as ‘physics-regularizer’ and unphysical solutions are penalized; we have thus encoded the physics directly into the neural network. Since neural networks also return high precision derivatives through automatic differentiation, equation 4.11 can be accurately determined. Note that while we know the form of  $g$ , its coefficients  $\lambda_i$  are unknown. However, we can treat the coefficients as variables of the cost function, i.e.  $\mathcal{L}(w^l, b^l, \lambda)$  and thus by training the network on the dataset  $u(x, t)$ , we automatically infer the coefficients. Consequently, we don’t need a vast set of training data, as we solve the problem *by* training the network.

Theoretically, PINNs should not only be able to infer constant coefficients, but also coefficient *fields*. Instead of treating the coefficients as a variable to be optimized, we add another

output to the network. Such a network is known as a multi-output PINN and the difference between a single and multi output network is shown in figure 4.2. PINNs can also be used to numerically solve PDEs. By removing the mean squared error term from the cost function but adding initial values and boundary conditions, training the network will now result in the network learning the solution to the PDE  $g$ , whilst respecting the given boundary and initial conditions. This alternative means of numerically solving a model doesn't need advanced meshing of the problem domain required in computational fluid dynamics or carefully constructed (yet often unstable) discretization schemes, as it requires the physics to be fulfilled at every point in the spatiotemporal domain.



**Figure 4.2:** Left panel: a single output PINN. Right panel: A multi-output PINN. The network now also predicts the coefficients values at each data point.

#### 4.2.2 PINNs IN PRACTICE

Before applying a PINN to the RUSH data, we study a toy problem to gain more insight into its behaviour. We also prove that a PINN is able to correctly infer a coefficient field from noisy data. Our toy problem of an initial gaussian 1D concentration profile:

$$c(x, o) = e^{-\frac{(x-x_0)^2}{2}}$$

with  $x=0.5$  and  $\sigma = 0.01$  diffusing in a box of length  $L$  according to:

$$\frac{\partial c(x, t)}{\partial t} = \nabla \cdot [D(x) \nabla c(x, t)] \quad (4.12)$$

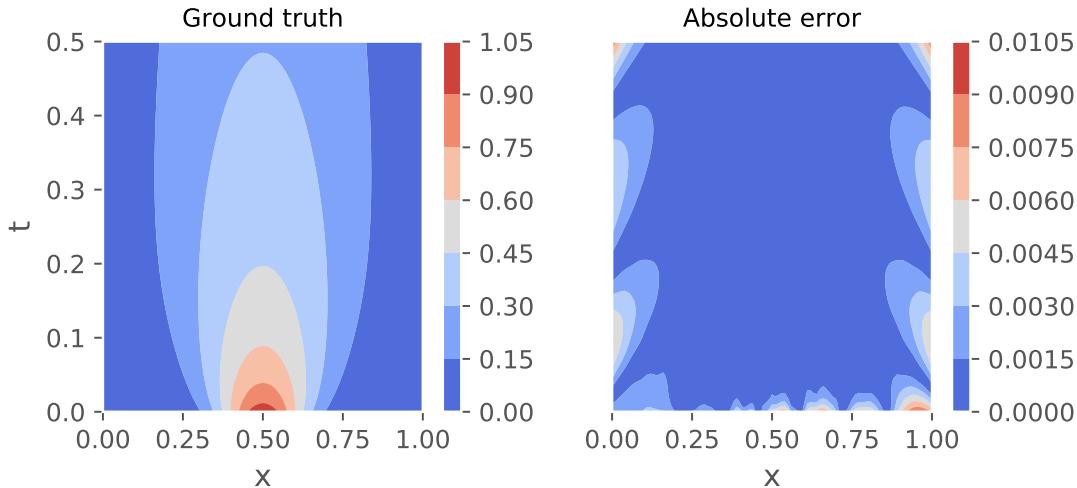
on the spatial domain  $[0, 1]$  with perfectly absorbing boundaries at the edges of the domain:

$$c(0, t) = c(1, t) = 0$$

If  $D(x) = D$ , this problem can be solved using a Greens function. Although being a simple problem, it contains all the essential features of a PINN. For the application of a PINN to more complex systems such as the Burgers, Schrodinger or Navier-Stokes equations, we refer the reader to the papers of M. Raissi et al (46, 31).

#### 4.2.2.1 CONSTANT DIFFUSION COEFFICIENT

We first numerically solve equation 4.12 with a diffusion coefficient of  $D(x) = D_0 = 0.1$  between  $t = 0$  and  $t = 0.5$ . Using a spatial and temporal resolution of 0.01, our total dataset consists of 5151 samples, while we have configured the neural network with 6 hidden layers of 20 neurons each and have set  $\lambda = 1$ . The left panel of figure 4.3 shows the ground truth (i.e. the numerical solution of equation 4.12) and the absolute error with respect to the groundtruth of the neural network output.

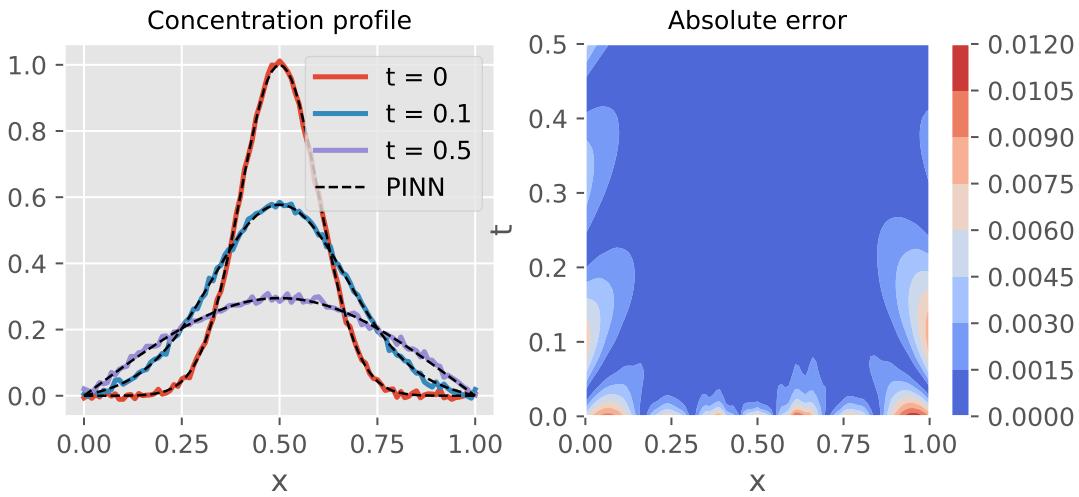


**Figure 4.3:** **Left panel:** Simulated ground truth of the problem. **Right panel:** The absolute error of neural network. Note that most of the error is located at areas with low concentration, i.e. signal.

The inferred diffusion coefficient is  $D_{pred} = 0.100026$ : an error of 0.026%. From the absolute error we observe that the error seems to localize in areas with low concentration. This is a feature we've consistently observed: in areas with low ‘signal’, the neural network struggles. Considering that in these areas there is simply not much data to learn from, this is not unexpected.

The input data of the previous problem is noiseless and thus of limited practical interest. We add 5% white noise to the data of the previous problem and train the network on this

noisy dataset. Note that the network is now doing two tasks in parallel: it's both denoising the data and performing a fit. In the left panel of figure 4.4 we show the concentration profile at times  $t = 0, 0.1$  and  $0.5$ , with the prediction of the PINN superimposed in black dashed lines at each time. On the right panel we show the absolute error with respect to the ground truth. Observe that the error again localizes in areas with low concentration. The inferred diffusion constant is  $D_o = 0.10052$ : an error of  $0.52\%$ . Although the error is an order of magnitude higher compared to the noiseless data, an error of less than  $1\%$  is extremely impressive.

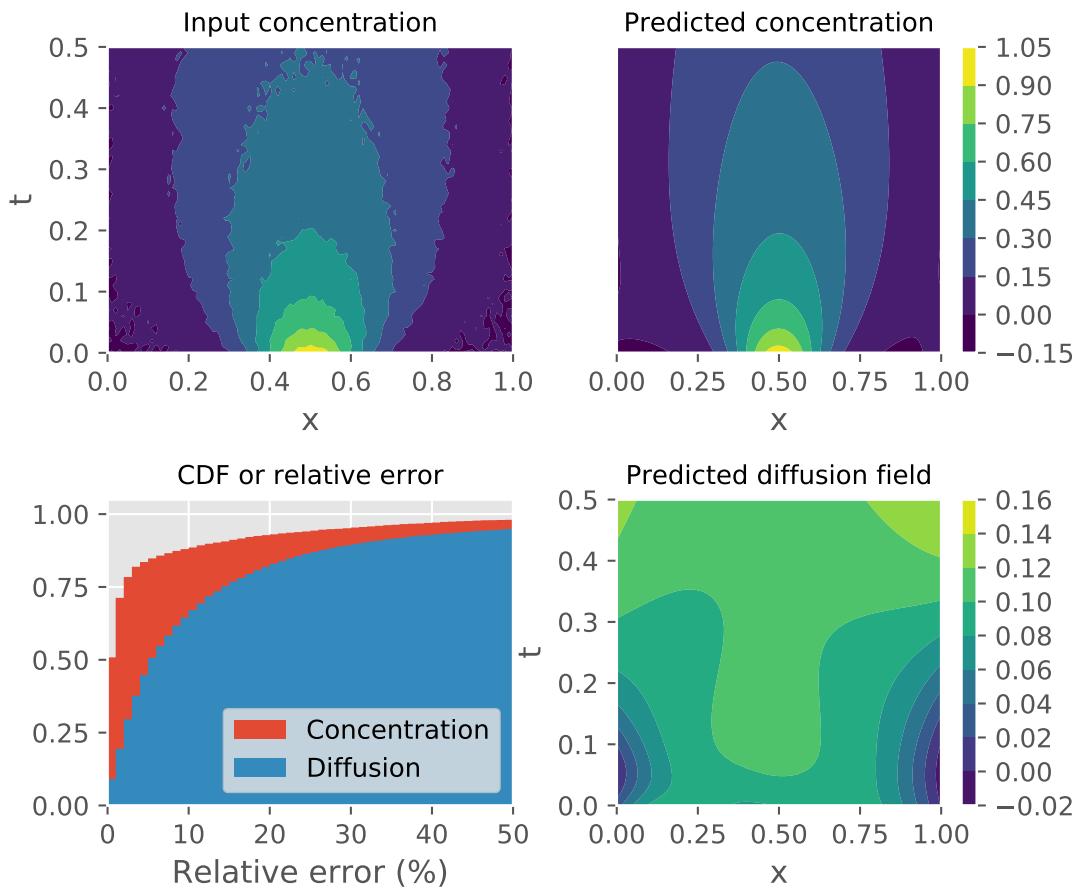


**Figure 4.4:** **Left panel:** The original noisy concentration profile at several times with the neural network inferred de-noised version superimposed. **Right panel:** The absolute error of neural network with respect to the ground truth. Note that most of the error is located at areas with low concentration.

#### 4.2.2.2 VARYING COEFFICIENTS

As stated, it should be possible to infer coefficient fields by using a two output neural network. We first test this on the noisy constant diffusion ( $D_o = 0.1$ ) dataset of the previous problem. In this case, while the neural network is allowed to assign a different diffusion constant to each point in the spatiotemporal domain, it should return  $D = 0.1$  for each. Figure 4.5 shows a summary of the results in four panels. In the upper left we show the data on which the network is trained, while the upper right panel shows the predicted concentration profile. Note the excellent match between the two. In the lower right panel we show the inferred diffusion field. We observe a good match in the middle of the plot, but the neural network again struggles in areas with low concentration, such as close to the edges of the

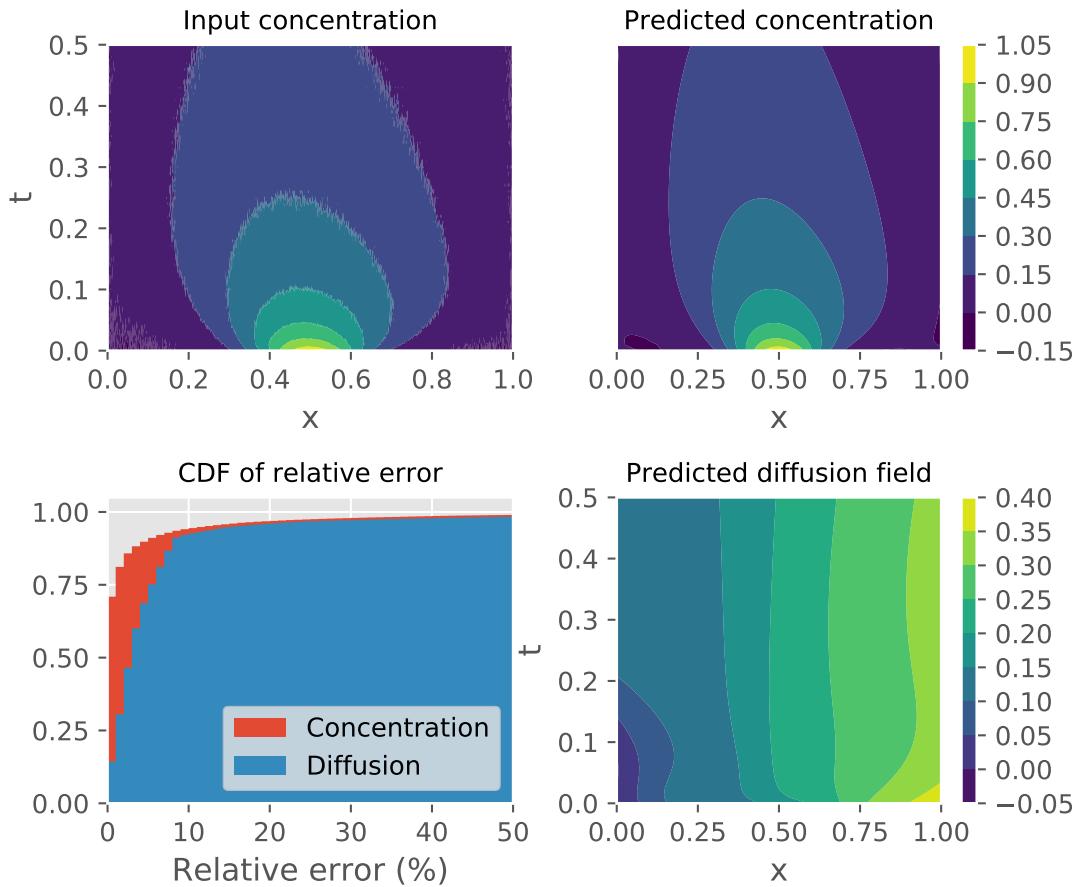
system. A more quantitative analysis of the predicted diffusion and concentration is presented in the lower left corner. Here we plot the Cumulative Distribution Function (CDF) of the absolute relative error of both the concentration and the diffusion constant. Note that the PINN predicts the concentration very well, with roughly 80% of the points having less than 5% error, but struggles more with the diffusion coefficient. Given that the diffusion coefficient is inferred self-consistently through its role in the physics-informed part of the cost function, this is not unexpected.



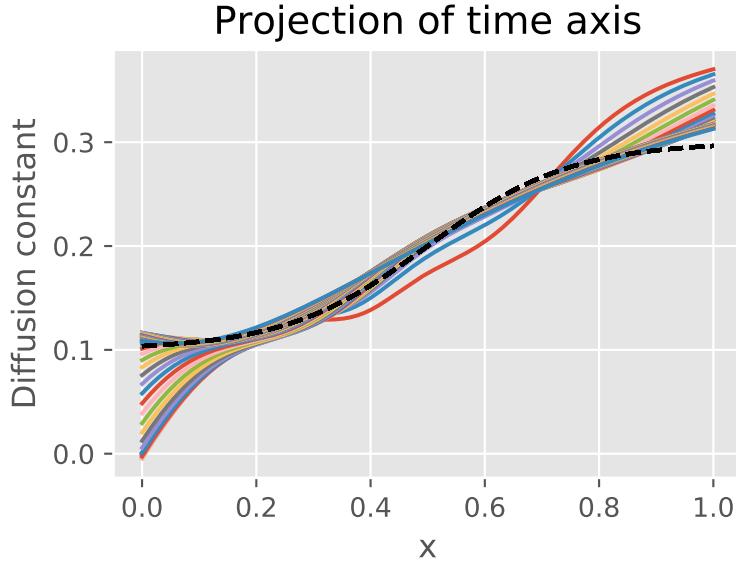
**Figure 4.5:** We show the training data and predicted concentration profile in the upper left and right panels. The lower right panel shows the inferred diffusion field while the lower left panel shows the CDF of the relative error of the diffusion and concentration.

In figure 4.6 we show a similar analysis for data with a non-constant diffusion field. Equation 4.12 has been numerically solved on a grid consisting of 50000 points and diffusion

constant profile  $D(x) = 0.2 + 0.1 \tanh(x)$ . Remarkably, the neural network is able to accurately infer the network with 85% of the diffusion field having an error of less than 10%. In figure 4.7 we show the inferred diffusion profiles in more detail by projecting them along the time axis. Observe that, yet again, the error is largest where the signal is lowest. Nonetheless, we've proven that a neural network is able to accurately infer a coefficient field from noisy data.



**Figure 4.6:** We show the training data and predicted concentration profile in the upper left and right panels. The lower right panel shows the inferred diffusion field while the lower left panel shows the CDF of the relative error of the diffusion and concentration.



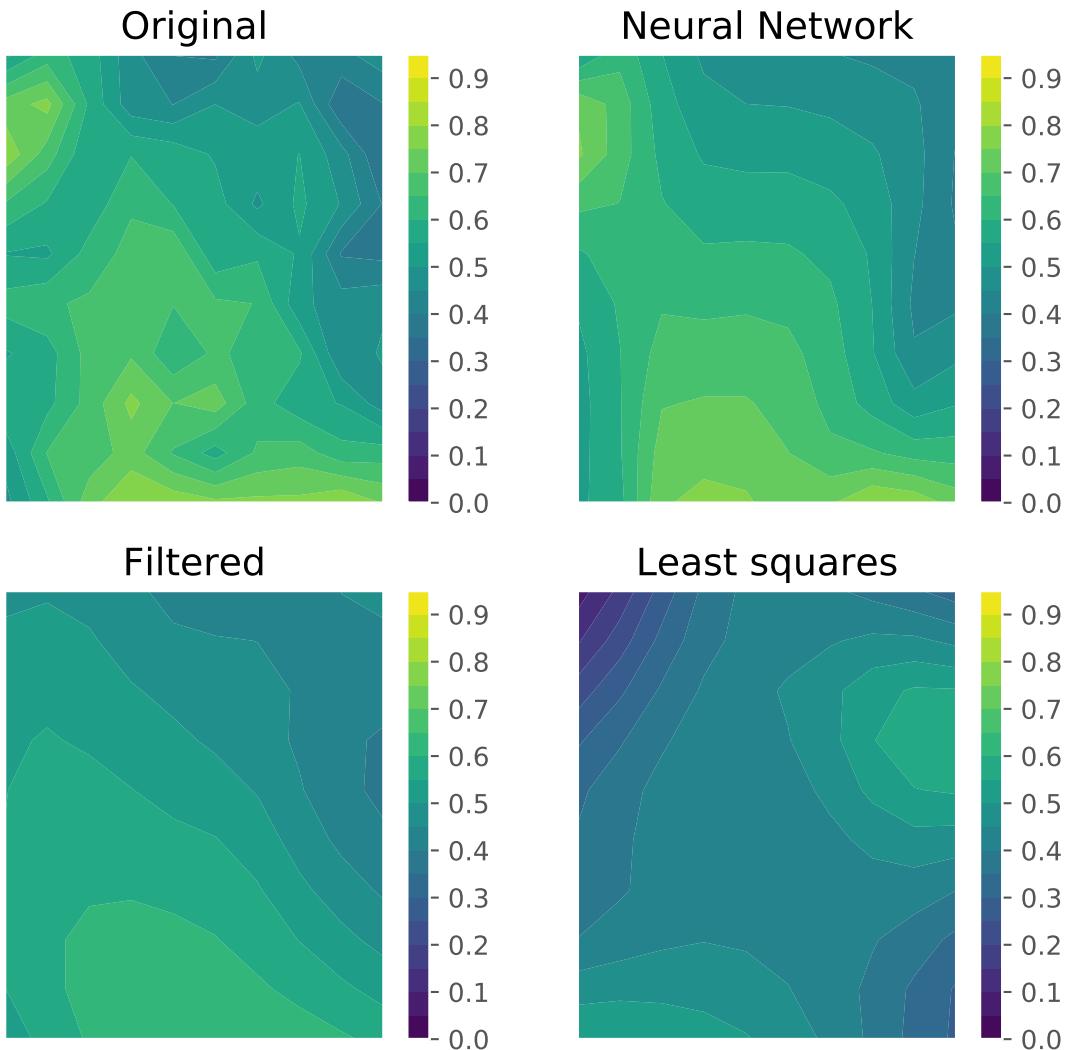
**Figure 4.7:** Projection of the inferred diffusion profile along the time axis.

#### 4.2.2.3 REAL CELL

We now apply the PINN technique to the RUSH data. Having observed in the previous section that the technique struggles in domains with low signal, we select a subset of the data consisting of 10 by 10 pixels during the first 30 frames, thus giving a dataset of 3000 points. We first fit this data assuming that it is described by a single diffusion coefficient and advection speed. The physics informed part of the cost function is thus:

$$g = o = -\partial_t c + D(\partial_{xx}c + \partial_{yy}c) - v_x \partial_x c - v_y \partial_y c$$

We train the network on the raw data: none of the filtering procedures presented in the model fitting chapter are used. The neural network gives the following results:  $D = -3 \cdot 10^{-6}$ ,  $v_x = 0.82$ ,  $v_y = 0.32$ , whereas the least-squares fitting gives  $D = 0.049$ ,  $v_x = -0.046$ ,  $v_y = 0.013$ . These results are completely different: the least squares predicts a diffusion constant and velocity roughly on the same order, whereas the neural network predicts a negligibly small diffusion constant. The direction and magnitude of the advection is different as well. To gain more insight into the fit, we study the concentration profiles of frame 5 as given by the original noisy signal, the output of the neural network, the filtered signal and the reconstructed signal of the least-squares fit. The signal is reconstructed by propagating the first frame using the calculated time derivative with the optimal fit parameters. The result is shown in figure 4.8.



**Figure 4.8:** In the upper right panel we show the unfiltered data of our subset for frame 5. The upper right panel shows the inferred profile by the PINN, while the lower two panels show respectively the filtered data and the data reconstructed from the least squares fit.

As can be observed from figure 4.8, the inferred concentration profile by the neural network matches the raw data very well, while the least squares fit does not. Since we have taken a 10 by 10 pixel patch of the data, this a fairly small scale and seeing such a close resemblance to the raw data might mean our model is fitted to the noise instead of the signal. Later frames reveal an inferred concentration profile less like the raw data however. Conclusively, the neural network seems to outperform the least-squares method, but due to

the small scale of our data, no clear verdict can be rendered. Increasing the scale of our data however make a cost function with constant coefficients unlikely.

In the previous section we proved that PINNs are able to infer coefficient fields. We now try to infer the coefficient fields for our subset of data. In figure 4.9 we show the result for a single frame.

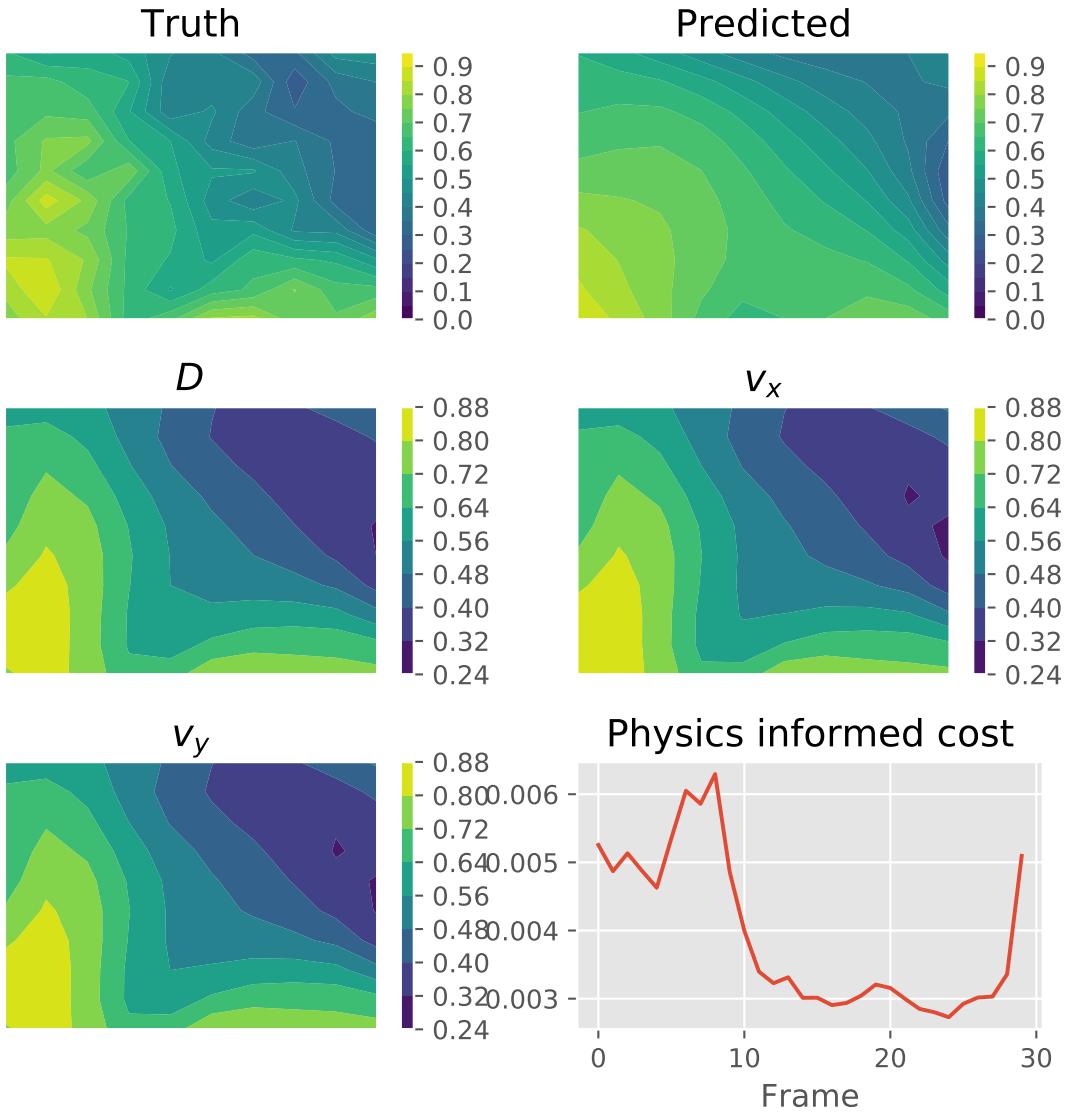


Figure 4.9: Caption.

In the six panels, we show respectively the raw data, the inferred concentration profile, the diffusion coefficient and advection and in the lower right corner the physics informed cost  $g$  per frame. Observe that the diffusion and advection profiles are exactly equal. Inspection of the concentration profile derives  $c_t, c_x, c_{yy} \dots$  shows no aberrant behaviour, implying that the neural network is functioning properly. These diffusion and advection profiles thus minimize the cost function but having similar coefficient values at each point is unlikely. Recall also that the diffusion coefficient obtained in the constant coefficient model above is orders of magnitude smaller than the advection. Inspection of the physics informed part of cost function shows that is on the order of  $10^{-3}$ . Although one to two orders of magnitude higher a cost for synthetic data, for real experimental data this does not seem suspiciously high. Concludingly, our results are clearly incorrect, but the neural network seems to perform properly.

# 5

## Golgi as a phase separated droplet

In this second part of the thesis we develop a model linking the Golgi function and size to the properties of the intracellular transport. We start with a general section on phase separation, followed by a section where we introduce an approximation known as the effective-droplet approximation. This approximation makes the phase separation analytically tractable. We then introduce our model and its biological justification. The results of our model will be presented in the next section.

### 5.1 PHASE SEPARATION

Consider a mixture of two molecules, type A and B, with underling interaction strengths  $\chi_{ij}$ . Depending on the strength and sign of these interactions, the system is either in a mixed state with a constant concentration of A and B, or in a phase-separated state. Landau showed that instead of a complete statistical description, phase separation could be modeled by a system with a double well free energy function<sup>47</sup>. We can define an *order parameter*  $c = N_A/N_B$  which describes the state of the system and define a free energy density function  $f(c)$  with minima at  $c_o^-$  and  $c_o^+$ :

$$f(c) = \frac{b}{2(\Delta c)^2} (c - c_o^-)^2 (c - c_o^+)^2$$

where  $b$  characterizes the strength of molecular interactions and  $\Delta c = |c_o^- - c_o^+|$ . Once the system phase separates, the system will have two areas of concentration  $c_o^+$  and  $c_o^-$  with

a boundary inbetween. Associated with this boundary is a surface tension, so that the full free energy of the system becomes

$$F(c) = \int dV(f(c) + \frac{1}{2}\kappa(\nabla c)^2)$$

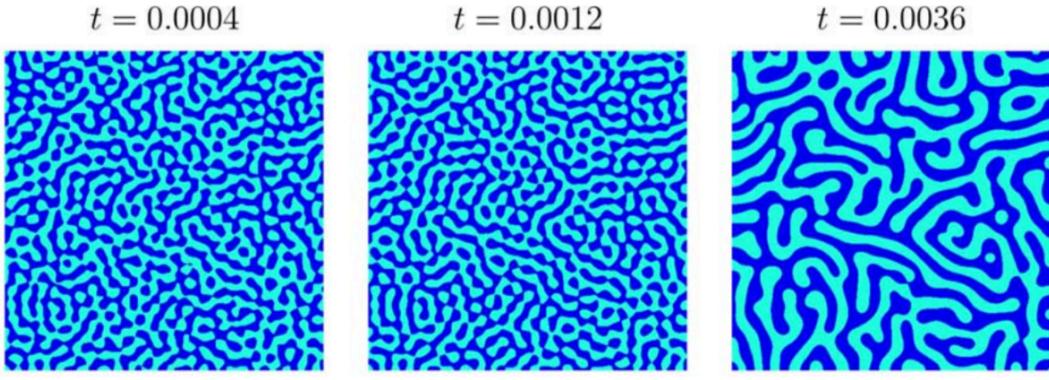
To find the equilibrium concentration profile, we minimize this free energy:

$$\frac{\delta F}{\delta c} = f'(c) - k\nabla^2 c = \mu(x) = 0 \quad (5.1)$$

where  $\delta F/\delta c$  is a functional derivative, as we minimize with respect to the concentration *profile*. Solving such an equation is generally not possible due to the third order terms of the free energy density function, but in 1D equation 5.1 has been solved to yield:

$$c(x) = \frac{c_o^- + c_o^+}{2} + \frac{c_o^+ - c_o^-}{2} \tanh\left(\frac{x}{w}\right) \quad (5.2)$$

where  $w = \sqrt{\kappa/b}$  is the width of the boundary. When we quench a system from a mixed state into a phase separated state (a process known as spinodal decomposition), the actual concentration profile is a far cry from equation 5.2. Inside the system, maze-like domains form and keep growing until a single dense and dilute phase are left. This is shown in figure 5.1.



**Figure 5.1:** Cahn-Hilliard domains

In this process, the dynamics need to be taken explicitly into account. In the case of liquid-liquid phase separation, the order parameter  $c$  is conserved, as a molecule of type A cannot

change into type B. This means that the order parameter can only exchange locally, so that:

$$\partial_t c = -\nabla \cdot \mathbf{j}$$

where  $\mathbf{j}$  is a flux. We can relate the flux to the chemical potential:

$$\mathbf{j} = -m\nabla\mu$$

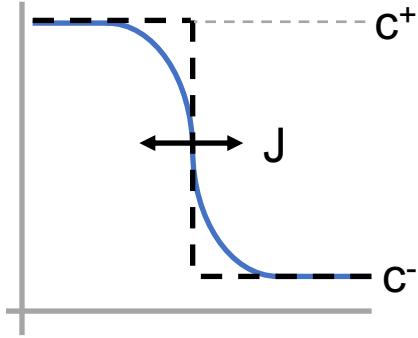
where  $m$  is a coefficient characterizing the mobility. Equation 5.1 also gives us an expression for the chemical potential, so that we finally obtain the *Cahn-Hilliard equation*:

$$\frac{\partial c}{\partial t} = m\nabla^2[f(c) - k\nabla^2c]$$

It is this equation which governs the behaviour observed in figure 5.1. Due to its non-linearity and fourth order derivatives simply solving the Cahn-Hilliard is usually forsaken in favour of deriving a scaling relation, which relates the domain growth speed  $dR/dt$  to the domain size  $R$  or some other system parameters. Another option is to study the system in the so-called effective droplet approximation, as we do in the next section.

## 5.2 EFFECTIVE DROPLET

Consider again equation 5.2. If  $w \ll 1$ , we can approximate the system by describing it as two bulk phases, separated by an interface. If we apply correct boundary conditions at the interface, we can essentially split the system into two separate problems for the dense and dilute phase and match them at the interface. We thus model the phase separated system as a droplet which exchanges material with its environment through its interface, as shown in figure 5.2



**Figure 5.2:** Model of an effective droplet. Blue line is full Cahn-Hilliard, black dashed line effective droplet.

By assuming the interface to be at thermodynamic equilibrium, the growth of the droplet is described in terms of the fluxes across the interface. Consider again equation 5.1. As we can neglect the interfacial term, we have  $\mu = f(c)$ . In each bulk phase, we linearize the chemical potential around its equilibrium density ( $c_o^-$  or  $c_o^+$ ) to yield:

$$\frac{\partial c}{\partial t} = D \nabla^2 c \quad (5.3)$$

where we've absorbed all the coefficients into a single coefficient  $D$ . Observe that we have obtained a diffusion equation. Since we now have a linear equation in  $c$ , we can analytically solve this. Moreover, it is possible to add additional effects such as decay, production or advection and still keep an analytically solvable model, as we show in the next section and chapter.

By introducing the interface and breaking the problem into two separate parts, we need two new boundary conditions to solve equations such as 5.3. By assuming the interface to be at thermodynamic equilibrium, we will derive a set of boundary conditions independent of the position, size or kinetic parameters of the droplet. Consider such a phase-separated system with an infinitely thin interface. The total free energy of the system can then be written as:

$$F = V_1 f(\varphi_1) + V_2 f(\varphi_2)$$

where  $V_i$  and  $\varphi_i$  are respectively the volume and concentration of phase  $i$  and  $f(\varphi_i)$  is the free energy density. Assuming incompressibility ( $V_1 + V_2 = V$ ) and conservation of particles ( $V_1 \varphi_1 + V_2 \varphi_2 = V \varphi$ ) constrains the system to two free variables, so that minimizing the free energy with respect to  $\varphi_1$  and  $V_1$  gives us two conditions:

$$f(\varphi_1) = f(\varphi_2)$$

$$0 = f(\varphi_1) + f(\varphi_2) + (\varphi_2 - \varphi_1)f'(\varphi_2)$$

Since  $f(\varphi) = \mu(\varphi)$ , the first condition states that both phases must have the same chemical potential, while the second one states that both phases must have equal pressure. The obvious solution to these equations is a completely mixed state with  $\varphi_1 = \varphi_2$ . A non-trivial phase-separated solution exists as well, where  $\varphi_1$  and  $\varphi_2$  are the two minima of the free energy density function  $f(\varphi)$ . Note that this is valid for droplets in 1D. In higher dimensions, the curvature of the droplet will affect the boundary conditions due to the Laplace pressure, but one can show that this leads to an extra term which scales with  $1/R^{48}$ .

Having defined boundary conditions, equations such as 5.3 can be solved. Although one could solve these equations fully time-dependent using Green's functions, we assume a quasi-steady state so that  $dc/dt = 0$ . This will give us the concentration profiles in and outside the droplet, droplet growth however is determined by the fluxes across the interface. We show this in the next section.

### 5.2.1 FLUXES, ACTIVITY AND INTERFACES

Given a concentration profile  $c(x)$ , a diffusive flux can be calculated by applying Ficks' second law:

$$J(x) = -D \frac{\partial c}{\partial x}$$

Using this expression, the flux at the interface on the inside and outside of the droplet,  $J_{in}$  and  $J_{out}$ , can be calculated. Note that in and out respectively refer to inside and outside of the droplet rather than the direction of the flux; our boundary conditions fix the concentration at the interface but not the (direction of the) fluxes. If these fluxes are not balanced, there exists a net flux across the interface which leads to either growth or decay of the droplet. As the droplet changes size, the interface moves with a speed  $v_n$ . We derive an expression for  $v_n$  in terms of the fluxes across it. To move the interface a distance  $\Delta x$ , a net material gain of  $\Delta x \Delta c$  is needed. This net gain is given by the net flux in a time  $\Delta t$ , so that:

$$\Delta x \Delta c = (J_{in} - J_{out}) \Delta t$$

which can be rewritten as:

$$\frac{\Delta x}{\Delta t} = v_n = \frac{J_{in} - J_{out}}{\Delta c} \quad (5.4)$$

In the passive case (and assuming quasi-steady state), the concentration both inside and outside the droplet would be described by a solution to laplace's equation (i.e.  $\nabla^2 c = 0$ ), leading to a flat concentration profile  $c_{out}(x) = c_o^-$ ,  $c_{in}(x) = c_o^+$  and hence  $J = 0$  everywhere; the system is at thermodynamical equilibrium with  $\mu = 0$  and the droplet doesn't change size as  $v_n = 0$ . Placing such a droplet in a supersaturated environment where  $c_{out}(x) > c_o^-$  would lead to a lead to a non-zero  $J_{out}$ , resulting in the droplet growing to infinity by a process known as Ostwald ripening. As we show in the next section, active droplet suppress the Ostwald ripening<sup>49</sup>, leading to a droplet with a finite radius.

We now make the droplet *active* by adding a chemical reaction in the droplet which decays the droplet material A into some other other material B. Assuming material B diffuses very fast and is thus always in equilibrium, we ignore material B and describe solely A. Adding a decay term to equation 5.3 gives:

$$D\nabla^2 c - kc = 0 \quad (5.5)$$

where  $k$  is a decay constant. Solving equation 5.5 will always give a convex solution and hence a finite  $J_{in}$ . A typical concentration profile for an active droplet is shown in figure 5.3.

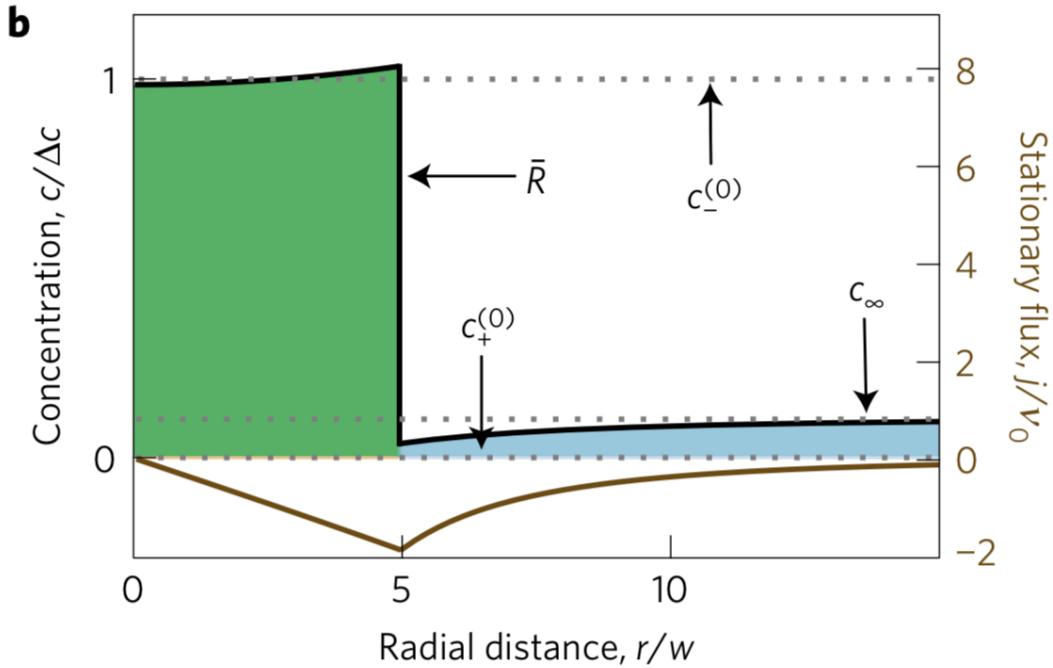


Figure 5.3: Typical concentration profile of active droplet. Taken from 21

The ‘decay flux’  $J_{in}$  will need to be balanced by a flux into the droplet  $J_{out}$  for a stable droplet to exist. This is a key property of active systems: while the system is at steady state (i.e.  $v_n = 0$ ), it is not at thermodynamical equilibrium, as  $\mu \neq 0$  and  $J \neq 0$ . Another fascinating property of active droplets is that they can propel themselves. To see this, consider a droplet of radius  $R$  at position  $x_o$  with two interfaces moving respectively at  $v_l$  and  $v_r$ . In a time  $dt$ , the droplet moves to a new position  $x_o + dx$  and will have a new radius  $R + dR$ :

$$x_o - R + v_l dt = x_o + dx - (R + dR)$$

$$x_o + R + v_r dt = x_o + dx + (R + dR)$$

Solving this set of equations for  $dx$  and  $dR$  gives:

$$\frac{dR}{dt} = \frac{1}{2\Delta c} (v_r - v_l) \quad (5.6)$$

$$\frac{dx_o}{dt} = \frac{1}{2\Delta c} (v_l + v_r) \quad (5.7)$$

Combining these equations with equation 5.4 finally relates the growth and movement of the droplet to the fluxes across the interface:

$$\frac{dR}{dt} = \frac{I}{2\Delta c} [(J_{in}^{x=R} - J_{in}^{x=-R}) + (J_{out}^{x=-R} - J_{out}^{x=R})]$$

$$\frac{dx_o}{dt} = \frac{I}{2\Delta c} [(J_{in}^{x=-R} + J_{in}^{x=R}) - (J_{out}^{x=-R} + J_{out}^{x=R})]$$

Studying equations 5.6 and 5.7 shows that movement only happens if the fluxes are asymmetrical; the droplet center is displaced because one side of the droplet grows faster than the other. This imbalance is caused by a concentration gradient and thus *droplets will move up the gradient*, as the flux on the high concentration side will be higher than on the low side.

These equations allow us to find steady states both with respect to the size and the position of the droplet. In the next section we adapt the equations to model the Golgi and couple it to the intracellular transport through the calculations of the fluxes  $J_{out}$ . We do so in the next section.

### 5.3 GOLGI AS AN ACTIVE DROPLET

In the introduction we justified using a phase-separation approach to describe the Golgi. In this section we develop our model for the Golgi from biological considerations, but having established the mathematical background of phase separation, we parallelly present the mathematical description.

We can recognize four different populations in our system: immature cargo -heading to the Golgi-, mature cargo, -originating from the Golgi and which is produced from immature cargo in the Golgi-, the Golgi itself and the cytoplasm, which acts as the solute. We start by reducing this set of populations to a system described by a single concentration  $c$ . Assuming maturation from the Golgi as a oneway process, i.e. immature cargo turns into mature cargo but not otherwise, and no interaction between the mature and immature cargo during intracellular transport, we can ignore the mature cargo. Modeling the solvent implicitly, the immature cargo in the cytoplasm is then represented by a dilute phase in some concentration  $c$ , while the Golgi is described by a dense phase in the same concentration.

Upon adding the drug nocadazole to mammalian cells, the microtubules are depolymerized and the Golgi ribbon breaks up into separate stacks<sup>50</sup>. These stacks are fully functional<sup>7</sup> and move away from their perinuclear location to collocate with an ERES. If we model not the complete Golgi but a single stack, we can reduce our problem to 1D, where a droplet

can move from one side of the system, representing the Golgi ribbon, to the other side, representing the ERES. As each stack is fully functional, we make no simplifications with respect to the function of the Golgi. Although many complex models of the maturation exist, we model it as a simple decay-like term:

$$\frac{\partial c}{\partial t} = -\nabla J - kc$$

$$J = -D\nabla c$$

where  $k$  is a maturation constant. We now turn our attention to the intracellular transport. In our model fitting chapter we presented an argument that we could model the intracellular transport as an advection-diffusion equation. Evidence exists of vesicles refusing with the ER<sup>8</sup>, so we add an additional decay term, so that the concentration outside the droplet is described is described by:

$$D\partial_x^2 c(x) - v\partial_x c(x) - \alpha c(x) = 0 \quad (5.8)$$

with  $v$  an advection velocity and  $\alpha$  some decay constant. Solving this equation will lead to a concentration profile with a gradient and we thus model our Golgi as an active droplet growing in a concentration gradient, inspired by 51. A study of the biogenesis of the Golgi<sup>8</sup> shows that stacks are transported to the ribbon over the microtubules, so we thus add the advection also to the dense phase:

$$D\partial_x^2 c(x) - v\partial_x c(x) - kc(x) = 0 \quad (5.9)$$

Note that the dense and dilute phase description are thus almost similar, save for a different decay (maturation) constant. One could pick different diffusion constants and advection speeds, but for simplicity and without loss of generality we pick the same. As our free energy function has minima at  $c_o^+$  and  $c_o^-$ , our boundary conditions at the interface are:

$$c(x_o \pm R) = \begin{cases} c_o^+, & \text{inside} \\ c_o^-, & \text{outside} \end{cases}$$

As stated, we model our system in 1D, with one boundary representing the ERES and the other boundary as the location of Golgi Ribbon. We place the ERES on the left side of the system and thus model this boundary as source:

$$(-D\partial_x c + vc)|_{x=o} = J_{in}$$

whereas the right boundary is merely the edge of the system and we thus model it as a zero-flux boundary:

$$(-D\partial_x c + vc)|_{x=L} = 0$$

We solve this set of equations in the next chapter.

# 6

## Results model

The previous chapter introduced phase separation and our model for the Golgi as a phase-separated droplet. In this chapter we study the behaviour of our model. In the first section, we analytically solve the model for a free droplet, i.e. a droplet free to move throughout the system. Using these expressions, we investigate the effect of advection on an active droplet and study the steady states of our model. Considering the biology, the diffusion constant  $D$  and the decay rates  $k$  and  $\alpha$  will most likely be system parameters and thus fixed. On the other hand, the advection speed  $v$  encompasses the active transport across the microtubules and could easily vary, depending on the amount of molecular motors available and the rate at which they use ATP; the influx  $J_{in}$  is dependent on the activity of the ER and would probably vary too. We are thus interested in creating a phase diagram with the stable radius and position as a function of  $J_{in}$  and  $v$ .

The Golgi stack is either located at the ribbon or at the ERES. In the second section we investigate this using a droplet stuck to the edge of the system. Taking a broader view, we study when phase separation takes place and if an effective droplet exists when phase separation should take place. We also validate the effective droplet model by checking mass conservation and this section with a numerical investigation. The chapter ends with a short section discussing our conclusions and possible biological connections.

### 6.1 EFFECTIVE DROPLET

In this section we derive analytical expressions for the fluxes across the interface of the droplet. We present the most general case, including advection, decay and maturation,

and derive simplified expressions later. Both the dense and dilute phase are described by an advection-diffusion-decay equation, which has a general solution given by

$$c(x) = C_1 e^{-\frac{x}{l^-}} + C_2 e^{\frac{x}{l^+}}. \quad (6.1)$$

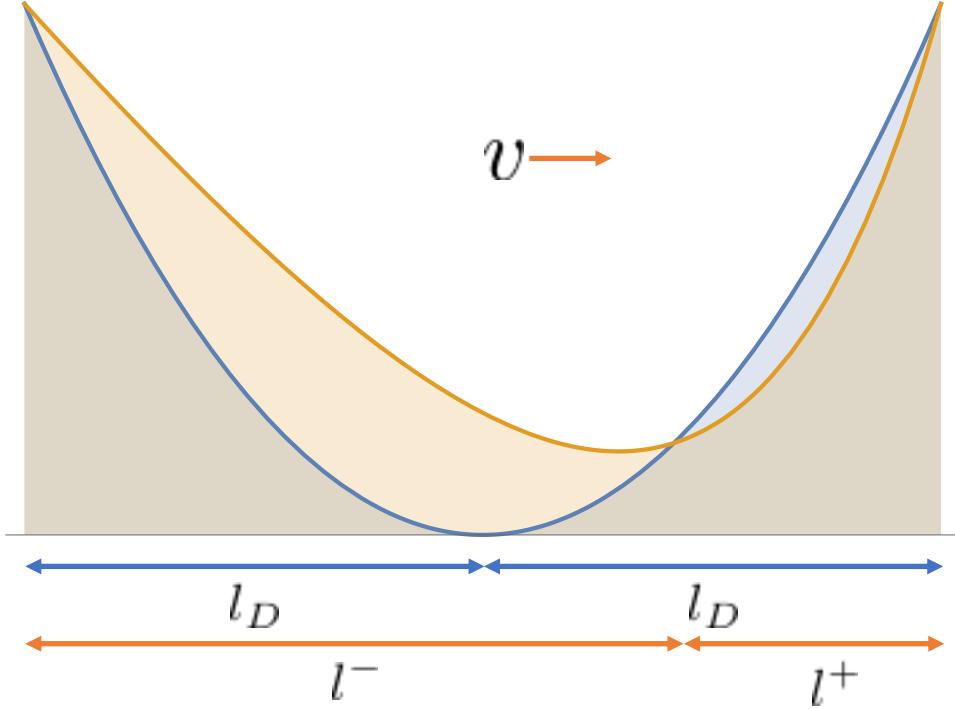
We have defined a lengthscale  $l^\pm$  as

$$l^\pm = \frac{2D}{\sqrt{4kD + v^2} \pm v}, \quad (6.2)$$

where the maturation rate  $k$  should be replaced by the decay rate  $\alpha$  in the dilute phase. Note it is a combination of a lengthscale set by the diffusion  $l_D = \sqrt{D/k}$  and a lengthscale set by the advection  $l_v = 2D/v$ :

$$\frac{l}{l^\pm} = \sqrt{\frac{k}{D} + \left(\frac{v}{2D}\right)^2} \pm \frac{v}{2D} = \sqrt{\frac{l}{l_D^2} + \frac{l}{l_v^2}} \pm \frac{l}{l_v}.$$

We have defined symmetric boundary conditions for the droplet,  $c(R) = c(-R) = c_o^+$ . Solving equation 6.1 with these boundary conditions will lead to a convex concentration profile. We can thus associate  $l^-$  with  $x < 0$ , the left side of the droplet, and  $l^+$  with  $x > 0$ , the right side. In a system without advection we have  $l^+ = l^- = l_D$  and the concentration profile will thus be symmetric around  $c(o)$ . If  $v > 0$  however, we have  $l^- > l^+$  and the droplet is no longer symmetric around  $c(o)$ ; rather, the position of the minimum concentration moves right, while the minimum concentration itself increases. This is shown in figure 6.1, where we have plotted a concentration profile for  $v = 0$  in blue and  $v > 0$  in orange.



**Figure 6.1:** Concentration profiles inside an active droplet for  $v=0$  (blue) and  $v>0$  (orange). Note that the minimum concentration increases and that its location moves right.

For a diffusive-advection flow, the flux is determined by  $J(x) = -D\partial_x c(x) + vc(x)$  and applying this to the droplet concentration yields the fluxes. The fluxes themselves are not particularly insightful, but considering equations 5.6 and 5.7, we can define a *maturity flux*  $J_{mat} = J_{in}^{x=R} - J_{in}^{x=-R}$  and a *positional flux*  $J_{pos} = J_{in}^{x=R} + J_{in}^{x=-R}$ , so that

$$\frac{dR}{dt} = \frac{I}{2\Delta c} [J_{mat} + (J_{out}^{x=-R} - J_{out}^{x=R})] \quad (6.3)$$

$$\frac{dx_o}{dt} = \frac{I}{2\Delta c} [J_{pos} - (J_{out}^{x=-R} + J_{out}^{x=R})] \quad (6.4)$$

The maturity flux  $J_{mat}$  is the flux at the interface due to the maturation in the droplet. Note it is solely determined by the diffusive flux, as

$$J_{in}^{x=R} - J_{in}^{x=-R} = (-D\partial_x c(x) + vc(x))|_{x=R} - (-D\partial_x c(x) + vc(x))|_{x=-R} = D(\partial_x c(x)|_{x=-R} - \partial_x c(x)|_{x=R}).$$

The maturation flux does have a dependence on the advection through its effect on the concentration profile. Since  $J_{rad}$  is solely determined by the diffusive flux and the solutions of 6.1 are convex, the fluxes at the two interfaces have opposite signs. More so,  $J_{in}^{x=R} < 0$  and  $J_{in}^{x=-R} > 0$ , so that  $J_{mat} < 0$ . This means that the droplet will shrink unless sustained by some influx from outside the droplet, as can be seen from equation 6.3. If the maturation flux is exactly balanced by this influx, the droplet radius remains stable. Whereas passive droplets will grow to an infinite radius, active droplets remain at a finite radius due to their suppression of the Ostwald Ripening<sup>49</sup>. For our particular choice of boundary conditions, we have derived for  $J_{mat}$ :

$$J_{mat} = \frac{-2c_o^+ D}{l} \frac{\sinh \frac{R}{l^-} \sinh \frac{R}{l^+}}{\sinh \frac{R}{l}}, \quad (6.5)$$

where we have defined an ‘effective lengthscale’  $l$  as

$$l = \frac{l^+ l^-}{l^+ + l^-}.$$

For a small, non-adverted droplet,  $l^- = l^+ = l_D$ ,  $l = l_D/2$  and  $R \ll l_D$ , we can approximate the maturation flux as

$$J_{mat} = -2c_o^+ kR. \quad (6.6)$$

Effectively, we have approximated the concentration profile inside the droplet as  $c(x) = c_o^+$ , so that the flux lost due to decay with rate  $-k$  for a droplet with size  $2R$  indeed gives equation 6.6. One would thus expect that the limit of  $R \rightarrow \infty$  would yield an infinite flux but taking the limit of 6.5 gives

$$\lim_{R \rightarrow \infty} = -2c_o^+ \sqrt{kD},$$

which does not yield the shocking result that  $\infty = \sqrt{D/k}$ , but that the flux saturates for  $R > \sqrt{D/k} = l_D$ . When  $R \gg l_D$ , the concentration in the middle of the droplet drops to zero and since the maturation scales with the concentration, the flux saturates. In this regime, the effective droplet theory is not valid and hence we require that  $R < l_D$ . In the case of an advected droplet this is a more subtle point, as advection increases the minimum concentration inside the droplet (as can be seen in figure 6.1). We study this numerically in the next section.

The positional flux  $J_{pos}$  is the internal flux which leads to droplet movement. For our set of boundary conditions, we have derived

$$J_{pos} = 2c_o^{in} D \left[ \frac{Pe_-}{l_-} \frac{\sinh \frac{R}{l_+} \cosh \frac{R}{l_-}}{\sinh \frac{R}{l}} - \frac{Pe_+}{l_+} \frac{\sinh \frac{R}{l_-} \cosh \frac{R}{l_+}}{\sinh \frac{R}{l}} \right], \quad (6.7)$$

where we have defined the Peclet-like numbers

$$Pe^\pm = 1 \mp \frac{vl^\pm}{D}$$

In a passive droplet  $c(x) = c_o^+$  so that the positional flux equals  $2c_o^+ v$ , but in an active droplet we need to take into account the internal diffusion. Recall that the diffusive fluxes point inwards and hence are aligned antiparallelly, whereas the advective fluxes are aligned parallel. The net flux at the two interfaces is thus different, leading to equation 6.7 instead of  $2c_o^+ v$ .

We now turn to the fluxes on the outside of the droplet. A droplet of radius  $R$  at position  $x_o$  has its interfaces at  $x_o \pm R$  and defining  $x_1 = x_o - R$  and  $x_2 = x_o + R$  we have derived the following expressions for the flux at the interfaces

$$J_{out}^{x=-R} = J_{in} \frac{(1 + \frac{l_-}{l_+}) e^{\frac{-x_1}{l_-}}}{Pe_- + Pe_+ \frac{l_-}{l_+} e^{\frac{-x_1}{l_-}}} + \frac{c_o^{out} D}{l_-} \frac{Pe_+ (1 - e^{\frac{-x_1}{l_-}})}{\frac{l_+}{l_-} + \frac{Pe_+}{Pe_-} e^{\frac{-x_1}{l_-}}} \quad (6.8)$$

$$J_{out}^{x=R} = -c_o^{out} D \frac{Pe_- Pe_+ (1 - e^{\frac{-x_2+L}{l_-}})}{l_+ Pe_- + e^{\frac{-x_2+L}{l_-}} l_- Pe_+} \quad (6.9)$$

Although not particularly enlightening, we note the similarity between the second term of 6.8 and 6.9. The flux on the left of the droplet has another term in  $J_{in}$ , accounting for the source we have placed at the left boundary. In the next section we study the phase diagram equations 6.5, 6.7, 6.8 and 6.9 give rise to.

## 6.2 FREE DROPLET

We now wish to study the phase diagrams of free droplets and their steady states. More specifically, we wish to investigate when droplets have a stable state (i.e.  $dR/dt = dx_o/dt = 0$ ) at some position  $x^*$  in the system. The first configuration we study ignores the decay outside the droplet, i.e.  $\alpha = 0$ . In this case, the outside fluxes become constant and independent of the location of the droplet, as the only way for the cargo to ‘exit’ the system is to mature in the droplet:

$$J_{out}^{x=-R} = J_{in}$$

$$J_{out}^{x=R} = 0$$

The flux on the right interface of the droplet is zero as there is no source nor decay and in our quasi-steady state approximation the flux then must become zero. The equations for the flux in the droplet remain unchanged as they are independent of the transport parameters. Developing the internal droplet fluxes for  $R \ll l^\pm$  gives:

$$J_{rad} \approx -2c_o^+ kR \quad (6.10)$$

$$J_{pos} \approx 2c_o^+ v \quad (6.11)$$

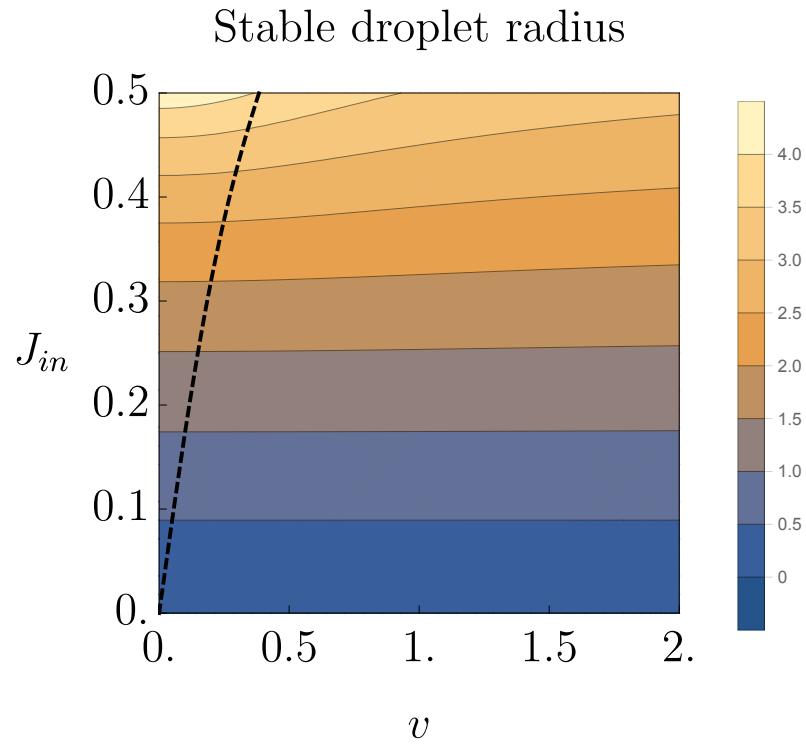
Putting these expressions in equations 6.3 and 6.4 gives

$$\frac{dR}{dt} \approx \frac{1}{2\Delta c} (J_{in} - 2c_o^+ kR), \quad (6.12)$$

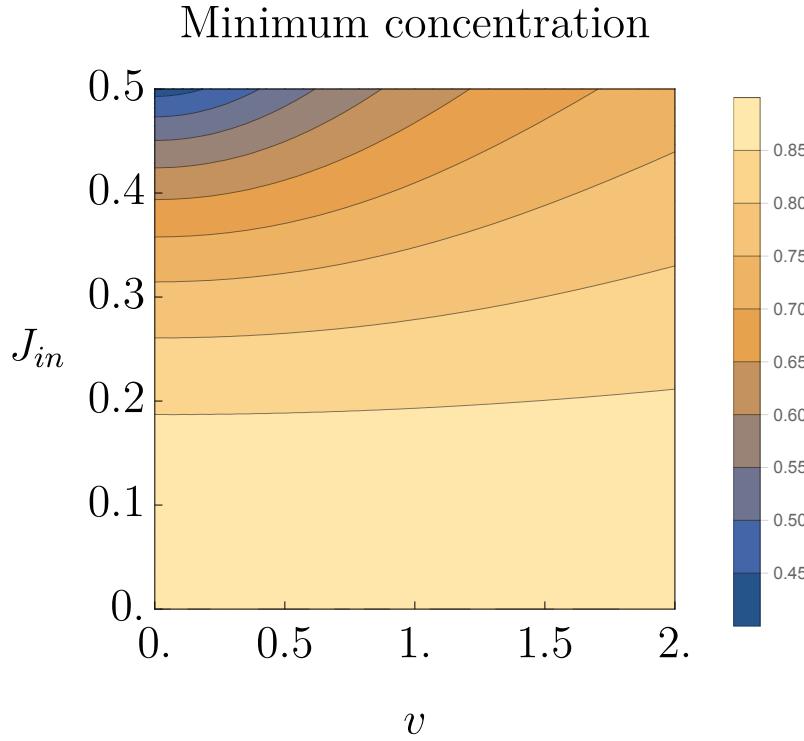
$$\frac{dx_o}{dt} \approx \frac{1}{2\Delta c} (2c_o^+ v - J_{in}).$$

The stable radius,  $R_{stable} \approx J_{in}/2c_o^+ k$  is thus independent of the velocity  $v$  and the droplet will maintain its position if  $v_{stable} \approx J_{in}/2c_o^+$ . This means that, save for  $v_{stable}$ , the droplet will always move either right or left and that the movement direction switches at the switching velocity  $v_{stable}$ . It is non-zero due to the self-movement of an active droplet; recall that an active droplet will move itself up a concentration gradient. The advection needs to compensate for this movement, giving rise a non-zero  $v_{stable}$ .

We now study this system numerically. As the fluxes on the outside of the droplet are independent of the velocity, we are in fact studying the effect of advection on an active droplet, irrespective of its environment. We plot the stable radius of the droplet in figure 6.2 and the corresponding minimum concentration in 6.3. We have used the following parameters:  $D = 1$ ,  $k = 0.1$ ,  $c_o^+ = 0.9$ .



**Figure 6.2:** The stable radius as a function of the velocity  $v$  and influx  $J_{in}$ . The dashed line is the line  $dx_o/dt = 0$ .



**Figure 6.3:** The minimum concentration in a stable droplet as a function of the velocity  $v$  and influx  $J_{in}$ . In areas with a low concentration the effective droplet model is not valid.

Note that the stable radius of small droplets is independent of the velocity, but that we do observe some dependence for bigger droplets. However, concurrently with the size increase is the minimum concentration decrease, as shown in figure ?? . For very low  $v$  and high  $J_{in}$ , the concentration even drops to 0.4 - a concentration corresponding to the dilute well of the free energy and thus clearly unphysical. Increasing  $v$  raises the minimum concentration, while also slightly decreasing the radius of the droplet. To understand this decrease in radius, consider again figure 6.1. Calculating some average concentration  $\bar{c} = \frac{1}{V} \int c(x)dV$ , it is clearly visible that this is higher for the advected droplet. Estimating the maturation flux as  $J_{mat} \propto -2Rk\bar{c}$ , an advected droplet thus has a higher maturation flux than a non-adverted droplet. The maturation flux needs to be balanced by the influx  $J_{in}$  for a stable droplet so that

$$R = \frac{J_{in}}{2k\bar{c}}. \quad (6.13)$$

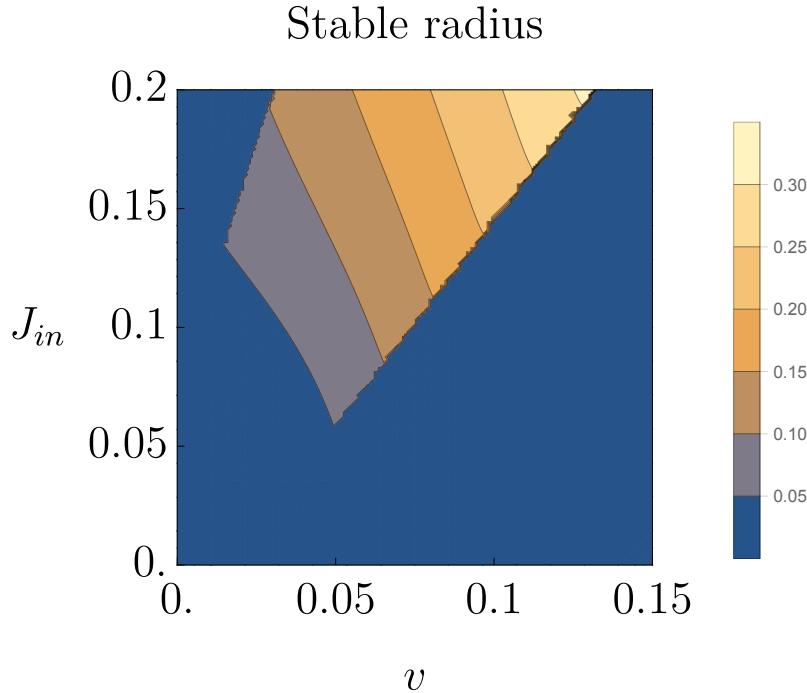
Since both  $J_{in}$  and  $k$  are fixed,  $R$  must decrease and thus advection compacts active droplets. The superimposed dashed line in figure 6.2 corresponds to  $dx_o/dt = 0$  and thus represents

the stable droplets for which  $dR/dt = dx_o/dt = 0$ . Observe that for small  $v$  it indeed shows a linear dependence between  $J_{in}$  and  $v$  as predicted, but that for higher  $v$  we do observe some non-linearity. Due to the low concentrations those areas are unphysical however. We now investigate the stability of this line by perturbing 6.12 around  $R_{stable}$ . We obtain

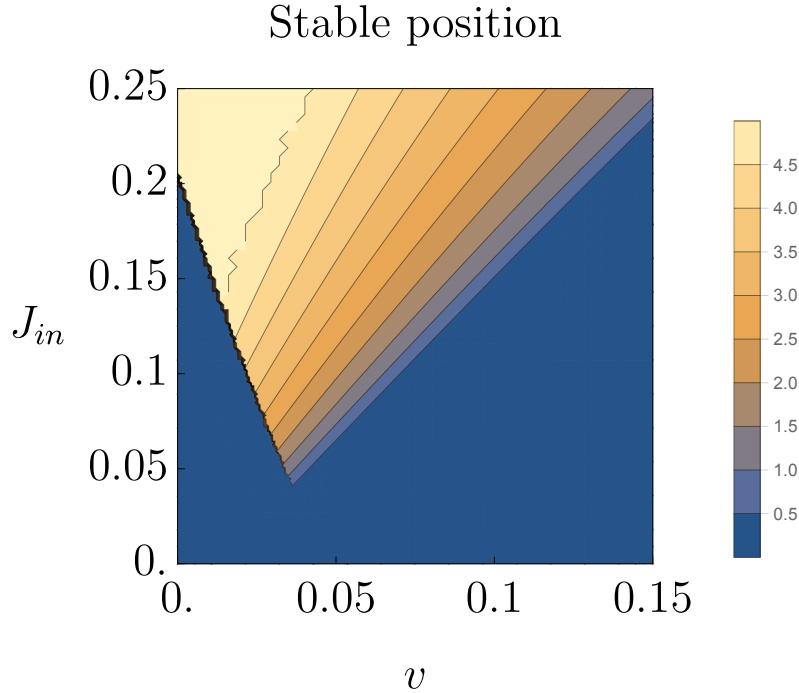
$$\frac{d\delta r}{dt} = -2c_o^+ k \delta r.$$

Since both  $k > 0$  and  $c_o^+ > 0$ , any fluctuations cancel; the steady state is stable. The system we have studied so far is completely independent of the position in the system as the outside fluxes are constant. By including decay outside the droplet, i.e.  $\alpha \neq 0$ , the outside fluxes will become dependent on the position of the droplet.

We solve equations 6.5, 6.7, 6.8 and 6.9 numerically by finding the  $x_o^*$  and  $R^*$  for which  $dx_o/dt = dR/dt = 0$  inside our system, i.e.  $0 < x_o^* < L$ ,  $0 < R^* < L/2$ . Using  $k = 0.3$ ,  $\alpha = 0.1$ ,  $D = 1$ ,  $c_o^- = 0.1$ ,  $c_o^+ = 0.9$ ,  $L = 5$ , we plot the steady state radii and positions in figures 6.4 and 6.5.



**Figure 6.4:** Steady state radius as a function of  $v$  and  $J_{in}$  made using  $k = 0.3$ ,  $\alpha = 0.1$ ,  $D = 1$ ,  $c_o^- = 0.1$ ,  $c_o^+ = 0.9$ ,  $L = 5$ . Blue areas correspond to no droplet.



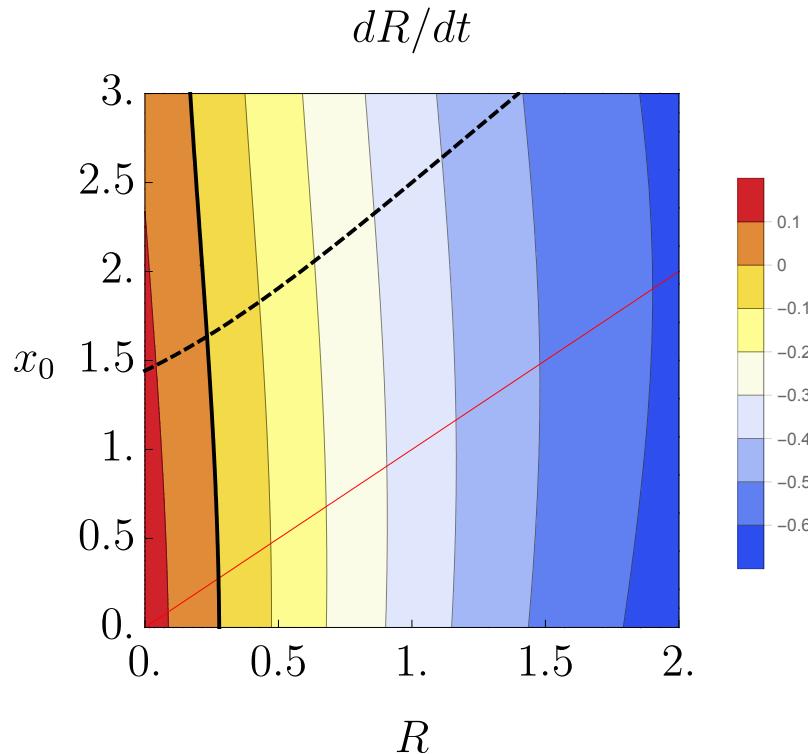
**Figure 6.5:** Steady state position as a function of  $v$  and  $J_{in}$  made using  $k = 0.3$ ,  $\alpha = 0.1$ ,  $D = 1$ ,  $c_o^- = 0.1$ ,  $c_o^+ = 0.9$ ,  $L = 5$ . Blue areas correspond to no droplet.

In areas which are blue in both plots no droplet exists in the system. We identify two causes, each connected to a corresponding ‘cutoff line’ in the stable position plot. First, by adding decay, we have added another ‘exit’ for the contents of the system. Thus, for low  $J_{in}$  and  $v$  a droplet won’t exist. This explains the lower left cutoff and is supported by the fact that this edge corresponds to the line  $R = 0$ . The other edge has  $x_o = 0$ , meaning that the droplet moved past the edge of the system. In the radius plot we also observe a third cutoff in the upper left corner. This corresponds to the  $x_o = 5$  edge and represents a droplet at the far end of the system. To satisfy the no-flux boundary condition, the droplets’ radius must go to zero. Hence this area is shaded blue in the radius plot, but not in the position plot.

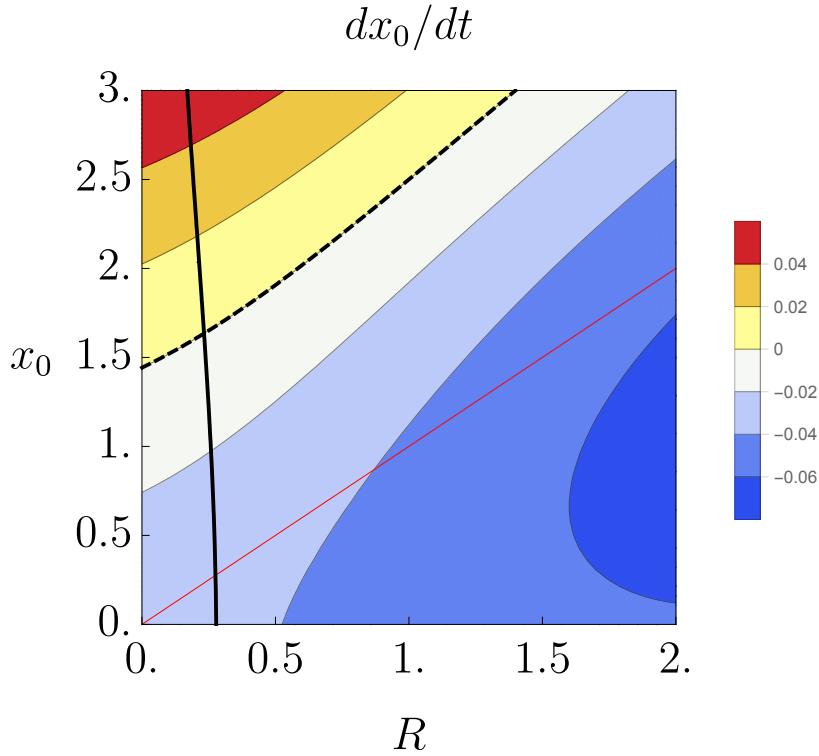
Note that advection increases the droplet radius, contrary to the no-decay case. Recall that advection decreased the radius because the outside fluxes were constant. Having added decay to the system, this is not the case anymore. For a droplet at a fixed point  $x_o$ , increasing  $v$  increases the outside flux as less is lost to decay. Although increasing  $v$  also increases the maturation flux inside the droplet, the increase of the outside flux is dominant and hence the droplet radius increases with increasing  $v$ . Also observe in figure 6.5 that increasing  $v$  de-

creases  $x_o$ . Increasing the flow thus leads to the droplet moving further up that flow, a very counterintuitive situation. To see why this happens, consider a droplet of fixed radius  $R$  at position  $x_o$ . Remembering equation 6.13, increasing  $v$  increases  $\bar{c}$ , which can only be compensated by a higher influx  $J_{in}$ . In a system with decay and advection, the influx is higher upstream and hence the droplet moves upstream.

We study the stability of these steady states by plotting  $dx_o/dt$  and  $dR/dt$  at  $J_{in} = 0.18$  and  $v = 0.1$  in figures 6.6 and 6.7.



**Figure 6.6:**  $dR/dt$  as a function of  $x_o$  and  $R$ . The solid black line denotes the  $dR/dt = 0$  and the dashed line  $dx_o/dt = 0$ . The red line is the line  $x_o = R$ .



**Figure 6.7:**  $dx_o/dt$  as a function of  $x_o$  and  $R$ . The solid black line denotes the  $dR/dt = 0$  and the dashed line  $dx_o/dt = 0$ . The red line is the line  $x_o = R$ .

The solid black lines denote  $dR/dt = 0$  and the dashed lines  $dx_o/dt = 0$ . The red line is the line  $x_o = R$ ; a steady state needs to be above this line, as below this line  $x_o - R < 0$ , meaning that the droplets' left interface is outside of the system. We observe that the line  $dR/dt = 0$  is stable, but as  $dx_o/dt = 0$  is not, the steady state is unstable. This plot is typical for all parameters, so we conclude that all steady states are unstable: the droplet either moves left or right until it hits the edges of the system. The free droplet model does not properly describe this situation, as it always has two interfaces. In reality, when the droplet hits the edges of the system one of the two interfaces disappears and the droplet becomes like a wetting layer. We investigate this in the next section.

### 6.3 DROPLET STUCK TO WALLS

In the previous section we showed that a droplet will always move left or right until it hits the edges of the system, but that this situation is not properly described by our free droplet

model. In this section we present a slightly modified model to account for this situation. Once the droplets' interface hits the edge of the system, it ceases to be an interface between a dense and a dilute phase: rather, the boundary condition of the droplet must become the boundary condition of the system. For a droplet on the left of the system we thus have the boundary conditions  $c(R) = c_o^+$  and  $J(0) = J_{in}$ , while for the droplet on the left we have  $c(L - R) = c_o^-$  and  $J(L) = 0$ . We present the behaviour of this modified model in this section, taking a slightly wider view than before. Instead of assuming the existence of a droplet, we first investigate when droplets phase separate at the edges of the system. We will then prove the existence of a stable effective droplet when such a phase separation should take place and that mass is conserved. Finally, we present a phase diagram and discuss the biological connection and implications.

### 6.3.1 OCCURENCE OF PHASE SEPARATION

Consider again the double well free energy with minima at  $c_o^-$  and  $c_o^+$ :

$$f(c) = \frac{b}{2\Delta c^2} (c - c_o^-)^2 (c - c_o^+)^2 \quad (6.14)$$

This free energy describes a system phase separating into a dense area with concentration  $c_o^+$  and a dilute area of concentration  $c_o^-$ . Our system is open

We thus ask at which concentration the dilute and dense phases becomes unstable. For the thermodynamics we know that stability requires  $d^2f/dc^2 > 0$ . In areas where  $d^2f/dc^2 < 0$ , fluctuations keep growing and the system will phase separate. For the free energy density given by 6.14, the area between the two inflection points  $c = (c_o^+ + c_o^-)/2 \pm \sqrt{3}(c_o^+ - c_o^-)$  is unstable. Thus, if the concentration in the dense phase becomes lower than upper inflection point, a dilute phase will form. On the other hand, if the dilute phase reaches the lower inflection point, a droplet will form. Note that we have neglected the gradient term. We thus have a criterium for when a droplet is formed.

To study when this minimum concentration is reached, we consider our system without a droplet which is described by a single advection-diffusion-decay equation with boundary conditions  $J(0) = J_{in}$  and  $J(L) = 0$ . Resulting concentration profiles will have the highest concentrations at the edges and we thus calculate at which  $J_{in}$  the concentration reaches the lower inflection point. We obtain:

$$\text{Left: } J_{in} = -\frac{2al((1 - \frac{1}{\sqrt{3}}c_o^+) + (1 + \frac{1}{\sqrt{3}}c_o^-))}{\frac{vl}{D} - \coth(\frac{L}{2l})} \quad (6.15)$$

$$\text{Right: } J_{in} = al \left( \left( 1 - \frac{I}{\sqrt{3}} c_o^+ \right) + \left( 1 + \frac{I}{\sqrt{3}} c_o^- \right) \right) \left( e^{-\frac{L}{2l}(\frac{vl}{D}-1)} - e^{-\frac{L}{2l}(\frac{vl}{D}+1)} \right), \quad (6.16)$$

where  $l$  is a lengthscale defined as  $l = D/\sqrt{4aD + v^2}$ . Equations 6.15 and 6.16 represent the minimum influx required to form a droplet on the left or right and we plot these two curves in figure 6.8

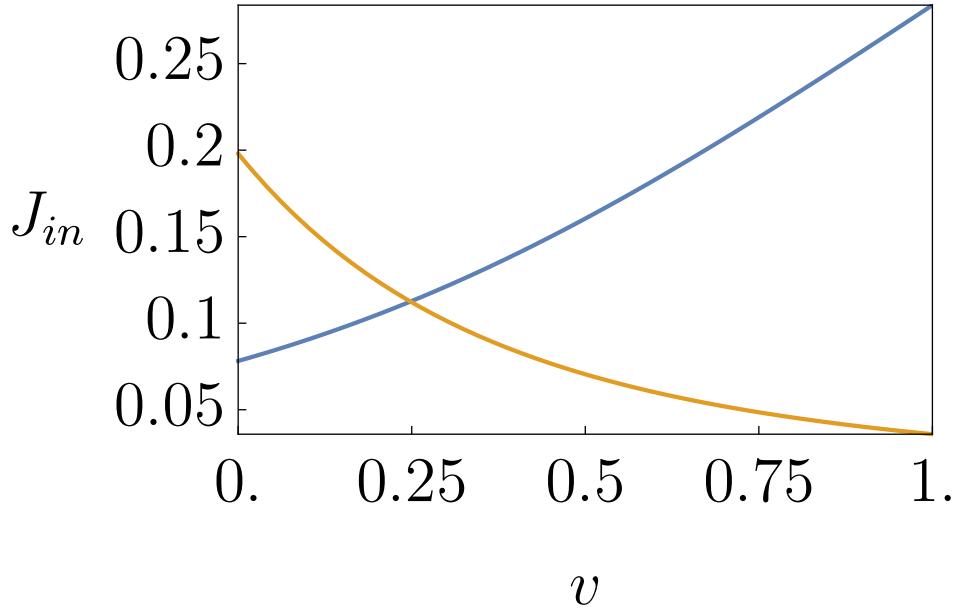


Figure 6.8: Blue line: Minimum  $J_{in}$  left side. Orange line: minimum  $J_{in}$  right side.

The blue line shows the minimum  $J_{in}$  for the left side of the system, while the orange line shows the minimum for the right. We can recognize four areas in figure 6.8. Below both the blue and the orange line, the concentration never reaches the lower inflection point and thus no droplets will be formed. In the area below the orange line but above the blue line, only a droplet on the left is formed, while exactly the reverse happens on the right side of the plot: only a droplet on the right is formed. Finally, we note that in the upper area both droplets can be formed. In this regime,  $J_{in}$  and  $v$  are high enough for the concentration to reach the inflection point at both sides of the system.

This approach only determines when a droplet is formed and does not yield any information about the size and stability of such newly-formed droplets. In fact, we require the existence of a stable droplet with non-zero radius in areas where figure 6.8 predicts a droplet is formed. We investigate this in the next section

Above both lines the concentration is high enough on both sides. This plot only tells us when phase-separation should happen; it doesn't tell if it does. We investigate this in the next section by comparing the phase diagram of figure 6.8 with the phase diagram of the effective droplet model.

### 6.3.2 EFFECTIVE DROPLET

We construct the effective droplet phase diagram for this system by determining for which  $J_{in}^*$  a stable droplet ( $dR/dt=0$ ) with radius  $R = 0$  exists. For an influx higher than  $J_{in}^*$ , a droplet with  $R > 0$  then exists, so that  $J_{in}^*$  is the minimum influx required for a droplet to exist. For the left and right droplet, we find the following:

$$\text{Left: } J_{in} = -\frac{2ac_o^-l}{\frac{vl}{D} - \coth(\frac{L}{2l})} \quad (6.17)$$

$$\text{Right: } J_{in} = ac_o^-l \left( e^{-\frac{L}{2l}(\frac{vl}{D}-1)} - e^{-\frac{L}{2l}(\frac{vl}{D}+1)} \right) \quad (6.18)$$

Note that these equations have the same form as 6.15 and 6.16, save for some prefactor. Defining the minimum flux as defined by equations 6.15 and 6.16 as  $J^{AD}$  and the minimum flux as calculated by the effective droplet model in equations 6.17 and 6.18 we obtain the same ratio for both the left and right sides:

$$\frac{J^{AD}}{J^{ED}} = \frac{(3 - \sqrt{3})c_o^+ + (3 + \sqrt{3})c_o^-}{6c_o^-}$$

Note that  $J^{AD} > J^{ED}$  if  $c_o^+ > c_o^-$ . In other words, the minimum flux required for a stable droplet is smaller than the minimum flux required to form a droplet if the concentration in the dense phase is higher than the concentration in the dilute phase, which it is by definition. We thus see that a stable droplet with a non-zero radius is guaranteed to exist if phase separation should occur as determined by equations 6.15 and 6.16. A second criterium would be mass conservation: the mass in a separated system should be similar to the mass in a phase separated system. Given any concentration profile (either above or below critical concentration) given by an advection-diffusion-decay equation, the mass in the system is:

$$\int_0^L c(x)dx = \frac{J_{in}}{a}$$

which makes sense as  $J_{in}$  is what comes into the system and  $\alpha$  what goes out. Without loss of generality, we assume  $J_{in}$  and  $\alpha$  are such the droplet appears on the left side of the system. The total mass inside such a system is:

$$\int_0^{R^*} c_{in}(x) dx + \int_{R^*}^L c_{out}(x) dx$$

Where the stable droplet radius  $R^*$  corresponds to  $dR/dt|_{R=R^*} = 0$ . Assuming the droplet radius remains small, we can determine the stable droplet radius. For simplicity assuming that  $k = \alpha$ , we obtain:

$$\int_0^{R^*} c_{in}(x) dx + \int_{R^*}^L c_{out}(x) dx = \frac{J_{in}}{\alpha}$$

We thus see that mass is conserved when phase separation occurs.

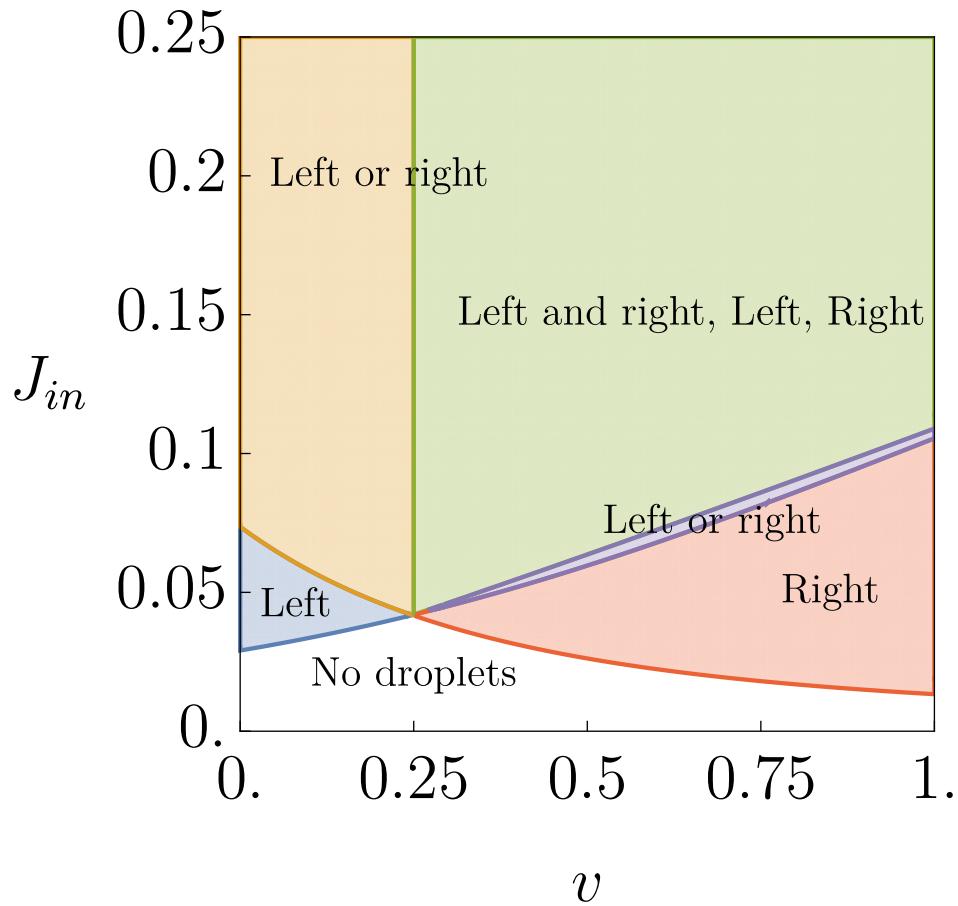
The effective droplet phase diagram also predicts an area in which droplets on both the left and right are stable. We now investigate if both droplets can coexist, i.e. that the system has a droplet on the left and the right, using a two droplet model. Again solving the fluxes for  $dR_1/dt = dR_2/dt = R_1 = R_2 = 0$  yields a minimum influx  $J_{in}$

$$J_{in} = \frac{c_o^- D}{2l} \left( \frac{vl}{D} + \frac{(1 + e^{\frac{L}{l}} - 2e^{\frac{-L}{2l}(\frac{vl}{D}-1)})}{(1 - e^{\frac{L}{l}})} \right)$$

and if  $l_D, l_v \gg L$  we also obtain a minimum advection velocity  $v^*$ :

$$v^* = \frac{\alpha L}{2} \quad (6.19)$$

We plot the corresponding phase diagram in figure 6.9 .



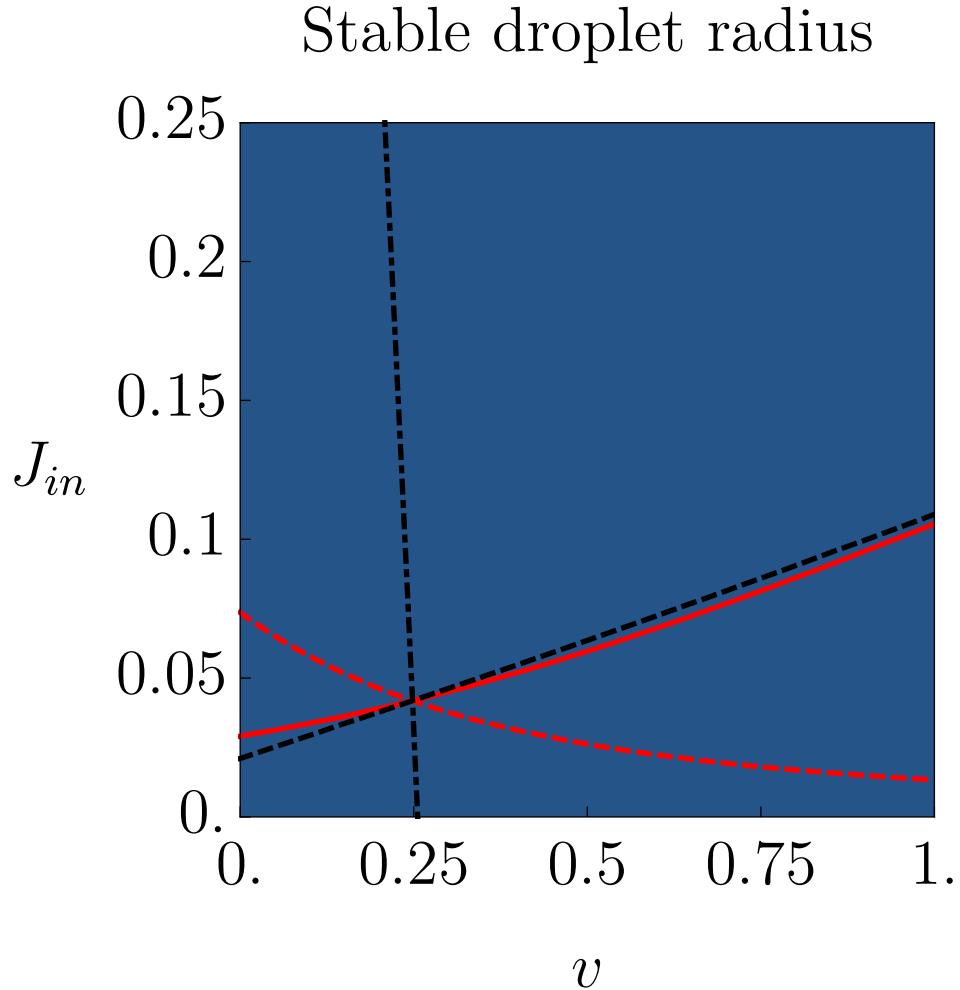
**Figure 6.9:** Caption.

We observe six ‘phases’ and that all these cross at a single point. Note some sort of symmetry exists: considering the area above the diagonal of the plot and following it clockwise from the origin, we first observe no droplets, then a droplet on the left, followed by either a droplet on the left or right and finally the area where its possible for a single droplet on the left or right or two droplets to exist. Below the diagonal we observe an analogous trajectory, where instead of a single droplet on the left we now have a droplet on the right. Considering the flow problem we presented in the previous section this symmetry makes perfect sense: for low  $v/J_{in}$  the concentration will be highest on the left hence favouring droplets being formed on the left. For high  $v/J_{in}$  exactly the opposite happens and droplets on the right are favoured. All phases intersect at a single point. Expanding the crossing of the mini-

mal fluxes for  $l_D, l_v \gg L$  we obtain

$$v_{cross} = \frac{aL}{2}$$

which is similar to equations 6.19, meaning that the intersection at this point is a system property and not just a side-effect of our parameter choice. In figure 6.10 we plot the results of numerically solving the equations.



**Figure 6.10:** Caption.

The red solid and dashed lines represent the minimum influx required for respectively a droplet on the left and right. The black dashed line corresponds to  $dR_{left}/dt = 0$  in the

two droplet model, while the dotted-dashed line is  $dR_{right}/dt = 0$ . Since only the line this line is determined numerically, all the other lines exactly match what is plotted in figure 6.9. Contrary to this plot however,  $dR_{right}/dt = 0$  is not vertical. This is because 6.19 has been derived assuming that both the left and right droplet appear at the same time; in reality, one of the droplet can have a non-zero radius before a second droplet appears. Numerically we also find that all the phases are connected through some critical point. To see why this happens, consider the crossing of the two minima  $J_{in}$ . At this point, the concentration profile is such that the concentration on each side of the system is similar:  $dR/dt = R = 0$  on both sides and by changing  $J_{in}$  or  $v$  any of the phases can be reached.

We now have several areas where multiple configurations are stable. Which one exactly depends on where the initial droplet is formed. Thus, the phase diagram depends on the initial dynamics. We have neglected the effect of the term penalizing the gradient and it could be that adding this term would yield a phase diagram with only a single stable state. This would require simulations however so for now the mystery remains.

#### 6.4 BIOLOGICAL IMPLICATIONS

We now link the behaviour of our model to biological observations. First off, our model predicts that the only stable position for the droplet is at the edges of the system. Biologically, stacks are either located at the ribbon or at the ERES, thus matching our model. Furthermore, this position is dependent on the microtubules: when the microtubules are depolymerized, the ribbon breaks up and the stacks colocalize with the ERES. Our free droplet model predicts a similar movement, with the switching happening at a finite advection velocity. Also note that our droplet size is dependent on the amount of trafficking through  $J_{in}$ . Observation have been made that the Golgi size is dependent on the amount of trafficking too.

Active droplets propel themselves by imbalanced fluxes; it grows on one side while decaying on the other. In the cisternal maturation model, the Golgi grows on one side by vesicles forming a cis compartment, whereas the opposite happens on the trans side. Cisternal maturation could thus be the process by which the stack moves from its position in the ribbon to the ERES.

All taken together, the model we've presented is able to explain the position of the Golgi stack, why it is formed around the ERES and why it moves close to the eres once the microtubules are depolymerized.

# 7

## Conclusion

In this thesis we have attempted to construct a model which couples the Golgi function and size to the intracellular transport and describe its biogenesis. We also wished to confront our model with data from the RUSH experiments. This was covered in the first part. Specifically, we have attempted to quantify the intracellular transport of the ManII protein from the ER to the Golgi. We have implemented two techniques. One was based on image gradients to determine the derivatives, which could then be used to perform a fit of the model. Applying this technique to the RUSH data yielded fits which showed some structure, but no clear pattern could be distinguished. Furthermore, the inferred diffusion field showed negative diffusion coefficients, which are probably due to errors in the fitting and not due to aggregation. We are thus unable to infer if the ER to Golgi transport is modeled by our model.

We foresee several directions to improve the image-gradient method. The most gains can be made in the step where the derivatives are calculated. The Sobel filter we've applied to determine the gradient is an extremely simple approximation and improving this step would vastly improve results. The fitting method we have applied - least squares - , is rather basic too and quite sensitive to outliers. Since the image gradients turn the model into a generic set of features, any fitting procedure could be used. In selecting a different fitting procedure, care should be taken to select a method which can properly handle noisy data with outliers.

We also implemented physics informed neural networks and applied it to the RUSH data. We have proven that it is possible to infer coefficient fields on synthetic data, but applying this method to the RUSH data yielded inconclusive and probably incorrect results. Despite

this, we believe PINNs show great promise and we propose several avenues to improve their performance. Implementing Bayesian neural networks would yield a distribution instead of a single value as an output. We believe this to be of prime importance, as the results of PINNs are not robust yet (i.e. sensitive to changes in the architecture). Secondly, the fitting of coefficient fields is a very hard problem with probably many local minima. To remedy this, we propose a ‘boots-trapping’ technique, in which one would first assume the coefficients to be locally constant, similar to the sliding window technique of the image-gradient method. Using this as a starting point, in the second step the fully spatially varying expressions is then used to improve the first step estimate.

We also make several recommendations to the experimentalists. First, they should settle on a quantification method: either single particle tracking or one of the methods we have presented in this thesis. Assuming our methods would be used to quantify the data, a calibration should be performed to learn the mapping of the fluorescence intensity to the concentration. Taking bright field images should also be considered, as these can be used for segmentation. Finally, the image acquisition procedure should be improved by ensuring a steady frame-rate and increasing it. Although increasing the frame rate increases photo bleaching, the last 200 frames of the roughly 300 frame movie contain almost no information about the intracellular transport, so there’s enough room to increase it by decreasing the movie length.

In the second part of the thesis we’ve studied active droplets. We’ve investigated the effect of advection on active droplets and found that it increases the concentration inside the droplet, compacting the droplet in an environment with a constant influx. We presented a model for the Golgi as such an active droplet, while the intracellular transport was modeled as an advection-diffusion equation. Such a model is able to explain how the Golgi is formed *de novo* and why the stacks collocate with the ERES. However, the phase diagram also showed the possibility for a droplet to exist on both sides of the system: something not observed in real cells. Furthermore, our model allowed for several configurations to exist, but did not predict which one would appear. Future work should thus address this using simulations.

# References

1. Emr, S. *et al.* Journeys through the Golgi—taking stock in a new era. *The Journal of Cell Biology* 187, 449–453 (2009).
2. Tang, D. & Wang, Y. Cell cycle regulation of Golgi membrane dynamics. *Trends in Cell Biology* 23, 296–304 (2013).
3. Rothman, J. E. The Future of Golgi Research. *Molecular Biology of the Cell* 21, 3776–3780 (2010).
4. Gosavi, P. & Gleeson, P. A. The Function of the Golgi Ribbon Structure - An Enduring Mystery Unfolds! *BioEssays* 39, 1700063 (2017).
5. Budnik, A. & Stephens, D. J. ER exit sites - Localization and control of COPII vesicle formation. *FEBS Letters* 583, 3796–3803 (2009).
6. Bressloff, P. C. & Newby, J. M. Stochastic models of intracellular transport. *Reviews of Modern Physics* 85, 135–196 (2013).
7. Wei, J.-H. & Seemann, J. Unraveling the Golgi Ribbon. *Traffic* 11, 1391–1400 (2010).
8. Ronchi, P., Tischer, C., Acehan, D. & Pepperkok, R. Positive feedback between Golgi membranes, microtubules and ER exit sites directs de novo biogenesis of the Golgi. *Journal of Cell Science* 127, 4620–4633 (2014).
9. Newby, J. M. & Bressloff, P. C. Quasi-steady State Reduction of Molecular Motor-Based Models of Directed Intermittent Search. *Bulletin of Mathematical Biology* 72, 1840–1866 (2010).
10. Newby, J. & Bressloff, P. C. Random intermittent search and the tug-of-war model of motor-driven transport. *Journal of Statistical Mechanics: Theory and Experiment* 2010, Po4014 (2010).
11. Alberts. *Molecular biology of the Cell*.
12. Staehelin, L. A. & Kang, B.-H. Nanoscale Architecture of Endoplasmic Reticulum Export Sites and of Golgi Membranes as Determined by Electron Tomography. *PLANT PHYSIOLOGY* 147, 1454–1468 (2008).
13. Griffiths, H. Rubisco is said to be both the most important enzyme on Earth and surprisingly inefficient. Yet an understanding of the reaction by which it fixes CO<sub>2</sub> suggests that evolution has made the best of a bad job. 2
14. Glick, B. S. & Luini, A. Models for Golgi Traffic: A Critical Assessment. *Cold Spring Harbor Perspectives in Biology* 3, a005215–a005215 (2011).
15. Hirschberg, K. *et al.* Kinetic Analysis of Secretory Protein Traffic and Characterization of Golgi to Plasma Membrane Transport Intermediates in Living Cells. *The Journal of Cell Biology* 143, 1485–1503 (1998).
16. Boncompain, G. *et al.* Synchronization of secretory protein traffic in populations of cells. *Nature Methods* 9, 493–498 (2012).

17. Holcman, D. Modeling DNA and Virus Trafficking in the Cell Cytoplasm. *Journal of Statistical Physics* 127, 471–494 (2007).
18. Lagache, T. & Holcman, D. Effective Motion of a Virus Trafficking Inside a Biological Cell. *SIAM Journal on Applied Mathematics* 68, 1146–1167 (2008).
19. Dinh, A.-T., Theofanous, T. & Mitragotri, S. A Model for Intracellular Trafficking of Adenoviral Vectors. *Biophysical Journal* 89, 1574–1588 (2005).
20. Brandenburg, B. & Zhuang, X. Virus trafficking – learning from single-virus tracking. *Nature Reviews Microbiology* 5, 197–208 (2007).
21. Zwicker, D., Seyboldt, R., Weber, C. A., Hyman, A. A. & Jülicher, F. Growth and division of active droplets provides a model for protocells. *Nature Physics* 13, 408–413 (2017).
22. Hyman, A. A., Weber, C. A. & Jülicher, F. Liquid-Liquid Phase Separation in Biology. *Annual Review of Cell and Developmental Biology* 30, 39–58 (2014).
23. Zwicker, D., Decker, M., Jaensch, S., Hyman, A. A. & Jülicher, F. Centrosomes are autocatalytic droplets of pericentriolar material organized by centrioles. *Proceedings of the National Academy of Sciences* 111, E2636–E2645 (2014).
24. Sbalzarini, I. F. Seeing Is Believing: Quantifying Is Convincing: Computational Image Analysis in Biology. in *Focus on Bio-Image Informatics* (eds. De Vos, W. H., Munck, S. & Timmermans, J.-P.) 219, 1–39 (Springer International Publishing, 2016).
25. Chen, K.-C., Qiu, M., Kovacevic, J. & Yang, G. Computational Image Modeling for Characterization and Analysis of Intracellular Cargo Transport. in *Computational Modeling of Objects Presented in Images. Fundamentals, Methods, and Applications* (eds. Hutchison, D. et al.) 8641, 292–303 (Springer International Publishing, 2014).
26. Lee, H.-C. & Yang, G. An image-based computational method for characterizing whole-cell scale spatiotemporal dynamics of intracellular transport. in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)* 699–702 (IEEE, 2015). doi:[10.1109/ISBI.2015.7163969](https://doi.org/10.1109/ISBI.2015.7163969)
27. Yang, G. Bioimage informatics for understanding spatiotemporal dynamics of cellular processes: Spatiotemporal dynamics of cellular processes. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* 5, 367–380 (2013).
28. Hebert, B., Costantino, S. & Wiseman, P. W. Spatiotemporal Image Correlation Spectroscopy (STICS) Theory, Verification, and Application to Protein Velocity Mapping in Living CHO Cells. *Biophysical Journal* 88, 3601–3614 (2005).
29. Kisley, L. et al. Characterization of Porous Materials by Fluorescence Correlation Spectroscopy Super-resolution Optical Fluctuation Imaging. *ACS Nano* 9, 9158–9166 (2015).
30. Semrau, S. & Schmidt, T. Particle Image Correlation Spectroscopy (PICS): Retrieving Nanometer-Scale Correlations from High-Density Single-Molecule Position Data. *Biophysical Journal* 92, 613–621 (2007).
31. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations. *arXiv:1711.10566 [cs, math, stat]* (2017).
32. Barron, J. L., Fleet, D. J. & Beauchemin, S. S. Performance of Optical Flow Techniques. 60 (1994).

33. Dong, G., Baskin, T. I. & Palaniappan, K. Motion Flow Estimation from Image Sequences with Applications to Biological Growth and Motility. in *2006 International Conference on Image Processing* 1245–1248 (IEEE, 2006). doi:[10.1109/ICIP.2006.312551](https://doi.org/10.1109/ICIP.2006.312551)
34. Vig, D. K., Hamby, A. E. & Wolgemuth, C. W. On the Quantification of Cellular Velocity Fields. *Biophysical Journal* 110, 1469–1475 (2016).
35. Garcia, D. Robust smoothing of gridded data in one and higher dimensions with missing values. *Computational Statistics & Data Analysis* 54, 1167–1178 (2010).
36. Zimoń, M., Reese, J. & Emerson, D. A novel coupling of noise reduction algorithms for particle flow simulations. *Journal of Computational Physics* 321, 169–190 (2016).
37. Zimoń, M. *et al.* An evaluation of noise reduction algorithms for particle-based fluid simulations in multi-scale applications. *Journal of Computational Physics* 325, 380–394 (2016).
38. Grinberg, L., Yakhot, A. & Karniadakis, G. E. Analyzing Transient Turbulence in a Stenosed Carotid Artery by Proper Orthogonal Decomposition. *Annals of Biomedical Engineering* 37, 2200–2217 (2009).
39. Grinberg, L. Proper orthogonal decomposition of atomistic flow simulations. *Journal of Computational Physics* 231, 5542–5556 (2012).
40. Bruno, O. & Hoch, D. Numerical Differentiation of Approximated Functions with Limited Order-of-Accuracy Deterioration. *SIAM Journal on Numerical Analysis* 50, 1581–1603 (2012).
41. Knowles, I. & Renka, R. J. METHODS FOR NUMERICAL DIFFERENTIATION OF NOISY DATA. 12
42. Rizk, A. *et al.* Segmentation and quantification of subcellular structures in fluorescence microscopy images using Squash. *Nature Protocols* 9, 586–596 (2014).
43. Karpatne, A., Watkins, W., Read, J. & Kumar, V. Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling. *arXiv:1710.11431 [physics, stat]* (2017).
44. Sharma, R., Farimani, A. B., Gomes, J., Eastman, P. & Pande, V. Weakly-Supervised Deep Learning of Heat Transport via Physics Informed Loss. *arXiv:1807.11374 [cs, stat]* (2018).
45. Pun, G. P. P., Batra, R., Ramprasad, R. & Mishin, Y. Physically-informed artificial neural networks for atomistic modeling of materials. *arXiv:1808.01696 [cond-mat]* (2018).
46. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. *arXiv:1711.10561 [cs, math, stat]* (2017).
47. Bray, A. J. Theory of phase-ordering kinetics. *Advances in Physics* 51, 481–587 (2002).
48. Weber, C. A., Zwicker, D., Jülicher, F. & Lee, C. F. Physics of Active Emulsions. *arXiv:1806.09552 [cond-mat]* (2018).
49. Zwicker, D., Hyman, A. A. & Jülicher, F. Suppression of Ostwald ripening in active emulsions. *Physical Review E* 92, (2015).
50. Sengupta, D. & Linstedt, A. D. Control of Organelle Size: The Golgi Complex. *Annual Review of Cell and Developmental Biology* 27, 57–77 (2011).
51. Weber, C. A., Lee, C. F. & Jülicher, F. Droplet ripening in concentration gradients. *New Journal of Physics* 19, 053021 (2017).