

# Numerical Simulation of Fluorescent Microscopy Images of Living Cells



Christian V. Hansen  
University of Southern Denmark

Master's Degree Project  
Advisor: Achim Schroll  
Department of Mathematics and Computer Science  
At University of Southern Denmark, Odense.  
September 16, 2014.

## Abstract

Fluorescence Loss In Photobleaching (FLIP) is a modern microscopy method for visualization of transport processes in living cells. Although FLIP is widespread, an automated reliable analysis of image data is still lacking. This report presents a well-posed computational model based on spatially resolved diffusion and transport rates. The model is a reaction–diffusion system, discretized by continuous finite elements. The cell geometry is segmented from FLIP images using an active contours algorithm and the PDE model is subsequently solved in real, two-dimensional geometry. Based on this model, FLIP images are simulated and thus molecular transport in living cells is reliably quantified.

As a result of this Master’s Degree Project, we have produced the paper *Computational Modeling of Fluorescence Loss in Photobleaching*, which can be found in Appendix A.

## Acknowledgements

I would like to thank my eminent advisor Achim Schroll, for his great support, advice, experience, patience, drive and enthusiasm. In our discussions and your lectures, I always find something fruitful. You are always there to help me, for that I am very grateful.

I would also like to thank Daniel Wüstner whose enormous knowledge about the FLIP method and other biological aspects have made this project possible. Thank you so much for sharing your knowledge with us and for a good working relationship.

Furthermore, I would like to thank Niels Chr. Overgaard from Lund University for introducing us to level set methods in image segmentation.

From the depth of my heart I am thankful to these people, without whom, especially my advisor Achim Schroll who has spend a lot of time working on this topic, this project would not have been possible.

# Contents

<b>1</b>	<b>Motivation</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>5</b>
<b>3</b>	<b>Time Dependent Diffusion</b>	<b>7</b>
3.1	Diffusion Equation . . . . .	7
3.1.1	Reaction-Diffusion Equation . . . . .	8
3.2	Chemical Reaction . . . . .	9
3.3	Boundary Conditions . . . . .	11
3.4	Discretization . . . . .	12
3.4.1	Von Neumann Stability Analysis . . . . .	14
3.4.2	Stability . . . . .	16
3.5	Positivity and Maximum Principle . . . . .	17
<b>4</b>	<b>A Reaction-Diffusion Model for FLIP in realistic geometry</b>	<b>20</b>
4.1	Chan-Vese Segmentation of FLIP Images . . . . .	20
4.2	A Membrane Transport Model . . . . .	22
4.2.1	Reaction-Diffusion Model for FLIP Experiments . . . . .	24
<b>5</b>	<b>Analysis</b>	<b>26</b>
5.1	Positivity Preserving Property for a Reaction-Diffusion Model . .	28
5.2	Stability and Convergence . . . . .	31
<b>6</b>	<b>Implementation</b>	<b>35</b>
6.1	Mesh and Cell Geometry . . . . .	35
6.2	Numerical Method . . . . .	37
6.3	FEniCS Implementation . . . . .	40
6.4	Tuning Reaction- and Diffusion Parameters . . . . .	43
6.5	Simulation on Real Cell Geometry . . . . .	44
6.6	Diffusion Rates as Transport Rates . . . . .	47
<b>7</b>	<b>Final Remarks</b>	<b>48</b>
<b>A</b>	<b>Paper: Computational Modeling of Fluorescence Loss in Photobleaching</b>	<b>49</b>

# Chapter 1

## Motivation

From today's research it is well known that diseases like Alzheimer, Parkinson, Corea Huntington and Arteriosclerosis are caused by a jam in intracellular membrane traffic [24]. Hence to improve treatment, a quantitative analysis of intracellular transport is essential. One important method for visualization of the transport processes in living cells, is the so called Fluorescence loss in photobleaching (FLIP). Since FLIP is a modern and widely used microscopy method, the motivation for making an automated reliable analysis of the image data is high.

To develop such an automated computer analysis or simulation for FLIP images, the first step would be to identify the geometry of the cell.

In my Bachelor project "*Segmentation of fluorescent microscopy images of living cells*" I used the "Chan-Vese" method, also known as "Active Contours Without Edges," to segment images. The project successfully implemented the Chan-Vese method, such that it can be used to detect cell edges on the FLIP images.

On the basis of this segmentation and the FLIP images, provided by Daniel Wüstner at BMB, SDU, we are now ready to take the next step towards an automated computer analysis/simulation of these cells.

## Chapter 2

# Introduction

For this project, we need base knowledge about Fluorescence Loss In Photobleaching (FLIP).

In studies of fluorescently tagged proteins the most commonly used techniques are FLIP and FRAP (Fluorescence Recovery After Photobleaching), where a cellular region is photobleached using a high-intensity laser pulse. These techniques are used to study the movement and thereby the kinetic properties of proteins in living cells such as diffusion rates and mobile fraction [21].

The most commonly used technique is FRAP. Usually, FRAP involves bleaching of a small area in a cell. After bleaching, the movement of unbleached molecules from the neighborhood into the bleached area is recorded as the fluorescence recovery, which is illustrated in Figure 2.1.

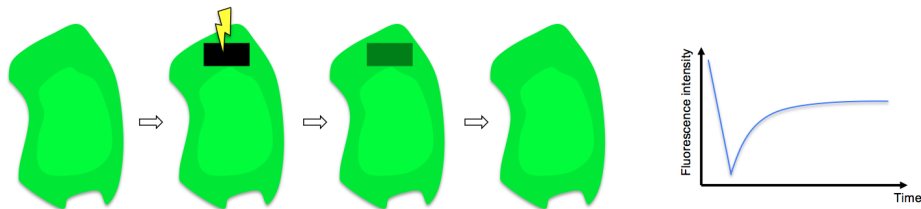


Figure 2.1: FRAP on a cell with a green fluorescent protein.

Contrary to FRAP, the area of interest in FLIP experiments is not only the bleaching area, but the whole cell. In FLIP, a small cellular region is repeatedly bleached, while images are taken in between the bleaches. This procedure is illustrated in Figure 2.2.

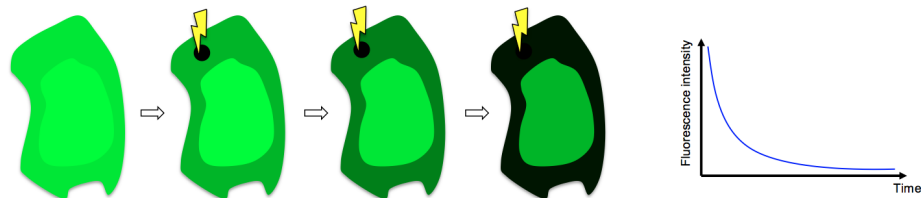


Figure 2.2: FLIP. Here bleaching occurs in cytoplasm, the lighter area illustrates nucleus.

FLIP is often used to study the cellular structure through the movement of proteins and is a very popular technique. But in spite of its popularity an automated, reliable analysis of image data is still lacking. So far only few FLIP analyses published, include the whole image information in the analysis. [21, 23]

Several fluorescent proteins exists. The most common is the green fluorescent protein (GFP), which refers to the protein isolated from the jellyfish *Aequorea victoria*. In the experiments introduced below we use the enhanced green fluorescent protein (eGFP) which, because of its structure, should not bind to any cellular structures. It is known that eGFP diffuses very rapidly ( $D \approx 25 \mu m^2/s$ ) [23]. In proportion to this the exchange of eGFP between nucleus and cytoplasm seems rather slow, a tendency that can be observed in Figure 2.3.

Studies show that proteins like eGFP are limited to transport through the nuclear pore complex. Several diffusion models have been proposed but none have so far been generally approved. It is known that the transport of eGFP between nucleus and cytoplasm is slower than the diffusion inside cytoplasm and nucleus [8, 25].

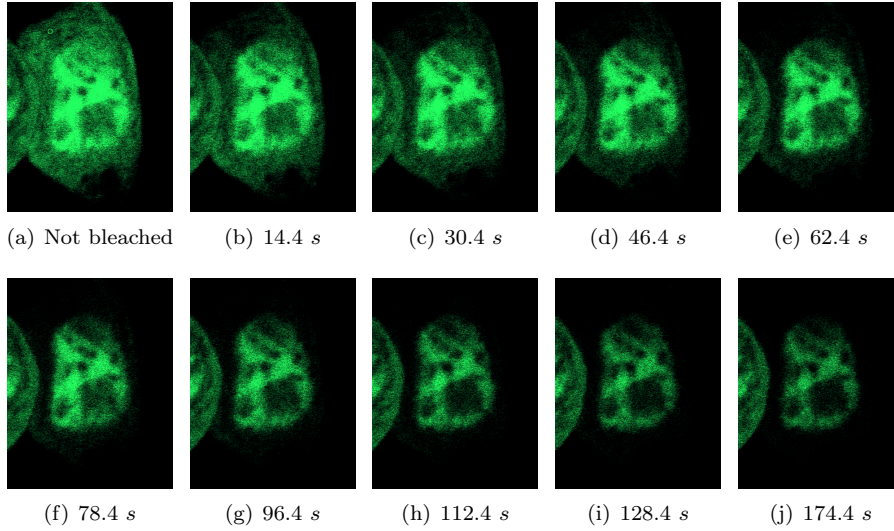


Figure 2.3: Diffusion-limited FLIP of eGFP in the cytoplasm of McA cells.

Figure 2.3 shows a sequence of FLIP images from an experiment on a McArdle RH77777 cell with eGFP in a temperature-controlled environment ( $35 \pm 1^\circ C$ ). The bleaching with full laser power was conducted on a small 10 pixel diameter circular region in the cytoplasm (green circle in Fig. 2.3(a)). The whole image was scanned with 0.5% laser output between the bleach scans and with a resolution at  $0.0448 \mu m/pixel$ . The frame rate therefor became 1.6 sec. Between frame 50 and 51 there occurred a short break of approximately 2 sec. The FLIP sequence shown in figure 2.3 is the one that we would like to model and simulate later via a Reaction-Diffusion model.

## Chapter 3

# Time Dependent Diffusion

Diffusion is the result of random molecular motions. It is hard to find examples of pure diffusion in our everyday life, since transport in liquids and gases on a larger scale is often due to convection. Thus we normally only observe a pure molecular diffusion in small-scale experiments. The difference between convection and diffusion is that the motion caused by convection is a result from a forced act upon the system, while diffusion refers to a result of random molecular motions.

Let us consider the example where a lump of sugar is dropped into a beaker of still water. We note that the molecules in the water are in a constant random motion which result in a evenly distribution of the water molecules. The lump of sugar consists of many sugar molecules which are in motion all the time, even in its solid form. In the water the lump dissolves, and sugar molecules randomly diffuse out in the water. This continues until all the sugar molecules are evenly distributed in the water.

Diffusion transport is the result of random motions of all molecules inside the system. Thus if we consider one specific molecule we observe that the motion of this is random and nearly always independent of the others. The random movements give raise to collisions and thereby change the direction of movement, thus the molecules have no preferred direction of motion neither towards a region of higher or lower concentration. Nevertheless we observe that molecules move from a region of higher concentration to one of lower concentration. In 1855 Adolf Fick was able to adapt this observation to what we today know as Fick's laws of diffusion or simply the "Diffusion Equation".

### 3.1 Diffusion Equation

**Fick's first law** describes the relation between the flux and the concentration, where flux is the flow rate or rate of diffusing substance per unit area. As mentioned Fick's first law is based on the hypothesis that the flux goes from a region of higher concentration to one of lower concentration, so that the flux is proportional to the concentration gradient. Thus Fick's first law is given by

$$J(t, x, u) = -d \frac{\partial u(t, x)}{\partial x} \quad (3.1)$$

where  $J = J(t, x)$  is the diffusion flux as described above and  $u = u(t, x)$  is the concentration of the substance. Both  $J$  and  $u$  depend on space  $x$  and time  $t$ . In cases where the material or solution (in which the diffusion is happening) is relatively homogeneous or diluted, the diffusion coefficient  $d$  is often chosen to be constant, while in heterogeneous mixture or in high polymers, it will depend on space and/or on the concentration.

**Fick's second law**, which is also known as the “Fundamental Differential Equation of Diffusion”, can be derived from (3.1) in combination with the law of conservation of mass.

Now let  $u, J \in C^1$  and consider the change in total concentration with respect to time inside the region  $\Omega \subseteq \mathbb{R}$ :

$$\frac{d}{dt} \int_{\Omega} u(t, x) \, dx . \quad (3.2)$$

Also observe that the total out flux is:

$$\int_{\partial\Omega} J(t, x, u) \, ds . \quad (3.3)$$

where  $\partial\Omega$  is the boundary of  $\Omega$ .

The principle of mass conservation is, that the system mass cannot change quantity if it is not added or removed. Consequently, the rate of change of the total concentration and the out flux needs to sum to zero. This can be expressed as:

$$\frac{d}{dt} \int_{\Omega} u \, dx = - \int_{\partial\Omega} J \, ds . \quad (3.4)$$

Applying Leibniz's rule and integration by parts, gives:

$$\int_{\Omega} \frac{\partial}{\partial t} u \, dx = - \int_{\Omega} \frac{\partial}{\partial x} J \, dx . \quad (3.5)$$

Since  $\Omega$  is arbitrary, we arrive at what is also known as the Diffusion Equation in one dimension:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( d \frac{\partial u}{\partial x} \right) . \quad (3.6)$$

Thus, if  $d$  is constant Fick's second law becomes:

$$\frac{\partial u}{\partial t} = d \frac{\partial^2 u}{\partial x^2} . \quad (3.7)$$

In this project we will use an extended version of this diffusion equation called the “Reaction-Diffusion Equation”.

### 3.1.1 Reaction-Diffusion Equation

Reaction-Diffusion Equation arises where you may observe or perform a local change in the concentration  $u$  which is not caused by the diffusion. We would describe this change by:

$$\frac{\partial u}{\partial t} = f(t, x, u) , \quad (3.8)$$



where  $f$  is a function depending on space, time and concentration. By combining this with (3.6) we get the Reaction-Diffusion Equation

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( d \frac{\partial u}{\partial x} \right) + f(t, x, u) . \quad (3.9)$$

In general we will consider (3.9) for  $x$  in the spatial domain  $\Omega \subseteq \mathbb{R}$ , for time  $t > 0$ . Likewise we would also have an initial condition and boundary condition on the boundary  $\partial\Omega$ . Thus, a typical Reaction-Diffusion problem would be:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial}{\partial x} \left( d \frac{\partial u}{\partial x} \right) + f(t, x, u) \text{ in } \Omega, \text{ for } t > 0 \\ u &= b \text{ on } \partial\Omega, \text{ for } t > 0 \\ u &= u^0 \text{ at } t = 0. \end{aligned} \quad (3.10)$$

Here,  $u$  varies in space and time, e.g.  $u = u(t, x)$  just as  $f$  and  $b$  may also vary with space and time, where the initial condition  $u^0$  only varies in space. The diffusion problem above is only considered in one-dimension, but the area of interest for this project is primarily in the two or three dimensional case. Next we will consider the multi-dimensional case.

### Multi-Dimensional Problem

Let us consider the reaction-diffusion problem from (3.9) in  $\mathbb{R}^3$ . First denote the gradient by  $\nabla = \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)^T$  and if  $s, t$  are vectors in  $\mathbb{R}^3$ , then let  $s \cdot t$  be the Euclidean inner product. Thus the divergence of a differentiable vector function  $s = (s_1, s_2, s_3)^T$  is  $\nabla \cdot s = \frac{\partial s_1}{\partial x} + \frac{\partial s_2}{\partial y} + \frac{\partial s_3}{\partial z}$ .

Later we will use the Laplace operator  $\Delta = \nabla \cdot \nabla$ , which for a differential scalar function  $u$  on  $\mathbb{R}^3$  is given by:

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} . \quad (3.11)$$

Thus the reaction-diffusion problem from (3.10) in multi-dimensional becomes:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \nabla \cdot (d \nabla u) + f(t, x, u) \text{ in } \Omega, \text{ for } t > 0 \\ u &= b \text{ on } \partial\Omega, \text{ for } t > 0 \\ u &= u^0 \text{ at } t = 0. \end{aligned} \quad (3.12)$$

The diffusion coefficient can be a matrix but often it is constant and then just viewed as a scalar. In this case the differential equation from (3.12) becomes:

$$\frac{\partial u}{\partial t} = d \Delta u + f(t, x, u) . \quad (3.13)$$

## 3.2 Chemical Reaction

To understand how a chemical reaction can be incorporated into the general Reaction-Diffusion Equation discussed above, let us now consider the single chemical reaction:



a reaction of two species U and V into a chemical complex of U and V, here denoted by the species W, with reaction rate  $k$ . Now let  $u = [U]$ ,  $v = [V]$  and  $w = [W]$  denote the concentrations of the three species, thus by the law of mass action, the speed of the reaction is proportional to the product of the reactants' concentrations. When the partial derivatives are denoted with the usual subscripts such that  $\partial_t = \frac{\partial}{\partial t}$ , this gives the following ODE system:

$$\begin{aligned}u_t &= -kuv \\v_t &= -kuv \\w_t &= kuv.\end{aligned}$$

Note that  $u_t + w_t = 0$  and  $v_t + w_t = 0$ . Consequently, we observe that  $u(0) + w(0) = u(t) + w(t)$  and  $v(0) + w(0) = v(t) + w(t)$ .

From these two equations we deduce that the total amount of atoms remains constant. This is in fact a well-known relation from the *laws of mass conservation*, which states that for any closed chemical reaction system, the mass of the system must remain constant over time.

When working with chemical reaction systems, an obvious but important property of the chemical concentrations is, that they are kept non-negative, a property that of course should hold in the mathematical model too. This is particularly important when working with numerical methods; even small negative values caused by numerical errors can lead to instability. It can relatively easily be shown that the above ODE system maintains positivity, see [18].

A more interesting chemical reaction for this project is a reversible reaction, which can be presented as:



Similar to the single reaction system, this provides basis for the following ODE system:

$$\begin{aligned}u_t &= k^-v - k^+u, \\v_t &= k^+u - k^-v,\end{aligned} \quad (3.16)$$

with reaction constants  $k^\pm > 0$  and concentrations  $u = [U]$  and  $v = [V]$ .

Observe that when equilibrium is obtained  $u_t = v_t = 0$ . From this we immediately obtain:

$$u \frac{k^+}{k^-} = v. \quad (3.17)$$

Where  $K_c = \frac{k^+}{k^-}$  is known as the equilibrium constant, which from (3.17) also can be observed as the ratio between the product and the reactant from the chemical system, i.e.:

$$K_c = \frac{k^+}{k^-} = \frac{v}{u}. \quad (3.18)$$

Now let  $w = \begin{pmatrix} u \\ v \end{pmatrix}$ , such that the system in vector form can be written as  $w_t = Aw$  with matrix:

$$A = \begin{pmatrix} -k^+ & k^- \\ k^+ & -k^- \end{pmatrix}. \quad (3.19)$$

It is now possible to derive the exact solution by the Eigenvalue method. First consider:

$$\begin{aligned}
 \det(A - \lambda I) &= \begin{vmatrix} -k^+ - \lambda & k^- \\ k^+ & -k^- - \lambda \end{vmatrix} \\
 &= (-k^+ - \lambda)(-k^- - \lambda) - k^-k^+ \\
 &= \lambda k^+ + \lambda k^- + \lambda^2 \\
 &= \lambda(k^+ + k^- + \lambda).
 \end{aligned} \tag{3.20}$$

Since the eigenvalues are the roots of the characteristic polynomial, i.e. when  $\det(A - \lambda I) = 0$ , we observe that  $\lambda = 0$  or  $\lambda = -(k^+ + k^-)$  both are eigenvalues. An eigenvector  $y$  associated to  $\lambda$  is given by the equation:

$$Ay = \lambda y. \tag{3.21}$$

Thus when  $\lambda = 0$  we observe that  $Ay = 0$ , which gives the eigenvector  $y = \begin{pmatrix} k^- \\ k^+ \end{pmatrix}$ . For  $\lambda = -(k^+ + k^-)$  we choose  $y = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$  to be the eigenvector. Thus the general solution is:

$$\begin{aligned}
 u &= c_1 k^- + c_2 e^{-(k^+ + k^-)t}, \\
 v &= c_1 k^+ - c_2 e^{-(k^+ + k^-)t},
 \end{aligned} \tag{3.22}$$

where  $c_1 = \frac{u(0)+v(0)}{k^-+k^+}$  and  $c_2 = \frac{k^+u(0)+k^-v(0)}{k^-+k^+}$  are constants.

### 3.3 Boundary Conditions

One of the most common boundary conditions is the Dirichlet Boundary Condition. The Dirichlet condition specifies the values that a solution should take on the boundary. In this project we will consider the cell as a closed system, hence we assume that the cell membrane works like a closed wall around the domain. As such the Dirichlet boundaries are not ideal.

Assuming that the cell is closed we thus have a non flux boundary. We can achieve such a boundary using another common boundary condition namely Neumann. The Neumann Boundary Condition specifies what value the derivative in the normal direction of a solution has to take on the boundary. Thus, in general, the Neumann Boundary Condition on a domain  $\Omega \in \mathbb{R}^n$  is:

$$\frac{\partial u}{\partial n} = g \quad \forall x \in \partial\Omega, \tag{3.23}$$

Where  $n$  is the outward normal on  $\partial\Omega$  and  $g$  is a scalar function depending on  $x$ . By Fick's first law (3.1) and the Neumann boundary from (3.23) we obtain a non flux boundary if  $g = 0$ .

### 3.4 Discretization

A great number of numerical methods exist for approximating the solution to a differential equation. One of the best known approximation methods is the Finite-Difference Methods (FDM), which is used to approximate derivatives. When using the FDM, the domain where the independent variables are defined, is replaced by a finite mesh of points in which the approximation is made. Another popular method is the Finite Element Method (FEM). For simplicity and illustration of FDM and FEM let us consider a two-point boundary value problem in one-dimension:

$$\begin{aligned} -u''(x) &= f(x), \quad \text{for } 0 < x < 1 \\ u(0) &= u(1) = 0. \end{aligned} \quad (3.24)$$

The basic idea behind the FDM is to replace every derivative with finite differences. In the following, we consider  $U_j$  to be the numerical approximation to the exact solution value  $u(x_j)$ . Let  $h = \frac{1}{m+1}$  be the grid spacing, where  $m$  is a positive integer, such that  $m \geq 1$  and  $x_j = jh$ . To achieve the discrete scheme, the second order derivative  $u''$  is approximated by the second-order central difference, therefore the discrete FDM scheme for (3.24) becomes:

$$-\left(\frac{U_{j-1} - 2U_j + U_{j+1}}{h^2}\right) = f(x) \quad \text{for } j = 1, \dots, m, \quad n \geq 0, \quad (3.25)$$

where  $U_0 = U_{m+1} = 0$ . Here the errors are quadratic in the space step,  $\delta U = O(h^2)$ . Thus the linear operator for homogeneous Dirichlet condition with boundaries as in (3.24) becomes:

$$-u_{xx} \approx A_h U = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix} U, \quad (3.26)$$

where  $U = (U_1, U_2, \dots, U_m)^T$  is the vector which holds the approximations on the inner grid points.

In this project we use the FDM in one dimensional simulation.

For simulation in the two dimensional case FEM will be used. In this case first step is to reformulate the PDE into a variational problem, then use the FEM for discretization after which the solution to the problem is found. To formulate the variational problem we introduce the linear test space:

$$V = \{\phi \in H^1(\Omega) : \phi(0) = \phi(1) = 0\}, \quad (3.27)$$

where  $H^1(\Omega)$  denotes the usual Sobolev space:

$$H^1(\Omega) = \left\{ u \in L^2(\Omega) \mid \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2} \in L^2(\Omega) \right\}. \quad (3.28)$$

Next we introduce the finite-dimensional subspace  $V_h \subset V$ . Let  $0 = x_0 < x_1 < \dots < x_m < x_{m+1} = 1$  be a partition of the interval  $[0, 1]$  into subintervals of

length  $h_j = x_j - x_{j-1}$ ,  $j = 1, \dots, m+1$ . The choice of  $V_h$  follows from the kind of finite elements we want. Here we let  $V_h$  be the set of functions  $\phi$ , such that,  $\phi$  is linear on each subinterval, continuous on  $[0, 1]$  and  $\phi(0) = \phi(1) = 0$ . Now we introduce the basis functions  $\rho_j \in V_h$ ,  $j = 1, \dots, m$  defined by:

$$\rho_j(x_i) = \begin{cases} 1, & j = i, \\ 0, & j \neq i. \end{cases} \quad (3.29)$$

For a function  $u_h \in V_h$  we may write  $U_j = u_h(x_j)$  for  $j = 0, \dots, m+1$ . Thus  $u_h$  can be written as:

$$u_h(x) = \sum_{i=1}^m U_i \rho_i(x), \quad x \in [0, 1]. \quad (3.30)$$

Therefore, each  $u_h \in V_h$  can be represented as a linear combination of  $\rho_i$ , i.e.  $V_h$  is a linear space of dimension  $m$  with basis  $\{\rho_i\}_{i=1}^m$ .

By Galerkin's method we now express the boundary value problem (3.24) as a variational problem. This is done by multiplying the PDE by a test function  $\phi$  and integrating the equation over  $\Omega$ :

$$-\int_0^1 u'' \phi \, dx = \int_0^1 f \phi \, dx. \quad (3.31)$$

Applying integration by parts on the left hand side gives:

$$-\int_0^1 u'' \phi \, dx = \int_0^1 u' \phi' \, dx - [u' \phi]_0^1 \quad (3.32)$$

Since  $\phi$  is required to vanish on the parts of the boundary where  $u$  is known, it follows from applying (3.32) on (3.31) that:

$$\int_0^1 u' \phi' \, dx = \int_0^1 f \phi \, dx \quad \forall \phi \in V. \quad (3.33)$$

Thus the discrete variational problem reads: find  $u_h \in V_h$  such that

$$\int_0^1 u_h' \phi' \, dx = \int_0^1 f \phi \, dx \quad \forall \phi \in V_h. \quad (3.34)$$

The Finite Element Method for (3.24) can thus be formulated as (3.34). If  $u_h$  satisfies (3.34) then in particular

$$\int_0^1 u_h' \rho_j' \, dx = \int_0^1 f \rho_j \, dx, \quad j = 1, \dots, m. \quad (3.35)$$

From (3.30) we can now write (3.35) as:

$$\sum_{i=1}^m U_i \int_0^1 \rho_i' \rho_j' \, dx = \int_0^1 f \rho_j \, dx, \quad j = 1, \dots, m, \quad (3.36)$$

which is a linear system of  $m$  equations in  $m$  unknowns  $U_1, \dots, U_m$ . In matrix form the linear system (3.36) can be written as:

$$AU = b. \quad (3.37)$$

The elements  $a_{ij}$  in the matrix  $A$  can easily be computed. Observe that if  $|i - j| > 1$  then  $a_{ij} = \int_0^1 \rho'_i \rho'_j dx = 0$ . Thus we get a tri-diagonal matrix, where the diagonal for  $j = 1, \dots, m$  is:

$$\int_0^1 \rho'_j \rho'_j dx = \int_{x_{j-1}}^{x_j} \frac{1}{h_j^2} dx + \int_{x_j}^{x_{j+1}} \frac{1}{h_{j+1}^2} dx = \frac{1}{h_j} + \frac{1}{h_{j+1}}. \quad (3.38)$$

For  $j = 2, \dots, m$ :

$$\int_0^1 \rho'_j \rho'_{j-1} dx = \int_0^1 \rho'_{j-1} \rho'_j dx = - \int_{x_{j-1}}^{x_j} \frac{1}{h_j^2} dx = -\frac{1}{h_j}. \quad (3.39)$$

Note that  $A$  is symmetric and positive definite, thus  $A$  is a regular (non-singular) matrix and it follows that the linear system has a unique solution.

In the case where all partitions of the domain have equal lengths  $h_j = h = \frac{1}{m+1}$ ,  $\forall j$  the system takes the form:

$$\frac{1}{h} \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ \vdots \\ U_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_m \end{pmatrix}. \quad (3.40)$$

Note that by division with  $h$  one can achieve a variant of the standard difference method, as in (3.26). The FEM in 2D follows the same procedure, however the basis functions from (3.29) are replaced by “pyramid-functions” on a triangular mesh. For further reading about FEM see [19].

### 3.4.1 Von Neumann Stability Analysis

In 1950 John von Neumann showed the application of Fourier analysis in finite difference schemes for time-dependent PDE's [7]. This stability analysis is used to analyze stability of the diffusion equation  $u_t = du_{xx}$ ,  $t > 0$ ,  $0 < x < 1$  with diffusion constant  $d > 0$  and a periodic boundary condition.

For discretization purpose we let  $U_j^n$  be the numerical approximation to  $u(x_j, t_n)$ . Let  $\Delta t > 0$  be the constant time discretization parameter such that  $t_n = n\Delta t$ ,  $n = 0, 1, 2, \dots$  and let  $h = \frac{1}{m+1}$  be the grid spacing. Combining the first-order forward differencing in time with the second-order central difference from (3.26), the fully discrete scheme becomes:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = d \left( \frac{U_{j-1}^n - 2U_j^n + U_{j+1}^n}{h^2} \right) \quad \text{for } j = 1, \dots, m, \quad n \geq 0, \quad (3.41)$$

where  $U_0^n = U_m^n$ ,  $U_{m+1}^n = U_1^n$  for all  $n$ . Here the errors are linear in time step and quadratic in the space step,  $\delta U^{n+1} = O(\Delta t) + O(h^2)$ .

For the initial value we consider:

$$U_j^0 = e^{2\pi i k x_j}, \quad (3.42)$$

such that  $U_j^0$  is a Fourier mode with wave number  $1 \leq k \leq m$ . We now assume that:

$$U_j^{n+1} = a_k U_j^n, \quad (3.43)$$

which gives us the Fourier mode:

$$U_j^n = (a_k)^n e^{2\pi i k x_j}, \quad n \geq 0. \quad (3.44)$$

To check stability, we insert this solution into our finite difference scheme from (3.41):

$$\begin{aligned} \frac{(a_k)^{n+1} - (a_k)^n}{\Delta t} e^{2\pi i k x_j} &= d(a_k)^n \frac{e^{2\pi i k x_{j-1}} - 2e^{2\pi i k x_j} + e^{2\pi i k x_{j+1}}}{h^2} \\ &= \frac{d(a_k)^n e^{2\pi i k x_j}}{h^2} (e^{-2\pi i k h} - 2 + e^{2\pi i k h}). \end{aligned} \quad (3.45)$$

Hence:

$$(a_k)^{n+1} = (a_k)^n \left( 1 + \frac{\Delta t d}{h^2} (e^{-2\pi i k h} - 2 + e^{2\pi i k h}) \right). \quad (3.46)$$

For stability we need to consider the amplification factor  $R = \left| \frac{(a_k)^{n+1}}{(a_k)^n} \right|$ . Thus from (3.46) we observe that:

$$\begin{aligned} \frac{(a_k)^{n+1}}{(a_k)^n} &= 1 + \frac{\Delta t d}{h^2} (e^{-2\pi i k h} - 2 + e^{2\pi i k h}) \\ &= 1 - \frac{4\Delta t d}{h^2} \sin^2(\pi k h), \end{aligned} \quad (3.47)$$

since from Euler's formula  $e^{-2\pi i k h} + e^{2\pi i k h} = 2 \cos(2\pi k h)$ . However, from the double angle formula from trigonometry we recall that  $\cos(2\pi k h) - 1 = -2 \sin^2(\pi k h)$ .

For stability we need  $R$  to be less than or equal to one. Thus imposing:

$$\left| 1 - \frac{4\Delta t d}{h^2} \sin^2(\pi k h) \right| \leq 1 \quad \text{for all } k = 1, 2, \dots, m, \quad (3.48)$$

arriving at the stability condition:

$$\frac{d\Delta t}{h^2} \leq \frac{1}{2} \quad (3.49)$$

since  $0 \leq \sin^2(\pi k h) \leq 1$ .

### Implicit Finite Difference Method

Above, we discussed the standard forward difference for the diffusion equation  $u_t = du_{xx}$ , but instead we could have used the backward difference, which at time  $n+1$  gives the following approximation:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j-1}^{n+1} - 2U_j^{n+1} + U_{j+1}^{n+1}}{h^2} \quad \text{for } j = 1, \dots, m, \quad n \geq 0, \quad (3.50)$$

where  $U_0^n = U_m^n$ ,  $U_{m+1}^n = U_1^n$  for all  $n$ . By using the same approach as before we can, by use of the Fourier ansatz from (3.44) and the implicit scheme above, derive the amplification factor to be:

$$\frac{(a_k)^{m+1}}{(a_k)^m} = \frac{1}{1 - \frac{d\Delta t}{h^2}(e^{-2\pi i k h} - 2 + e^{2\pi i k h})} \quad (3.51)$$

$$= \frac{1}{1 - \frac{d\Delta t}{h^2}(2 \cos(2\pi k h) - 2)} \quad (3.52)$$

$$= \frac{1}{1 + \frac{d\Delta t}{h^2}4 \sin^2(\pi k h)} . \quad (3.53)$$

From (3.53) we observe that  $\frac{d\Delta t}{h^2}4 \sin^2(k\pi h/2)$  is always positive, thus  $R \leq 1$  and our requirement for stability is always fulfilled. Thus our implicit scheme is unconditionally stable i.e. it is stable for any chosen parameters.

This stability property is very useful enabling us to take arbitrary large time steps. It is important to remember that larger step sizes also provide larger truncation errors and error growth.

### 3.4.2 Stability

Talking about the stability of methods for stiff problems, the most popular approach comes from Dahlquist [10]. It is based on the test equation:

$$\frac{\partial}{\partial t} u = \lambda u , \quad (3.54)$$

for which the general solution is:

$$u(t) = e^{\lambda t} u(0) . \quad (3.55)$$

It is easily seen that if the real part of  $\lambda$  is non-positive, i.e.  $\text{Re}(\lambda) \leq 0$ , then the solution remains bounded by:

$$|u(t)| \leq |u(0)| , \quad (3.56)$$

and

$$|u(t)| \rightarrow 0 \text{ for } t \rightarrow \infty . \quad (3.57)$$

Applying any one step method to (3.54) results in an iteration:

$$U^{n+1} = R(\Delta t \lambda) U^n , \quad (3.58)$$

where  $R(\Delta t \lambda)$  is the amplification factor. The method is called stable if (3.56) holds in the discrete case:

$$|U^{n+1}| \leq |U^n| , \quad (3.59)$$

which is equivalent to:

$$|R(\Delta t \lambda)| \leq 1 , \quad (3.60)$$

as already seen in Section 3.4.1.



Time stepping methods, for which (3.60) holds independent of the stepsize  $\Delta t > 0$  are called A-stable. The additional postulation of sufficient damping of stiff components, where  $\text{Re}(\lambda) \ll 0$ , means:

$$R(\Delta t \lambda) \rightarrow 0 \text{ for } \text{Re}(\Delta t \lambda) \rightarrow -\infty .$$

For implicit Runge-Kutta methods, the amplification factor  $R(z)$  is a rational function:

$$R(z) = \frac{P(z)}{Q(z)} , \quad (3.61)$$

where  $P$  and  $Q$  are polynomials of degree less than or equal to the number of Runge-Kutta stages. For a rational function it holds:

$$\lim_{z \rightarrow -\infty} R(z) = \lim_{z \rightarrow \infty} R(z) = \lim_{y \rightarrow \infty} R(iy) , z, y \in \mathbb{R} .$$

This motivates the definition of L-stability.

**Definition 3.4.1.** *A method is called L-stable if it is A-stable and if in addition*

$$\lim_{z \rightarrow \infty} R(z) = 0 . \quad (3.62)$$

L-stable methods are relevant for reaction-diffusion models, like the one that is presented in this report, because diffusive problems typically are stiff. Examples of L-stable methods are: Implicit Euler and the Singly Diagonal Implicit Runge-Kutta (SDIRK) method. For the initial value problem  $u_t = f(u)$  ,  $u(0) = u^0$ , the SDIRK method is:

$$\begin{aligned} k_1 &= f(t_n + \gamma \Delta t, U^n + \gamma \Delta t k_1) \\ k_2 &= f(t_n + (1 - \gamma) \Delta t, U^n + \Delta t((1 - 2\gamma)k_1 + \gamma k_2)) \\ U^{n+1} &= U^n + \Delta t(k_1 + k_2)/2 . \end{aligned}$$

The method is of order 3 with  $\gamma = (3 \pm \sqrt{3})/6$ . The stability function reads

$$R(z) = \frac{1 + z(1 - 2\gamma) + z^2(1/2 - 2\gamma + \gamma^2)}{(1 - \gamma z)^2} ,$$

and the method is A-stable for any  $\gamma \geq 1/4$ , especially for  $\gamma = (3 + \sqrt{3})/6$ .

Setting  $\gamma = (2 \pm \sqrt{2})/2$ , we find:

$$\lim_{z \rightarrow \infty} R(z) = \lim_{z \rightarrow \infty} \frac{1 + z(1 - 2\gamma)}{(1 - \gamma z)^2} = 0 ,$$

and the method is also L-stable. The order drops to 2, however. [16]

### 3.5 Positivity and Maximum Principle

One of the most famous second order elliptic equations is the stationary heat equation:

$$-\Delta u = f . \quad (3.63)$$

An interesting result from this well-studied area is, that given positive Dirichlet boundary conditions, a positive source term  $f$  implies that the solution becomes positive too. This positivity preserving property is well-known, and called the Maximum Principle. This result holds in general for second order elliptic problems [13, 26].

To illustrate the maximum principle for the stationary heat equation, let us consider a slightly expanded version but in one dimension only:

$$\begin{aligned} -u''(x) + c(x)u(x) &= f(x) , & x \in (0, L) \\ u(0) &= A , \\ u(L) &= B . \end{aligned} \tag{3.64}$$

Here  $f$  and  $c$  are functions defined for  $x \in (0, L)$  and  $A, B$  are constants, which define the values that the solution takes at the boundary. As mentioned earlier, this condition is known as the Dirichlet Boundary Condition.

**Theorem 3.5.1.** *If  $c$  is a continuous, non-negative function, then problem (3.64) has at most one solution of class  $C^2([0, L])$ .*

*Proof.* By proof of contradiction we now assume that both  $u_1$  and  $u_2$  are solutions to (3.64) of class  $C^2([0, L])$ . Now let  $w = u_2 - u_1$ . It can easily be demonstrated that  $w$  solves the homogeneous boundary value problem:

$$\begin{aligned} -w'' + cw &= 0 , & x \in (0, L) \\ w(0) &= w(L) = 0 . \end{aligned}$$

Now multiply by  $w$  and integrate over  $(0, L)$ :

$$-\int_0^L w''w \, dx + \int_0^L cw^2 \, dx = 0 .$$

Now from integration by parts, we obtain:

$$\int_0^L (w')^2 \, dx - [w'w]_0^L + \int_0^L cw^2 \, dx = 0 .$$

Note that  $[w'w]_0^L = 0$  from the boundary conditions. Since  $c$  is continuous and non-negative the integrands inside the integrals become non-negative. By gathering the two integrals into one, we observe that the integral should be zero, thus  $w' = 0$  which implies that  $w = 0$  for all  $x$  because of the homogenous boundary values. Hence  $u_1 = u_2$ , which shows uniqueness and thereby proves the theorem.  $\square$

As mentioned earlier one important property of (3.64) is the positivity preserving property, called the Maximum Principle:

**Theorem 3.5.2.** *Assume that  $c$  is continuous, non-negative and that (3.64) has a solution  $u$  of class  $C^2$ . If  $f$  is non-negative for  $x \in (0, L)$ ,  $A \geq 0$  and  $B \geq 0$ , then the solution  $u$  becomes non-negative in  $[0, L]$ .*

*Proof.* By proof of contradiction, we now assume that  $u(x_0) < 0$  for an arbitrary point  $x_0$ . Since  $u(0) = A$  and  $u(L) = B$ , where  $A, B \geq 0$ , it follows that  $x_0$

lies inside the interval  $(0, L)$ . By assumption  $u$  is continuous, and thus by the intermediate value theorem there exists an interval  $[a, b]$  where  $u \leq 0$ , so that  $x_0 \in [a, b] \subset [0, L]$  and  $u(a) = u(b) = 0$ . Since  $u \leq 0$ ,  $c \geq 0$  and  $f \geq 0$  in the interval  $[a, b]$  we have:

$$u'' = cu - f \leq 0 .$$

In this way  $u$  is concave on  $[a, b]$ . From the definition of concave functions we recall that:

$$u(\lambda a + (1 - \lambda)b) \geq \lambda u(a) + (1 - \lambda)u(b) , \quad \lambda \in [0, 1] .$$

Note that the right hand side becomes zero. Now let  $x_0 = \lambda a + (1 - \lambda)b$  thus  $u(x_0) \geq 0$ , which gives the contradiction.  $\square$

Theorem 3.5.2 is in fact a monotonicity result, which carries over to finite difference discretization. The solution operator  $S : (f, A, B) \mapsto u(t, x)$  is monotone in the sense that it maps non-negative data  $(f, A, B)$  into a non-negative solution. The PDE operator

$$PDE(u) := \begin{pmatrix} -u'' + cu \\ u(0) \\ u(L) \end{pmatrix} , \quad (3.65)$$

is called inverse-monotone (see Definition 5.1.1) as its inverse  $PDE^{-1} = S$  is monotone. In Chapter 5 the PDE operator is replaced by a finite difference operator given as matrix (3.26). Lemma 5.1.2 says, that the finite difference operator inherits its inverse monotonicity from (3.65).

## Chapter 4

# A Reaction-Diffusion Model for FLIP in realistic geometry

In Section 3.1.1 we introduced the Reaction-Diffusion Equation, see (3.13). We will use this model to simulate the transport/diffusive process inside the cell.

Living cells are in continuous exchange with their environment which causes a lot of protein and lipid transport through the cell. Combining this with the intracellular traffic, such as Brownian motion and anomalous diffusion of proteins in the cytoplasm, active transport of vesicular shuttling, fission and fusion of membrane tubules constitute a very complex intracellular traffic network. To describe this complexity mathematically, the system has to be reduced to its most essential elements. The Diffusion Equation from (3.6) accounts for Brownian diffusion. Recently, indication of anomalous diffusion have been found for single-particle movements in cytoplasm. For further reading and other models see [27]. Recent work and observations from FLIP experiments have discovered a hierarchically organized architecture of both cytoplasm and nucleus. As a consequence the mathematical model may need space-dependent diffusion constants and of course real cell shapes for a realistic model.

### 4.1 Chan-Vese Segmentation of FLIP Images

As discussed above an important step towards a reliable FLIP simulation is to specify the geometry of any given cell.

By Chan-Vese's algorithm [6], which automatically detect the borders, the FLIP image can be segmented into background and foreground that may include the cell with its irregular borders, the nucleus or the bleaching area. Next we briefly introduce and discuss the basic idea behind the Chan-Vese algorithm, but for more reading we refer the reader to [6] and [15].

Image segmentation is a huge topic and several methods have been developed for this purpose. One of the strengths of Chan-Vese's algorithm is that it can distinguish several objects in the image whereas, for instance, the Snake method [20] works more like a rubber band which only spans one object. Unlike the Snake method, the Chan-Vese method is able to detect objects whose boundaries are not necessarily defined by gradients. Both methods minimize a fitting energy

until it reaches the desired boundary, but in the Chan-Vese method the stopping term will not be dependent on the gradients.

The result of the segmentation is a level set function  $L$ , where the zero level  $\Gamma = \{x \in \mathbb{R}^2 | L(x) = 0\}$  is the closed contour which describes the segmented area where  $L < 0$  inside  $\Gamma$  and  $L > 0$  outside  $\Gamma$ . Now let  $\mathcal{I}(x)$  be the grayscale image,  $c_-$  and  $c_+$  the average inside and outside  $\Gamma$  respectively. With this in mind, let us now consider the Chan-Vese fitting energy:

$$\begin{aligned} E(c_-, c_+, \Gamma) &= F_0(\Gamma) + F_1(\Gamma) + F_2(\Gamma) \\ &= \mu \int_{\Gamma} d\sigma + \lambda_1 \int_{L < 0} (\mathcal{I}(x) - c_-)^2 dx + \lambda_2 \int_{L > 0} (\mathcal{I}(x) - c_+)^2 dx, \end{aligned}$$

where  $\mu \geq 0$ ,  $\lambda_1, \lambda_2 > 0$  are fixed parameters.  $F_0(\Gamma)$  considers the length of  $\Gamma$  and thereby the smoothness of the contour. For the time being, let us concentrate on  $F_1$  and  $F_2$ , since they are the most essential parts of the fitting energy. For instance, consider an image with a black object on a white background. The fitting energy will only be minimal when  $\Gamma$  lies on the edge of the object, see Fig. 4.1.

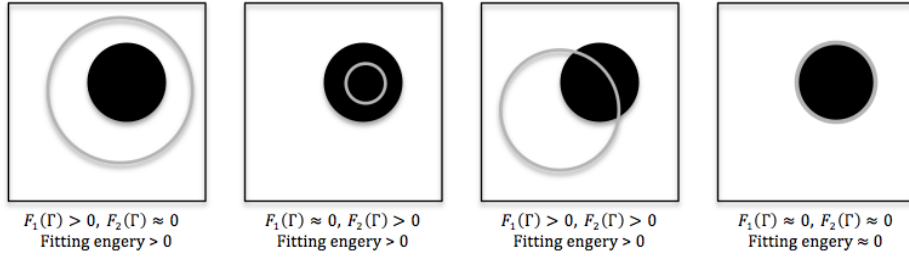


Figure 4.1: Four different fitting cases with  $\mu = 0$ . In the first case the area inside  $\Gamma$  contains the object and some of the background. This gives a wide span of pixels with different values, as a result  $F_1(\Gamma) > 0$  and  $F_2(\Gamma) \approx 0$ , since we expect the pixels outside  $\Gamma$  to have an approximately uniform value, the difference between each pixel value and the average  $c_+$  would be  $\approx 0$ . Opposite, if  $\Gamma$  is inside the object, then  $F_1(\Gamma) \approx 0$  and  $F_2(\Gamma) > 0$ . Only in the case where  $\Gamma$  lies on the edge of the object the energy would be minimal.

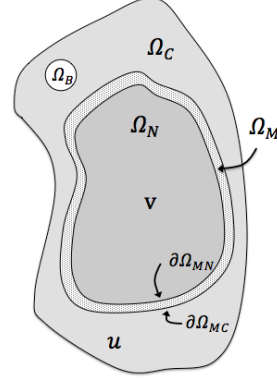
By keeping  $c_-$  and  $c_+$  fixed and minimize the Chan-Vese fitting energy with respect to  $L$ , it is by use of the Euler-Lagrange equation for  $L$  we are able to obtain the following PDE; see Appendix A, p. 11 or [6, 15],

$$L_t = \delta_\varepsilon(L) \left[ \mu \operatorname{div} \left( \frac{\nabla L}{|\nabla L|} \right) + \lambda_1 (\mathcal{I} - c_-)^2 - \lambda_2 (\mathcal{I} - c_+)^2 \right]. \quad (4.1)$$

Here we introduced an artificial time parameter to make an active contour. Thus  $\Gamma(t) = \{x(t) \in \mathbb{R}^2 : L(x(t), t) = 0\}$  would evolve to obtain a minimal energy. Generally we fix  $\lambda_1$  and  $\lambda_2$  to one, but if the image contains a more uniform colored object than the background, then  $\lambda_1$  should be higher than  $\lambda_2$ , and reverse if the background is more uniform colored than the object. In the FLIP

images the cell is represented on a black background therefore we set  $\lambda_2 > \lambda_1$ . The parameter  $\mu$  concerns the length of the curve, and is therefore set high to obtain a smooth boundary.

Consequently by use of Chan-Vese segmentation, the cell from a respective FLIP image is segmented into cytoplasm  $\Omega_C$ , nucleus  $\Omega_N$  and the nuclear membrane  $\Omega_M$ . Let  $\Omega = \Omega_C \cup \Omega_N \cup \Omega_M$  be the domain, which covers the full complexity of the living cell under observation, such that  $\Omega$  is simply-connected. The membrane  $\Omega_M$  needs to have a small but finite thickness, in which the transport process between cytoplasm and nucleus takes place. Let  $\partial\Omega_M$  denote the boundary of the nuclear membrane and for later reference we split  $\partial\Omega_M$  into  $\partial\Omega_{MC}$  and  $\partial\Omega_{MN}$  which are the boundaries that face cytoplasm and nucleus respectively. Furthermore, we denote  $\Omega_B$  as the bleaching area, generally  $\Omega_B \subset \Omega_C$ .



The membrane  $\Omega_M$  with a small but finite thickness cannot be constructed by use of Chan-Vese alone. A discussion on how it is made, can be found in section 6.1.

## 4.2 A Membrane Transport Model

In the few mathematical models that can be found for intracellular transport, it is possible to observe that the models used to describe the transport process in general are indirect transport models, i.e. models that are pure diffusion models with various diffusion constants [5, 29]. Often the diffusivity is assumed to be large in both nucleus and cytoplasm, but small in the neighborhood of the nucleus membrane. The consequence of this model is that the concentration inside nucleus and cytoplasm over time will become equal. By looking into the FLIP image from Figure 2.3(a) and assuming that the fluorescence intensity is direct proportional to the eGFP concentration, we observe that the intensity and thereby the concentration is much higher inside nucleus than in cytoplasm. This is inconsistent with the usual models.

As discussed earlier it has been shown, that eGFP is limited to transport through the nuclear pore complex. Nuclear pore complexes are often described as a small passage which allows transport via diffusion through the pore complexes. It is still unknown exactly what the nuclear pore complex looks like, and how the transport through them works [8, 25]; but the higher amount of fluorescence from eGFP inside nucleus on the FLIP images cannot only be caused by diffusion. It therefore needs to be a consequence of some type of transport mechanism.

To present a model allowing transport across the membrane, we now distinguish between the concentrations inside cytoplasm  $u$  and nucleus  $v$ , let  $u$  and  $v$  be defined on  $\Omega$  such that:

$$\begin{aligned} u(t, x) &= 0 \text{ for } x \in \Omega_N, \\ v(t, x) &= 0 \text{ for } x \in \Omega_C. \end{aligned} \tag{4.2}$$

Consequently,  $c = u + v$  denotes the concentration of eGFP in  $\Omega$ . Here the transport across the membrane is modeled as a direct transport process by the reversible chemical reaction equation:



where  $k^+$  and  $k^-$  are positive reaction rates. From Section 3.2 the dynamics of (4.3) are represented by the following differential system:

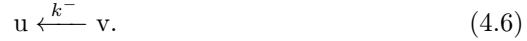
$$\begin{aligned} u_t &= k^- v - k^+ u, \\ v_t &= k^+ u - k^- v, \quad x \in \Omega_M, t > 0. \end{aligned} \quad (4.4)$$

Obviously,  $u_t + v_t = 0$  and thus  $c = u + v$  is constant over time, which shows that the law of mass conservation is fulfilled. In an effort to present a realistic model the transport process from the above differential system (4.4) would only appear inside the membrane domain.

To guarantee the complete transformation of  $u$  into  $v$  across the membrane, space dependent reaction rates are needed. For (4.2) to hold true, the reaction:



should dominate near nucleus, i.e.  $k^+ \gg 1$ , while  $k^- \ll 1$ , in such a way that  $v$  would dominate near nucleus. On the other side of the membrane, near cytoplasm, the dominating reaction should be:



Consequently  $k^- \gg 1$  and  $k^+ \ll 1$ . As discussed we know that a system is in equilibrium when  $u_t = v_t = 0$ . With this in mind we introduce space dependent reaction rates to force the system (4.4) into equilibrium:

$$k^-(x)v = k^+(x)u. \quad (4.7)$$

By choosing  $k^- = 0$  and  $k^+ \gg 1$  on boundary  $\partial\Omega_{MN}$ , we force  $u = 0$ . Likewise, on the opposite membrane boundary  $\partial\Omega_{MC}$  the reaction rates are chosen so that  $k^+ = 0$  and  $k^- \gg 1$  which thereby implies that  $v = 0$ , so that (4.2) would hold true.

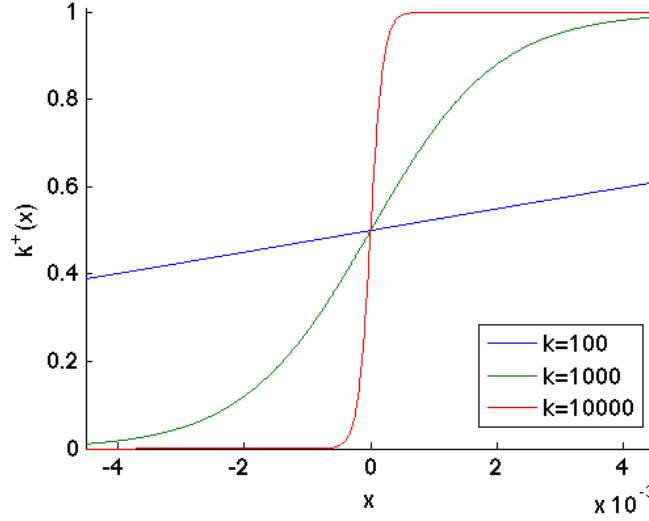
An example of an applicable reaction rate is the logistic function:

$$f(x) = \frac{1}{\epsilon} \frac{1}{1 + e^{-k(x-x_m)}} , \quad (4.8)$$

where  $k$  is growth rate,  $\epsilon$  is scale parameter and  $x_m$  is center for the logistic function. For the two-dimensional case we have:

$$k^+ = \frac{1}{\epsilon} \frac{1}{1 + e^{-k(x-x_m)n}} \quad \text{and} \quad k^- = \frac{1}{\epsilon} \frac{1}{1 + e^{k(x-x_m)n}}. \quad (4.9)$$

For symmetry across the membrane in the multidimensional case, we choose  $x_m$  such that it is the orthogonal projection of  $x$  onto the center of the membrane and thereby  $(x-x_m)n$  is the signed distance, where  $n$  is the outward unit normal. For illustration consider a cross section of a 9nm thick membrane centered in

Figure 4.2: Reaction rate  $k^+$  centered at origin.

origin. Now let  $x$  be measured in  $\mu\text{m}$  such that  $k^+$  from (4.9) with three different growth rate is as illustrated in Figure 4.2.

As discussed above,  $k^+$  must be zero at the boundary facing cytoplasm. From Figure 4.2 it can be observed, that the growth rate  $k$  has to be chosen high enough, in order that  $k^+ \approx 0$  at the boundary facing cytoplasm.

#### 4.2.1 Reaction-Diffusion Model for FLIP Experiments

From [23] we adopt a simple photophysical model of excitation, emission and bleaching. It is assumed that eGFP only contains the ground state  $u_{-1}$  and the excited state  $u$ , such that the excitation from  $u_{-1}$  to  $u$  occurs from absorption of light with rate constants  $k_{ex}$  and light intensity  $I_{ex}$ . The relaxation to  $u_{-1}$  results in either a fluorescence emission or a nonradiative emission with rate constants  $k_r$  or  $k_{nr}$  respectively.

In FLIP experiments, bleaching happens only on a small part of the cell. To model this we need to add an irreversible reaction step with rate constant  $b$  to the excited state:



where  $k_1 = k_{ex}I_{ex}$  and  $k_{-1} = k_{nr} + k_r$ .

Written into a coupled ordinary differential equation system we get:

$$\begin{aligned} \frac{du_{-1}}{dt} &= -k_1 u_{-1} + k_{-1} u \\ \frac{du}{dt} &= k_1 u_{-1} - (k_{-1} + b) u . \end{aligned} \quad (4.11)$$

For simplification, let us assume that the excitation and relaxation between the ground state and excited state is much faster than photobleaching, such that



$k_{1,-1} \gg b$ . Thus focus on

$$u_{-1} \xrightleftharpoons[k_{-1}]{k_1} u, \quad (4.12)$$

with equilibrium constant  $q$ , given by:

$$q = \frac{u}{u_{-1}} = \frac{k_1}{k_{-1}} = \frac{k_{ex} I_{ex}}{k_r + k_{nr}}. \quad (4.13)$$

Such that:

$$\frac{d(u_{-1} + u)}{dt} = \frac{d\left(\frac{u}{q} + u\right)}{dt} = \frac{d\left(\frac{1+q}{q}u\right)}{dt}. \quad (4.14)$$

From (4.11) we observe that  $\frac{d(u_{-1}+u)}{dt} = -bu$ , thus:

$$\frac{du}{dt} = -b \frac{q}{q+1} u. \quad (4.15)$$

This differential equation now describes the photobleaching reaction which occurs in the bleaching area  $\Omega_B$ . As already mentioned,  $b$  is the bleaching rate and  $q$  is the equilibrium constant from (4.13) which also depend on the light intensity  $I_{ex}$  from the laser. Assuming that bleaching only occurs in the bleaching area and eGFP has a diffusive behavior that follows the multidimensional diffusion equation from (3.13), our time-dependent PDE-model with membrane transport as described in (4.4) reads:

$$\begin{aligned} u_t &= \nabla \cdot (\alpha \nabla u) + (k^- v - k^+ u) \Big|_{\Omega_M} - \theta b \frac{q}{1+q} u \Big|_{\Omega_B}, \\ v_t &= \nabla \cdot (\beta \nabla v) + (k^+ u - k^- v) \Big|_{\Omega_M}, \quad x \in \Omega, \quad t > 0, \end{aligned} \quad (4.16)$$

where  $\theta$  is an indicator function, which is one when the bleaching area  $\Omega_B$  is bleached by the high-intensity laser and zero otherwise. Note that the membrane transport is restricted only to occur in the membrane. Since the speed of transport across the membrane is regulated by the diffusion coefficients, they need to take different values inside cytoplasm, membrane and nucleus. Thus, the diffusion coefficients are piecewise defined by:

$$\alpha = \begin{cases} \alpha_C & \text{if } x \in \Omega_C, \\ \alpha_M & \text{if } x \in \Omega_M, \\ \alpha_N & \text{if } x \in \Omega_N. \end{cases} \quad \text{and} \quad \beta = \begin{cases} \beta_C & \text{if } x \in \Omega_C, \\ \beta_M & \text{if } x \in \Omega_M, \\ \beta_N & \text{if } x \in \Omega_N. \end{cases} \quad (4.17)$$

As discussed in Section 3.3, we assume that there would be no exchange between the exterior and the interior of the cell, and therefore the cell membrane would work as a closed wall without any flux. To maintain mass conservation inside the cell, this model comes with the non-flux boundary condition, the Neumann boundary, such that for both  $u$  and  $v$  the boundary conditions become:

$$n \cdot \nabla u = n \cdot \nabla v = 0, \quad x \in \partial\Omega. \quad (4.18)$$

Finally, the initial conditions are given by:

$$u(0, x) = u^0(x), \quad v(0, x) = v^0(x), \quad x \in \Omega. \quad (4.19)$$

## Chapter 5

# Analysis

By definition of chemical reactions and the mass concentration, the concentration  $c$  is the current amount of mass  $m$  divided by the volume  $V$  of the mixture, i.e.  $c = m/V$ , which in SI-units is measured in  $\text{kg}/\text{m}^3$ . Since mass and volume are both non-negative, the concentration will always remain non-negative as well. This, and the fact that negative values in some systems for chemical reactions may lead to instabilities, prompts us to look at our system to check if positivity is preserved.

Besides this we also take a closer look at stability and convergence of the PDE system.

For simplicity, consider the PDE model from (4.16) in one dimension and thus we focus on the membrane transport. With this simplified system and a diffusion coefficient scaled to unity we see that:

$$\begin{aligned} u_t &= u_{xx} + k^-v - k^+u , \\ v_t &= v_{xx} + k^+u - k^-v , \quad x \in (0,1) , \quad t > 0 . \end{aligned} \quad (5.1)$$

For illustration purposes we consider a model which takes homogeneous Dirichlet boundary condition for  $u$  and Neumann condition for  $v$ :

$$\begin{aligned} u(t,0) &= u(t,1) = 0 , \\ v_x(t,0) &= v_x(t,1) = 0 , \end{aligned} \quad (5.2)$$

with initial data as in (4.19).

The combination of both Dirichlet and Neumann conditions makes the proof for the positivity preserving property more general, however, the Dirichlet condition can easily be replaced by a Neumann condition to achieve the same boundary conditions as in (4.18). The analysis uses elements by E. Bohl [3] and takes-off from the discretization of the space derivative.

For simplicity and stability reasons we now use the backward Euler method to achieve the time-discrete model. The backward Euler method can be seen as a result from the truncated Taylor series, such that the time derivative can be approximated by:

$$\frac{\partial u^{n+1}}{\partial t} \approx \frac{u^{n+1} - u^n}{\Delta t} , \quad (5.3)$$

where  $\Delta t$  is the time discretization parameter, such that at time  $t_n = n\Delta t$  for integers  $n \geq 0$ . Contrary to previous sections let  $u^n$  denote an approximation of  $u(t_n)$ . Thus the coupled semi-implicit system becomes:

$$\begin{aligned}\frac{u^{n+1} - u^n}{\Delta t} &= u_{xx}^{n+1} + k^- v^n - k^+ u^{n+1}, \\ \frac{v^{n+1} - v^n}{\Delta t} &= v_{xx}^{n+1} + k^+ u^n - k^- v^{n+1}.\end{aligned}\quad (5.4)$$

To achieve a full discrete model, we now discretize the space derivative by a finite difference approximation. To approximate the double derivative on the inner grid, we use the three point centered difference formula:

$$\frac{\partial^2 u_i^n}{\partial x^2} \approx \frac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{h^2} \quad (5.5)$$

where  $u_i^n \approx u(t_n, x_i)$  and  $h > 0$  such that  $x_i = ih$ ,  $i = 1, 2, \dots, m$  and  $x_{m+1} = (m+1)h = 1$ .

Thus recall from Section 3.4 that the linear operator for homogeneous Dirichlet condition with defined boundaries from (5.2) becomes:

$$-u_{xx} \approx A_h u = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix} u, \quad (5.6)$$

where  $u$  is the vector which holds the approximations on the inner grid points.

Next consider the Neumann boundary condition and observe that  $v_0 = v_1$  and  $v_m = v_{m+1}$  are consequences of the Neumann boundary condition from (5.2). Consequently, the second derivative for  $v$  on the inner grid points is approximated by:

$$-v_{xx} \approx B_h v = \frac{1}{h^2} \begin{pmatrix} 1 & -1 & & \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix} v. \quad (5.7)$$

Thus we can rewrite our system from (5.4) into a fully discrete system:

$$\begin{aligned}\frac{u^{n+1} - u^n}{\Delta t} &= -A_h u^{n+1} + k^- v^n - k^+ u^{n+1}, \\ \frac{v^{n+1} - v^n}{\Delta t} &= -B_h v^{n+1} + k^+ u^n - k^- v^{n+1}.\end{aligned}\quad (5.8)$$

By simple linear algebra we find that:

$$\begin{aligned}[(1 + \Delta t k^+)I + \Delta t A_h]u^{n+1} &= u^n + \Delta t k^- v^n, \\ [(1 + \Delta t k^-)I + \Delta t B_h]v^{n+1} &= v^n + \Delta t k^+ u^n,\end{aligned}\quad (5.9)$$

which we will refer to as our linearly coupled discrete system, with positive step sizes  $\Delta t, h > 0$ , non-negative reaction rates  $k^\pm \geq 0$  and  $I$  being the identity matrix.

## 5.1 Positivity Preserving Property for a Reaction-Diffusion Model

The key argument to prove the positivity preserving property for our discrete system in (5.9), lies in the fact that the finite difference operators  $(1 + \Delta tk^+)I + \Delta tA_h$  and  $(1 + \Delta tk^-)I + \Delta tB_h$  are inverse-monotone with bounded inverse.

To make an unambiguous statement, now consider the definition of inverse-monotonicity [4]. First note that we use the notation  $\mathcal{L}(V)$  for the set of all linear operators on a normed space  $V$ . Whereas the set of all linear monotone operators on  $V$  is denoted by  $\mathcal{L}_+(V)$ .

**Definition 5.1.1.**  *$A \in \mathcal{L}(V)$  is called inverse-monotone if  $A$  has an inverse in  $\mathcal{L}_+(V)$ .*

Thus a matrix  $A$  is inverse-monotone if it is invertible with elementwise non-negative inverse  $A^{-1} \geq 0$ . Consequently, if  $Ae \geq 0$  then  $e \geq 0$ , thus solving the system preserves positivity. In fact  $A_h$  inherits its inverse-monotonicity from the negative Laplacian (see Section 3.5), while  $B_h$  is singular.

**Lemma 5.1.2.** *With non-negative parameters  $\Delta t \geq 0$ ,  $k^\pm \geq 0$ , and  $h > 0$  both system matrices  $(1 + \Delta tk^+)I + \Delta tA_h$  and  $(1 + \Delta tk^-)I + \Delta tB_h$  are regular with non-negative and uniformly bounded inverse.*

As already discussed, an obvious consequence is:

**Corollary 5.1.3.** *With non-negative initial data  $u^0, v^0 \geq 0$  and reaction rates  $k^\pm \geq 0$ , the solutions and thereby approximations computed by (5.9) remain non-negative for all time  $t_n = n\Delta t \geq 0$ .*

The proof of Lemma 5.1.2 is in fact based on inverse-monotone  $Z$ -matrices. The class of  $Z$ -matrices are matrices whose off-diagonal elements are non-positive:

$$Z = (z_{ij}) , \quad z_{ij} \leq 0 , \quad i \neq j.$$

A related class of matrices is the  $M$ -matrices. Here an inverse-monotone  $Z$ -matrix is called a  $M$ -Matrix.

The weighted max-norm is defined for any positive vector  $p > 0$  as:

$$\|x\|_p = \max_i \left\{ \frac{|x_i|}{p_i} \right\} .$$

Let  $A$  be an elementwise non-negative matrix (from now on called monotone matrix) thus the corresponding operator norm becomes  $\|A\|_p = \|Ap\|_p$ .

**Lemma 5.1.4.** *For  $0 < p \in \mathbb{R}^n$  and  $0 \leq A \in \mathbb{R}^{n \times n}$ ,*

$$\|A\|_p = \|Ap\|_p . \quad (5.10)$$

*Proof of Lemma 5.1.4.* This proof comes in two parts, the first part proves  $\|A\|_p \leq \|Ap\|_p$ .

Let  $x \in \mathbb{R}^n$  be arbitrary. Since  $A \geq 0$ ,

$$\begin{aligned} \frac{|(Ax)_i|}{p_i} &\leq \frac{\sum_j a_{ij} |x_j|}{p_i} \\ &= \sum_j \left( \frac{a_{ij} p_j}{p_i} \frac{|x_j|}{p_j} \right) \\ &\leq \sum_j \left( \frac{a_{ij} p_j}{p_i} \right) \|x\|_p \\ &= \frac{(Ap)_i}{p_i} \|x\|_p \end{aligned}$$

Hence,

$$\|Ax\|_p = \max_i \frac{|(Ax)_i|}{p_i} \leq \max_i \frac{(Ap)_i}{p_i} \|x\|_p .$$

By positivity  $Ap \geq 0$  now observe:

$$\|Ax\|_p \leq \|Ap\|_p \|x\|_p . \quad (5.11)$$

From definition of operator norms recall that:

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} .$$

Thus  $\|A\|_p$  is the smallest constant for which the following inequality holds:

$$\|Ax\|_p \leq \|A\|_p \|x\|_p . \quad (5.12)$$

From (5.11) and (5.12) now observe:

$$\|Ax\|_p \leq \|A\|_p \|x\|_p \leq \|Ap\|_p \|x\|_p ,$$

which shows:

$$\|A\|_p \leq \|Ap\|_p .$$

To complete the proof we now have to show that the reversed inequality  $\|A\|_p \geq \|Ap\|_p$  holds. The proof is trivial and follows from the definition of operator norms, observe that:

$$\|Ap\|_p \leq \|A\|_p \|p\|_p .$$

By definition  $\|p\|_p = 1$ , which completes the proof.  $\square$

**Lemma 5.1.5.** *A Z-matrix  $A$  is M-matrix if and only if  $A$  is semi-positive. That is, there exists  $e > 0$  with  $Ae > 0$ .*

*Proof of Lemma 5.1.5.* We first prove the forward direction.

1) Let  $A$  be a M-matrix. Now let  $\xi = (1, \dots, 1)^T$  be the vector consisting of just ones. Let  $e = A^{-1}\xi$ , since  $A^{-1} \geq 0$  is invertible it cannot have a complete row consisting of just zeros, therefore  $e = A^{-1}\xi > 0$ . Obviously  $Ae = \xi > 0$ .

2) Let  $e > 0$  and  $Ae > 0$ . As  $A$  is  $Z$ -matrix with non-positive off-diagonal elements:

$$(Ae)_i = a_{ii}e_i - \sum_{j \neq i} |a_{ij}|e_j > 0 ,$$

hence  $a_{ii} > 0$ , consequently  $D = \text{diag}(a_{ii})$  is regular and:

$$P = D^{-1}(D - A) \geq 0 , \quad A = D(I - P) .$$

Since  $a_{ii} > 0$  and  $Ae > 0$  observe that

$$(I - P)e = D^{-1}Ae > 0 ,$$

hence

$$Pe < e .$$

Furthermore, by (5.10)

$$\|P\|_e = \|Pe\|_e < 1 .$$

Now let  $(\mathbb{R}^n, d)$  be a metric space, with  $d(x, y) = \|x - y\|$ . Recall that  $P$  is a map  $P : \mathbb{R}^n \rightarrow \mathbb{R}^n$  and observe that

$$\|Px - Py\| \leq \|P\|\|x - y\| < \|x - y\| ,$$

hence  $P$  is a contraction mapping on  $\mathbb{R}^n$ . Thus by Banach fixed point theorem  $P$  admits a unique fixed point  $x \in \mathbb{R}^n$ , i.e  $Px = x$ . Thus,

$$0 = (I - P)x . \quad (5.13)$$

Since  $x \in \mathbb{R}^n$  is unique we see that  $(I - P)$  is regular and  $(I - P)^{-1}$  exists.

Next consider the first  $n$  terms in the series  $\sum_{j=1}^{\infty} P^{j-1}$  and let

$$S_n = I + P + P^2 + \dots + P^{n-1} ,$$

then

$$P \cdot S_n = P + P^2 + P^3 + \dots + P^n .$$

Thus

$$S_n - P \cdot S_n = I - P^n$$

Now let  $n$  go to infinity, thus by rewriting we get

$$(I - P) \lim_{n \rightarrow \infty} S_n = I - \lim_{n \rightarrow \infty} P^n .$$

Since  $\|P\| < 1$ ,  $\lim_{n \rightarrow \infty} P^n$  converges to zero and  $\lim_{n \rightarrow \infty} S_n = \sum_{j=0}^{\infty} P^j$ , thus

$$(I - P) \cdot \sum_{j=0}^{\infty} P^j = I .$$

From (5.13) we know that  $(I - P)$  is regular and thus

$$(I - P)^{-1} = \sum_{j=0}^{\infty} P^j \geq 0 . \quad (5.14)$$

Therefore,

$$A^{-1} = (I - P)^{-1} D^{-1} = \sum_{j=0}^{\infty} P^j D^{-1} \geq 0 . \quad (5.15)$$

Thus  $A$  is inverse-monotone and the proof is complete.  $\square$

**Lemma 5.1.6.** *Let  $A$  be  $M$ -matrix and  $D = \text{diag} \geq 0$ . Then  $A + D$  is  $M$ -matrix.*

*Proof of Lemma 5.1.6.* From Lemma 5.1.5 we recall that since  $A$  is a  $M$ -matrix, there exists an element  $e > 0$  such that  $Ae > 0$ . As  $D$  is diagonal,  $A + D$  is  $Z$ -matrix. Since  $D \geq 0$  and from definition of  $Z$ -matrices  $(A + D)e > De \geq 0$ . Thus by Lemma 5.1.5,  $A + D$  is a  $M$ -matrix.  $\square$

Now we are ready to prove Lemma 5.1.2

*Proof of Lemma 5.1.2.* Once again let  $\xi = (1, \dots, 1)^T$  and note that  $A_h \xi = (1, 0, \dots, 0, 1)^T / h^2$ ,  $B_h \xi = 0$ . As mentioned earlier  $B_h$  is singular. Also note that both  $(I + \Delta t A_h)$  and  $(I + \Delta t B_h)$  are  $Z$ -matrices and both are semi-positive:

$$\begin{aligned} (I + \Delta t A_h) \xi &\geq \xi > 0, \\ (I + \Delta t B_h) \xi &\geq \xi > 0. \end{aligned}$$

Thus by Lemma 5.1.5  $(I + \Delta t A_h)$  and  $(I + \Delta t B_h)$  are  $M$ -matrices. From Lemma 5.1.6 we know, that we can add a positive diagonal matrix  $\Delta t k I$  such that  $((1 + \Delta t k^+)I + \Delta t A_h)$  and  $((1 + \Delta t k^-)I + \Delta t B_h)$ , which are in fact our system matrices, both are  $M$ -matrices, with non-constant and non-negative reaction rates  $k^\pm$ . The fact that our system matrices are  $M$ -matrices proves, that both matrices are regular and inverse-monotone. Thus it only remains to show the uniform bound of the inverse. Now observe that:

$$\begin{aligned} [(1 + \Delta t k^+)I + \Delta t A_h] \xi &\geq \xi \quad \Leftrightarrow \quad [(1 + \Delta t k^+)I + \Delta t A_h]^{-1} \xi \leq \xi, \\ [(1 + \Delta t k^-)I + \Delta t B_h] \xi &\geq \xi \quad \Leftrightarrow \quad [(1 + \Delta t k^-)I + \Delta t B_h]^{-1} \xi \leq \xi. \end{aligned}$$

Thus the norms becomes:

$$\begin{aligned} \|[(1 + \Delta t k^+)I + \Delta t A_h]^{-1} \xi\|_\xi &\leq \|\xi\|_\xi, \\ \|[(1 + \Delta t k^-)I + \Delta t B_h]^{-1} \xi\|_\xi &\leq \|\xi\|_\xi. \end{aligned}$$

Applying Lemma 2 on the left hand side, the desired bound follows:

$$\begin{aligned} \|[(1 + \Delta t k^+)I + \Delta t A_h]^{-1}\|_\xi &\leq 1, \\ \|[(1 + \Delta t k^-)I + \Delta t B_h]^{-1}\|_\xi &\leq 1, \end{aligned} \tag{5.16}$$

in this way the proof is complete.  $\square$

## 5.2 Stability and Convergence

Stability is a condition on the numerical solution, which states that all errors must remain bounded when the iteration process advances. Thus the error has to remain bounded independent of  $\Delta t$  and  $h$ , when the number of time step  $n$  goes to infinity. From Lax Equivalence Theorem it is known, that the a consistent linear finite difference scheme is convergent if and only if it is stable. Stability is therefore a necessary condition for accuracy. [18]

It can be shown that the recently proved uniform bound of the inverse system matrix implies stability of the numerical method with respect to errors in the initial data  $\delta u^0$  and  $\delta v^0$ . Where  $\delta u^n$  is the forward error of our model, such that

the global error  $\delta u^n = u^n - \hat{u}$  is the difference between the numerical result  $u^n$  and the true solution  $\hat{u}(x, t_n)$  with  $\hat{u}(x, 0) = \hat{u}^0$  as initial condition. By linearity, these errors are governed by the system (5.9) itself

$$\begin{aligned} [(1 + \Delta t k^+)I + \Delta t A_h] \delta u^{n+1} &= \delta u^n + \Delta t k^- \delta v^n, \\ [(1 + \Delta t k^-)I + \Delta t B_h] \delta v^{n+1} &= \delta v^n + \Delta t k^+ \delta u^n. \end{aligned} \quad (5.17)$$

Uniform boundedness from (5.16) implies:

$$\begin{aligned} \|\delta u^{n+1}\|_\xi &\leq \|\delta u^n\|_\xi + \Delta t k^- \|\delta v^n\|_\xi, \\ \|\delta v^{n+1}\|_\xi &\leq \|\delta v^n\|_\xi + \Delta t k^+ \|\delta u^n\|_\xi, \end{aligned} \quad (5.18)$$

thus the sum of the errors is bounded by:

$$\|\delta u^{n+1}\|_\xi + \|\delta v^{n+1}\|_\xi \leq (1 + \Delta t K) (\|\delta u^n\|_\xi + \|\delta v^n\|_\xi), \quad (5.19)$$

where  $K = \max(k^+, k^-)$ .

Finally,

$$\|\delta u^n\|_\xi + \|\delta v^n\|_\xi \leq (1 + \Delta t K)^n (\|\delta u^0\|_\xi + \|\delta v^0\|_\xi), \quad (5.20)$$

with  $(1 + \Delta t K)^n \leq \exp(n \Delta t K) \leq \exp(TK)$ ,  $n \Delta t \leq T$ .

The inequality (5.20) expresses stability in the sense that errors at later time depend continuously on initial errors.

Next we show that convergence towards a smooth solution of the reaction-diffusion system (5.1) follows from the discrete Gronwall lemma. Once again, let  $\hat{u}$  and  $\hat{v}$  be the true solution to the continuous problem (5.1), thus the semi-discrete problem takes the form:

$$\begin{aligned} \hat{u}_t &= -A_h \hat{u} + k^- \hat{v} - k^+ \hat{u} + O(h^2), \\ \hat{v}_t &= -B_h \hat{v} + k^+ \hat{u} - k^- \hat{v} + O(h^2), \end{aligned} \quad (5.21)$$

where  $O(h^2)$  is the error term, which derives from the discretization of  $\Delta u$  and  $\Delta v$ . From Taylor expansion we note that:

$$\hat{u}(t_n) = \hat{u}(t_{n+1}) - \Delta t \hat{u}_t(t_{n+1}) - \frac{\Delta t^2}{2} \hat{u}_{tt}(t_{n+1}) + O(\Delta t^3). \quad (5.22)$$

By rearranging the Taylor expansion of  $\hat{u}$  and combining it with the semi-discrete problem from (5.21), we see that:

$$\hat{u}(t_{n+1}) = \hat{u}(t_n) + \Delta t (-A_h \hat{u}(t_{n+1}) + k^- \hat{v}(t_{n+1}) - k^+ \hat{u}(t_{n+1}) + O(h^2)) + O(\Delta t^2). \quad (5.23)$$

Recall from Backward Euler that  $\hat{v}(t_{n+1}) = \hat{v}(t_n) + O(\Delta t)$ , which results in:

$$[(1 + \Delta t k^+)I + \Delta t A_h] \hat{u}(t_{n+1}) = \hat{u}(t_n) + \Delta t k^- \hat{v}(t_n) + O(\Delta t(\Delta t + h^2)). \quad (5.24)$$

Consequently, the global error, which is the difference between the discrete solution from (5.9) and the true solution from (5.24), can be expressed as:

$$[(1 + \Delta t k^+)I + \Delta t A_h] \delta u^{n+1} = \delta u^n + \Delta t k^- \delta v^n + O(\Delta t(\Delta t + h^2)). \quad (5.25)$$



In similar way, we can find:

$$[(1 + \Delta t k^-)I + \Delta t B_h] \delta v^{n+1} = \delta v^n + \Delta t k^+ \delta u^n + O(\Delta t(\Delta t + h^2)) . \quad (5.26)$$

For simplicity we now introduce the notation:

$$\begin{aligned} A &= (1 + \Delta t k^+)I + \Delta t A_h , \\ B &= (1 + \Delta t k^-)I + \Delta t B_h . \end{aligned}$$

Thus the error for our complete system can be written as:

$$\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} \begin{pmatrix} \delta u \\ \delta v \end{pmatrix}^{n+1} = \begin{pmatrix} \delta u \\ \delta v \end{pmatrix}^n + \Delta t \begin{pmatrix} k^- \delta v \\ k^+ \delta u \end{pmatrix}^n + O(\Delta t(\Delta t + h^2)) . \quad (5.27)$$

From Lemma 5.1.2 it is known that  $A$  and  $B$  are regular, in this case it is generally known that:

$$\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} & 0 \\ 0 & B^{-1} \end{pmatrix} . \quad (5.28)$$

Applying the norm and the fact that  $\|A^{-1}\|, \|B^{-1}\| \leq 1$ , from (5.16), we get the following bound:

$$\left\| \begin{pmatrix} \delta u \\ \delta v \end{pmatrix}^{n+1} \right\| \leq \left\| \begin{pmatrix} \delta u \\ \delta v \end{pmatrix}^n \right\| + \Delta t K \left\| \begin{pmatrix} \delta v \\ \delta u \end{pmatrix}^n \right\| + O(\Delta t(\Delta t + h^2)) , \quad (5.29)$$

where  $K = \max(k^-, k^+)$ . Letting  $z = \begin{pmatrix} \delta u \\ \delta v \end{pmatrix}$ , the bound can be rewritten as:

$$\|z^{n+1}\| \leq (1 + \Delta t K) \|z^n\| + O(\Delta t(\Delta t + h^2)) , \quad (5.30)$$

which leads us to the use of Gronwall's Lemma.

**Lemma 5.2.1** (Discrete Gronwall Lemma). *Let a positive sequence  $\{z^n\}_{n=0}^N$  satisfy*

$$z^{n+1} \leq a z^n + b , \quad \forall n = 0, \dots, N-1 ,$$

*and let  $a$  and  $b$  be constants with  $a > 0$ . Then for  $a \neq 1$ :*

$$z^n \leq a^n z^0 + \frac{b(a^n - 1)}{a - 1} , \quad \forall n = 0, \dots, N-1 ,$$

*and for  $a = 1$ :*

$$z^n \leq n b + z^0 , \quad \forall n = 0, \dots, N-1 .$$

*Proof of Lemma 5.2.1.*

For  $a = 1$ : The proof is elementary and follows from induction.

For  $a \neq 1$ : Obviously the result holds for  $n = 0$ . Assuming that the lemma holds for  $n = k$ , then

$$z^{k+1} \leq a z^k + b \leq \frac{ab(a^k - 1)}{a - 1} + a^{k+1} z^0 + b = \frac{b(a^{k+1} - 1)}{a - 1} + a^{k+1} z^0 .$$

□

Applying Gronwall's Lemma to (5.30) results in:

$$\begin{aligned} \|z^n\| &\leq (1 + \Delta t K) \|z^0\| + \frac{(1 + \Delta t K)^n - 1}{(1 + \Delta t K) - 1} O(\Delta t(\Delta t + h^2)) \\ &\leq \exp(TK) \|z^0\| + \frac{\exp(TK) - 1}{K} O(\Delta t + h^2) . \end{aligned} \quad (5.31)$$

Thus, for fixed  $T \geq n\Delta t$ , the global error goes linearly to zero as  $\Delta t$  goes to zero, which confirms that Euler's method is convergent of order 1, since  $\|\delta u^{n+1}\| = O(\Delta t)$ .

## Chapter 6

# Implementation

As we have seen in the preceding analysis our model preserves positivity and is stable. In Chapter 5 we saw that stability implies that the approximated solution will converge towards the correct solution of the PDE model. Therefore, next step would be to implement the system.

Through this Chapter the reader is referred to several files. These files can be found on <http://github.com/ChristianValdemar>.

### 6.1 Mesh and Cell Geometry

The Chan-Vese method from 4.1 is implemented in MATLAB in such a way that the contour, which encircles the object, is outputted as an polygon<sup>1</sup>. Polygons are especially easy to handle when working with meshes. To determine the geometry of one cell, the Chan-Vese method is very useful. However when two cells lie close to each other, like in Figure 6.1(a), the Chan-Vese algorithm cannot distinguish between the cells. Thus, for Chan-Vese to work, we need to remove the cell, which is not in our interest.

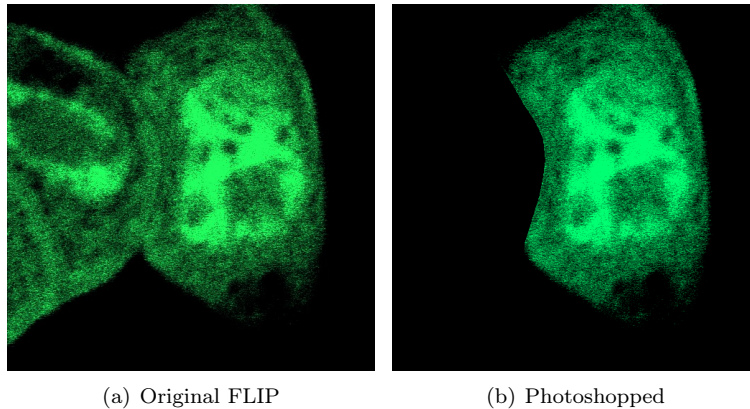


Figure 6.1: Photoshopped images for segmentation.

---

<sup>1</sup>The code can be found in `/Segmentation/Point-generator-whole-cell.m`

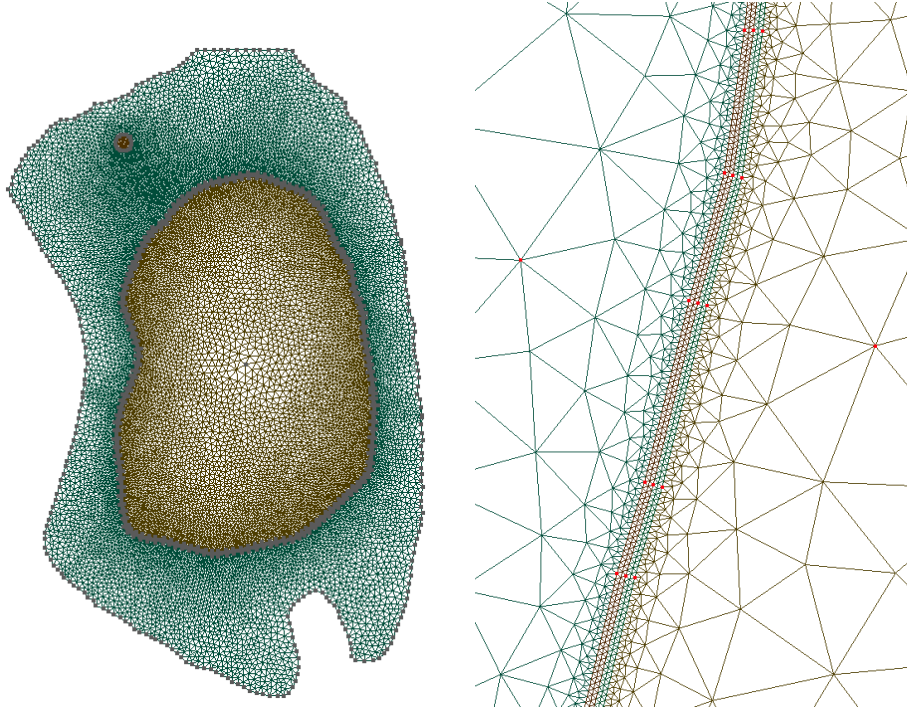


Figure 6.2: Finite element mesh on Chan-Vese active contours.

Figure 6.3: Close-up illustration of the mesh at the membrane.

Figure 6.1(a) comes from the original FLIP sequence. As can be seen in the photoshopped version Figure 6.1(b), the neighboring cell was manually erased, so that only the cell of interest remains.

As described, we use the Chan-Vese implementation to find the polygon, which spans the cell on Figure 6.1(b). The Chan-Vese implementation is also used to find and describe the bleaching area and the nucleus. The nucleus is segmented from the last image in the FLIP sequence, where only the nucleus remains, see Figure 2.3(j). However the segmentation of the nucleus needs to be expanded into a nuclear membrane which divides the cell into two cellular compartments, cytoplasm and nucleus. The membrane must have a small but finite thickness. The idea is to split the segmented polygon into two new polygons describing the nuclear membrane boundaries  $\partial\Omega_M$ . Each point on the polygon is moved in the normal direction to form a new polygon which is either smaller or larger than the segmented nucleus. Since the polygon from the Chan-Vese segmentation is derived from the level set function  $L$ , the outward normal  $n$  is given by:

$$\frac{\nabla L(x)}{|\nabla L(x)|} = n(x) . \quad (6.1)$$

Note that the polygon needs to be convex or close to convex [2].

The polygons describing the cell, the bleaching area and the nucleus membrane give us the main structure on which the mesh is produced. The MATLAB code

is written in such a way that it provides us with a `.geo` file. The `.geo` file holds points from the polygons, the lines between them and information about the surfaces<sup>2</sup>. All points are scaled, such that the cell geometry takes its real size. For this particular cell, the scale factor is  $0.0448 \mu\text{m}/\text{pixel}$ . The `.geo` file also holds informations about surfaces, each surface is formed by one or more polygons. The `.geo` file also tells which points belong to a particular surface. For each point a three dimensional coordinate is given, and the prescribed mesh element size is also set at that point. For points in the membrane, the mesh element size is set to be very small so that the mesh becomes dense inside the membrane. This leads to over-refinement both in the cytoplasm and the nucleus, in order to avoid an unnecessary fine mesh in these areas, we add “dummy” points. These dummy points are placed in the nucleus as well as in the cytoplasm. In Figure 6.3 two red dummy points can be seen, one in the cytoplasm and one in the nucleus.

The `.geo` file now holds all information about the cell geometry. Thus we can use Gmsh [14] to generate the mesh displayed in Figure 6.2. The mesh consists of 221558 triangles, where 68534 are located in the cytoplasm, 114 in the bleaching area, 89077 in the membrane and 63833 in the nucleus. The mesh provided directly by Gmsh can be found in `/Simulation/Mesh/mesh.msh` and opened directly in Gmsh. However for simulation purposes we use FEniCS, which is not capable to work with `.msh` files. FEniCS only handles `.xml` files.

Available online is the `dolfin-convert.py` by A. Logg (2006), that converts the `.msh` to a `.xml` file. However the `.msh` file holds information about which subdomain a specific triangle belongs to, an information that is missing in the converted mesh file of type `.xml`. We therefore introduce an extra `.xml` file, which holds the subdomain informations. To do this, the Dolfin-converter is modified, so that the mesh file from Gmsh is converted into two files of type `.xml` containing the mesh and subdomains respectively. For more reading about Gmsh and the file types, I refer the reader to [12, 22]. The `.xml` files can be found in the folder `/Simulation/Mesh/`.

Using the above procedure, it is now possible to run a simulation on the cell geometry by use of Finite Element Methods in FEniCS.

## 6.2 Numerical Method

FEniCS is a software which aims to automate scientific computing, particularly automated solutions of differential equations by Finite Element Methods [22]. Thus for FEniCS to solve our FLIP model from (4.16) we need to reformulate the PDE problem as a variational problem. The last step is then to write a Python program which via FEniCS solves our PDE system. The Python code contains all the necessary informations such as the variational problem, initial data, boundary conditions, mesh, domains etc.

To solve our time-dependent PDE by the Finite Element Method we first discretize the time derivative. For simplicity and stability reasons the backward Euler difference method from (5.3) is used, so that the time discrete problem yields:

<sup>2</sup>The cell geometry can be found in `/Simulation/Mesh/contourpoints.geo`

$$\begin{aligned}\frac{u^{n+1} - u^n}{\Delta t} &= \nabla \cdot (\alpha \nabla u^{n+1}) + (k^- v^{n+1} - k^+ u^{n+1}) \Big|_{\Omega_M} - \theta b \frac{q}{1+q} u^{n+1} \Big|_{\Omega_B}, \\ \frac{v^{n+1} - v^n}{\Delta t} &= \nabla \cdot (\beta \nabla v^{n+1}) + (k^+ u^{n+1} - k^- v^{n+1}) \Big|_{\Omega_M},\end{aligned}\quad (6.2)$$

for  $x \in \Omega$ ,  $t > 0$ , with diffusion coefficients, Neumann boundaries and initial conditions as in (4.17), (4.18) and (4.19) respectively.

To derive the weak form, let  $\phi$  and  $\psi$  be the usual test functions for  $u$  and  $v$  respectively. Now recall from Green's Theorem that

$$-\int_{\Omega} (\Delta u) \phi \, dx = \int_{\Omega} \nabla u \cdot \nabla \phi \, dx - \int_{\partial\Omega} \frac{\partial u}{\partial n} \phi \, ds. \quad (6.3)$$

Note that this is also the two-dimensional version of the divergence theorem.

By multiplying the time-discrete equations from (6.2) with the test functions  $\phi$  and  $\psi$  respectively and integrating we now obtain:

$$\begin{aligned}\int_{\Omega} \frac{u^{n+1} - u^n}{\Delta t} \phi \, dx - \int_{\Omega} \nabla \cdot (\alpha \nabla u^{n+1}) \phi \, dx \\ - \int_{\Omega_M} (k^- v^{n+1} - k^+ u^{n+1}) \phi \, dx + \int_{\Omega_B} \theta b \frac{q}{1+q} u^{n+1} \phi \, dx = 0,\end{aligned}\quad (6.4)$$

$$\begin{aligned}\int_{\Omega} \frac{v^{n+1} - v^n}{\Delta t} \psi \, dx - \int_{\Omega} \nabla \cdot (\beta \nabla v^{n+1}) \psi \, dx \\ - \int_{\Omega_M} (k^+ u^{n+1} - k^- v^{n+1}) \psi \, dx = 0.\end{aligned}\quad (6.5)$$

By applying Green's theorem from (6.3) and having in mind that  $\frac{\partial u}{\partial n} = 0$  on  $\partial\Omega$  the weak form reads: find  $u, v \in H^1(\Omega)$  such that

$$\begin{aligned}\int_{\Omega} \frac{u^{n+1} - u^n}{\Delta t} \phi \, dx + \int_{\Omega} \alpha \nabla u^{n+1} \cdot \nabla \phi \, dx \\ - \int_{\Omega_M} (k^- v^{n+1} - k^+ u^{n+1}) \phi \, dx + \int_{\Omega_B} \theta b \frac{q}{1+q} u^{n+1} \phi \, dx = 0,\end{aligned}\quad (6.6)$$

$$\begin{aligned}\int_{\Omega} \frac{v^{n+1} - v^n}{\Delta t} \psi \, dx + \int_{\Omega} \beta \nabla v^{n+1} \cdot \nabla \psi \, dx \\ - \int_{\Omega_M} (k^+ u^{n+1} - k^- v^{n+1}) \psi \, dx = 0,\end{aligned}\quad (6.7)$$

for all  $\phi, \psi \in H^1(\Omega)$ .

FEniCS is able to solve this system in each time step by assembling the linear system and then solving it. This is a very time consuming operation, therefore it would be better to explicitly create the matrices and vectors from the linear system of the form  $Au = b$ , and thus solve it with respect to the

vector of unknowns. Before going deeper into this, let us consider the weak system from (6.6) on the form  $a(u^{n+1}, \phi) = L(\phi)$ ,

$$\begin{aligned} \int_{\Omega} \frac{u^{n+1}}{\Delta t} \phi \, dx + \int_{\Omega} \alpha \nabla u^{n+1} \cdot \nabla \phi \, dx + \int_{\Omega_M} k^+ u^{n+1} \phi \, dx + \int_{\Omega_B} \theta b \frac{q}{1+q} u^{n+1} \phi \, dx \\ = \int_{\Omega} \frac{u^n}{\Delta t} \phi \, dx + \int_{\Omega_M} k^- v^{n+1} \phi \, dx . \end{aligned} \quad (6.8)$$

By rewriting (6.7) into the same form, the bilinear and linear form  $a$  and  $L$ , can be seen as the left hand side and right hand side from the equation below respectively:

$$\begin{aligned} \int_{\Omega} \frac{v^{n+1}}{\Delta t} \psi \, dx + \int_{\Omega} \beta \nabla v^{n+1} \cdot \nabla \psi \, dx + \int_{\Omega_M} k^- v^{n+1} \psi \, dx \\ = \int_{\Omega} \frac{v^n}{\Delta t} \psi \, dx + \int_{\Omega_M} k^+ u^{n+1} \psi \, dx . \end{aligned} \quad (6.9)$$

Here  $u$  and  $v$  are finite element functions, which are expressed as the linear combination of basis functions  $\rho_j$  and the value  $u_j$  in vertex  $p_j$  for  $j = 1, \dots, N$ :

$$u = \sum_{j=1}^N u_j \rho_j . \quad (6.10)$$

Where  $\rho_j$  is defined as,

$$\rho_j(p_i) = \begin{cases} 1, & j = i, \\ 0, & j \neq i. \end{cases} \quad (6.11)$$

If  $u$  satisfies  $a(u, \phi) = L(\phi)$ ,  $\forall \phi \in H^1(\Omega)$ , then in particular

$$a(u, \rho_i) = L(\rho_i), \quad \forall i = 1, \dots, N. \quad (6.12)$$

Inserting (6.10) into (6.12) implies

$$\sum_{j=1}^N a(\rho_j, \rho_i) u_j = L(\rho_i), \quad i = 1, \dots, N, \quad (6.13)$$

which can be seen as a linear system of the form:

$$Au = b, \quad (6.14)$$

where the entries in  $A$  and  $b$  are given by

$$A_{ij} = a(\rho_j, \rho_i), \quad (6.15)$$

$$b_i = L(\rho_i). \quad (6.16)$$

It is worth noticing that  $a(v, \psi)$ , which is the left hand side from (6.9), is independent of time. From previous derivation it can be seen that the associated matrix  $A$  thus also will be independent of time. Thus the matrix  $A$  can be computed prior to the time stepping iteration. This will lower the computation

time, since we then only need to compute the right-hand side  $b$  at each time step.

Assembling the  $A$  matrix is a very time consuming operation, thus a prior assembling is highly preferable. Observe from (6.8) that  $a(u, \phi)$  holds the bleaching term. The bleaching term is turned on and off by the indicator function  $\theta$ , here the bleaching time  $\Delta t_b = 0.4$  sec whereas the frame rate is 1.6 sec, thus:

$$\theta = \begin{cases} 1, & \text{if } t(\bmod 1.6) \leq 0.4, \\ 0, & \text{else.} \end{cases} \quad (6.17)$$

Since  $\theta$  is piecewise constant we assemble two bilinear systems prior, one with and one without the bleaching term. The pulsing laser beam is thus realized by solving the system with the bleaching term inside the bleaching time interval, which length is a multiple of the time step:  $\Delta t_b = n\Delta t$ .

### 6.3 FEniCS Implementation

As discussed in section 6.1 the mesh needs to be dense inside the membrane since  $u$  and  $v$  goes to zero on a very small interval and transport happens rapidly. A consequence of the large amount of cells, the system becomes very large and the computation will be slower. To avoid a slow initial calculation process before each simulation run, the initialization of  $u^0$ ,  $v^0$ ,  $k^+$  and  $k^-$  is written in a separate Python code. The code, which can be found in `/Simulation/Initialize_logistic.py` is a FEniCS program, which writes the initial data in `.xml` files.

$u^0$ ,  $v^0$ ,  $k^+$  and  $k^-$  are all initialized as logistic functions of the form from (4.9). For the implementation we use  $\text{dist}(x)$  instead of  $(x - x_m)n$  to denote the shortest signed distance from a point  $P : (x_0, y_0)$  to the center of the membrane. From the Chan-Vese segmentation we have several `.txt` files. These files hold the points that represent the polygons, which form the membrane, (see the red dots at Figure 6.4). To illustrate, consider the blue point on Figure 6.4 to be our  $P$ . The approach from here is to find the shortest distance to the nearest edge in the polygon that represents the center of the membrane.

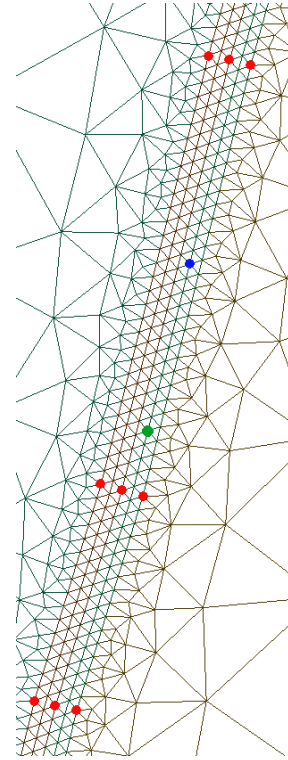


Figure 6.4: Closeup on mesh near membrane.

The first step of the algorithm is to find the points from the center polygon which are closest to  $P$ . This is done by the k-nearest neighbor algorithm, which calculates the euclidean distance between  $P$  and all points in the center polygon, and afterwards sort them. The distances can be found in a for-loop, however



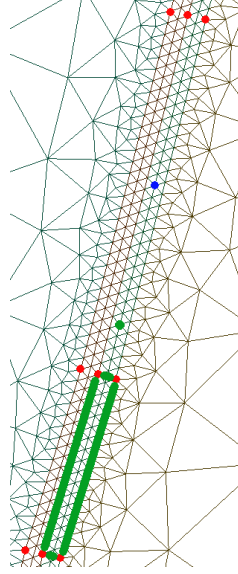


Figure 6.5: Close-up illustration of the mesh at the membrane. The first step in the algorithm is to calculate the distance from the nucleus edge.

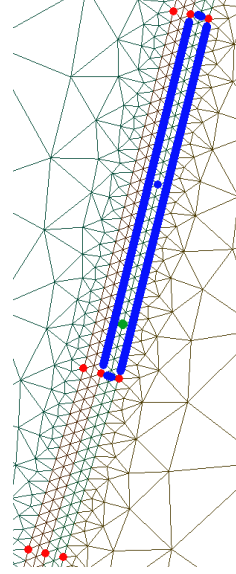


Figure 6.6: Close-up illustration of the mesh at the membrane. The second step in the algorithm is to calculate the distance from the nucleus edge.

this is a rather slow process. For faster computation in Python we use Numpy arrays. For  $P$  being the blue dot in Figure 6.4, the line between  $P$ 's two nearest neighbors will constitute the edge of the center polygon which lies closest to  $P$ . However if  $P$  is the green point, the line between the two nearest neighbors will not be identical to the closest edge in the center polygon. We therefore need to check, whether the line between the two nearest neighbors really is the closest polygon edge to  $P$ .

To check if the two nearest neighbors really are the correct ones, we check whether  $P$  lies inside the rectangle that the two points, together with its two neighbor points at  $\partial\Omega_M$ , forms. To illustrate we now choose the green point from Figure 6.4 to be our  $P$ . Thus the two nearest neighbors and their neighboring points at  $\partial\Omega_M$  form the green rectangle in Figure 6.5. We now have to check if  $x$  lies inside this rectangle, a problem, which is equivalent to the point-in-polygon (PIP) problem. One of the most popular algorithms to solve the PIP problem, is the Ray Casting Method [28]. In short the algorithm is based on the fact, that for a ray from any exterior point of the polygon to the point of interest, it is known, that if the ray passes the polygon boundary an odd number of times, then the point of interest lies inside the polygon. In this example the green point  $P$  lies outside the chosen rectangle in Figure 6.5. Therefore the nearest polygon edge must be spanned by two other points. It is obvious, that the point on the center polygon, which is closest to  $P$  must be one of the two points that we are looking for. Therefore, we keep that point and exchange the other point with the third closest point. From these two points we now get the blue rectangle in Figure 6.6. As the illustration shows our green  $P$  lies inside the rectangle, which is confirmed by the Ray Casting algorithm. Consequently,

we can construct the line that represents the polygon edge, which is nearest  $P$ . If these two points did not introduce the nearest polygon edge, we would simply continue our search through the nearest neighbor list.

Now denote the two points derived from the algorithm above by  $P_1 : (x_1, y_1)$  and  $P_2 : (x_2, y_2)$ . With the direction vector  $\overline{P_1 P_2} = (\alpha, \beta)^T$ , where  $\alpha = x_2 - x_1$  and  $\beta = y_2 - y_1$ . In  $\mathbb{R}^2$ , the line through  $P_1$  and  $P_2$  can be represented at affine form:

$$ax + by + c = 0, \quad (6.18)$$

where  $a = 1/\alpha$ ,  $b = -1/\beta$  and  $c = -x_1/\alpha + y_2/\beta$ . The distance between point  $P$  and the polygon is then found from Lemma 6.3.1, and afterwards used to initialize  $u^0$ ,  $v^0$ ,  $k^+$  and  $k^-$  as described earlier in this section. Notice that for the implementation, the distance from  $x$  to the center of the membrane is split into two cases. First for  $x$  in nucleus or the half section of the membrane which faces nucleus, the distance from  $x$  to the center of the membrane is  $(x - x_m)n = -\text{dist}(x)$ . Where as for  $x$  in cytoplasm or the outer half section of the membrane which faces cytoplasm, the distance from  $x$  to the center of the membrane is  $(x - x_m)n = \text{dist}(x)$ .

**Lemma 6.3.1.** *Let  $P$  be a point with coordinates  $(x_0, y_0)$  and let  $ax + by + c = 0$  be a line in  $\mathbb{R}^2$  then the distance between point and line can be represented by*

$$\text{dist}(P) = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}$$

*Proof of lemma 6.3.1.*

Let  $Q : (x_1, y_1)$  be a point on the line and  $n$  be the normal vector such that it starts in  $Q$  and  $n = (a, b)^T$ .

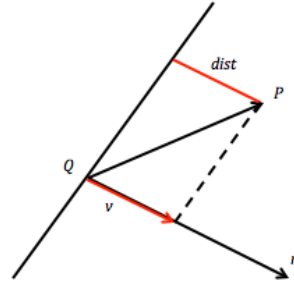
Projecting the vector  $\overline{QP}$  onto  $n$  gives a vector  $v = d\hat{n}$ , where  $\hat{n}$  is the unit normal and  $d$  is a scalar. This defines the scalar projection, where

$$d = \|\overline{QP}\| \cos \theta = \overline{QP} \cdot \hat{n} = \frac{\overline{QP} \cdot n}{\|n\|},$$

where  $\theta$  is the angle between  $\overline{QP}$  and  $n$ . Depending on the direction of  $n$ ,  $d$  may take both positive and negative values. Thus,

$$\text{dist}(P) = \|v\| = |d| = \frac{|\overline{QP} \cdot n|}{\|n\|}.$$

Note that  $\overline{QP} \cdot n = a(x_0 - x_1) + b(y_0 - y_1)$ ,  $\|n\| = \sqrt{a^2 + b^2}$  and since  $Q$  lies on the line  $c = -ax_1 - by_1$ , which proves the lemma.  $\square$



As already mentioned the initialization of  $u^0$ ,  $v^0$ ,  $k^+$  and  $k^-$  described above is implemented in `/Simulation/Initialize_logistic.py`. The main FEniCS program, in which the the weak form from (6.6) and (6.7) is implemented can be found in `/Simulation/bleaching_logistic.py`.

## 6.4 Tuning Reaction- and Diffusion Parameters

To examine the size of the reaction and diffusion rates, consider a one-dimensional version of our model without the bleaching term. Thus the system from (4.16) simplifies to:

$$\begin{aligned} u_t &= (\alpha u_x)_x + k^- v - k^+ u, \\ v_t &= (\beta v_x)_x + k^+ u - k^- v, \quad x \in \Omega, \quad t > 0. \end{aligned} \quad (6.19)$$

Here  $\Omega = [0, 0.029]$  measured in  $\mu\text{m}$  is the interval, constructed as a cross-section of the nucleus membrane of width 9nm. The membrane is located in the center of the interval such that  $\Omega_M = [0.01, 0.019]$ , cytoplasm  $\Omega_C = [0, 0.01]$  and nucleus  $\Omega_N = (0.019, 0.029]$ . For this parameter test we set the initial data to be:

$$u^0 = \begin{cases} 1 & \text{if } x \in \Omega_C \\ 0.5 & \text{if } x \in \Omega_M \\ 0 & \text{if } x \in \Omega_N \end{cases}, \quad v^0 = \begin{cases} 0 & \text{if } x \in \Omega_C \\ 0.5 & \text{if } x \in \Omega_M \\ 1 & \text{if } x \in \Omega_N \end{cases}. \quad (6.20)$$

Note that  $c^0 = u^0 + v^0 = 1$ . As mentioned in Chapter 2, it is known that eGFP diffuses very rapidly inside cytoplasm and nucleus  $\approx 25 \mu\text{m}^2/\text{s}$ . Likewise it has also been mentioned, that the diffusive transport through the nuclear membrane is rather slow. Thus the diffusion rates are set to:

$$\alpha = \begin{cases} 25 & \text{if } x \in \Omega_C \\ 5 \cdot 10^{-4} & \text{if } x \in \Omega_M \\ 10^{-3} & \text{if } x \in \Omega_N \end{cases}, \quad \beta = \begin{cases} 10^{-3} & \text{if } x \in \Omega_C \\ 2.5 \cdot 10^{-4} & \text{if } x \in \Omega_M \\ 25 & \text{if } x \in \Omega_N \end{cases}. \quad (6.21)$$

The reaction parameters  $k^+$  and  $k^-$  are chosen to be on the logistic form from (4.9) with scale parameter  $\epsilon = 10^{-4}$ . This model is implemented as described in Section 6.3 and can be found in `/1D-simulation/1D-bleaching-logistic.py`. The simulation is made with three different values of the rate parameter  $k$ , such that  $k = 10^2$ ,  $10^3$  or  $10^4$ . The eGFP concentrations dispatched from the simulation at time  $t = 5$  are depicted in Figures 6.7, 6.8 and 6.9.

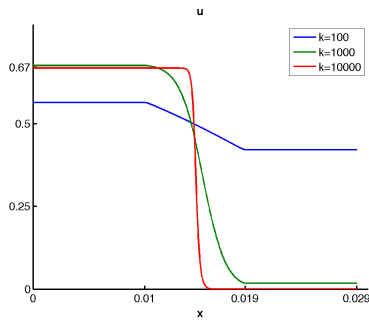


Figure 6.7:  $u$  at time  $t = 5$  sec.

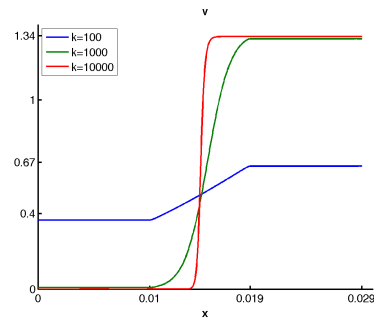


Figure 6.8:  $v$  at time  $t = 5$  sec.

Note that in agreement with Corollary 5.1.3, positivity is preserved. It is crucial for our model and mass preservation that  $u$  and  $v$  become zero in nucleus

and cytoplasm respectively, thus  $k$  and  $\epsilon$  need to be high enough. For the chosen diffusion coefficients inside the membrane,  $k \approx 10^4$  and  $\epsilon \approx 10^{-4}$  are sufficient.

Recall that  $c^0 = u^0 + v^0 = 1$ . For this simulation at time  $t = 5$  the resulting concentration in the entire interval is represented as  $c = u + v$  on Figure 6.9.

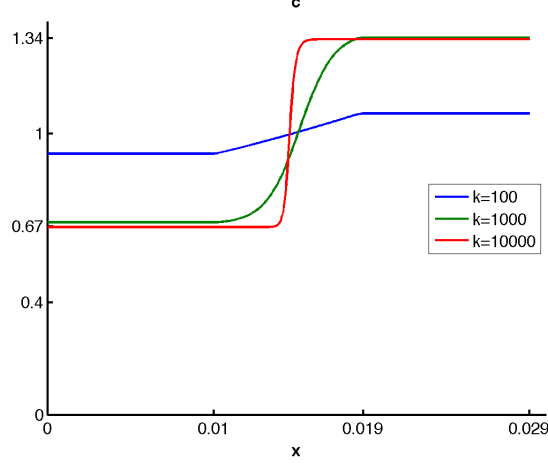


Figure 6.9:  $c = u + v$  at time  $t = 5$  sec.

As discussed several times through this report, the transport from cytoplasm into nucleus should be faster than the reversed transport, so that the concentration inside nucleus at equilibrium would be higher than in the cytoplasm. As can be seen in Figure 6.9, our reaction-diffusion model satisfies this requirement by achieving equilibrium, with concentration in cytoplasm and nucleus at  $c_C \approx 0.67$  and  $c_N \approx 1.34$  respectively. This observation is confirmed in the two-dimensional case too, in such a way, that it agrees with the FLIP images (see Section 6.5). The observant reader may have noticed, that the ratio between fluorescence intensities in cytoplasm and nucleus agrees with the ratio of the diffusion rate in the membrane,

$$2 = \frac{\alpha_M}{\beta_M} \approx \frac{c_N}{c_C}. \quad (6.22)$$

## 6.5 Simulation on Real Cell Geometry

To simulate the experiment from Figure 2.3, we compute the FLIP model from (4.16), (4.17), (4.18), (4.19) and (6.23) in FEniCS with the implementation described in Section 6.3 on the mesh displayed in Figure 6.2. As mentioned, the frame rate including bleaching for the FLIP experiment at Figure 2.3 was 1.6 seconds. The bleaching takes place inside the indicated area for  $\Delta t_b = 0.4$  seconds, the recovery/photo phase lasts 1.2 sec. For simulation purposes the bleaching rate is given by  $\frac{bq}{1+q} = 160$  and the time step is chosen as  $\Delta t = 0.1$ .

We choose  $\epsilon = 10^{-4}$ ,  $k = 10^4$  and diffusion rates as in (6.21). The initial data shown in Figure 6.10(a) are, as discussed in Section 4.2, logistic functions across the membrane

$$u^0 = \frac{53.9}{1 + e^{-k(x-x_m)n}} \quad \text{and} \quad v^0 = \frac{119.4}{1 + e^{k(x-x_m)n}}. \quad (6.23)$$

The initial data are chosen based on the average intensities from the first FLIP image.

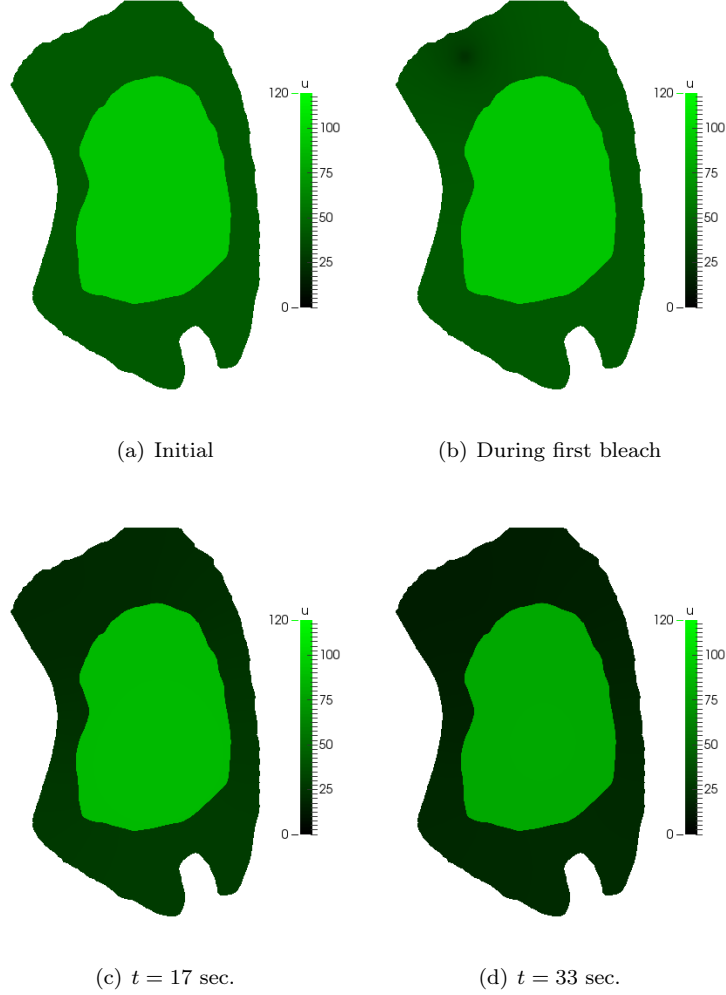


Figure 6.10: 2D simulation of diffusion-limited FLIP with eGFP in a McA cell.

Note that our observation in (6.22) does not match our initial data from (6.23):

$$\frac{c_N}{c_C} = \frac{119.4}{53.9} \approx 2.22 \neq 2 .$$

This is caused by the observation, that the difference in the average values from the FLIP sequence is significantly higher between the first and the second image than the rest. This may be caused by a small break between the first image and the first bleaching. For a consistent simulation, we therefore use the second FLIP image as basis for the simulation. The first bleach have already occurred on the second FLIP image, which is why the ratio between nucleus and

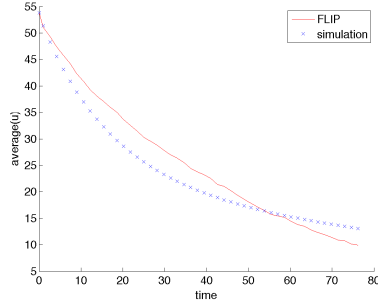


Figure 6.11: Average fluorescence intensity in cytoplasm.

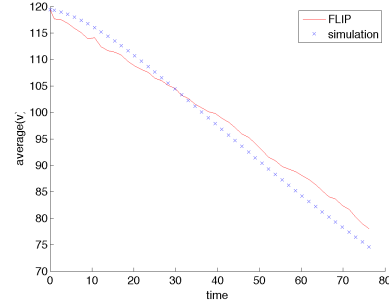


Figure 6.12: Average fluorescence intensity in nucleus.

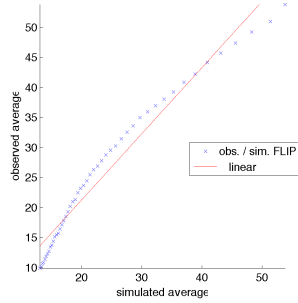


Figure 6.13: Average fluorescence intensity in cytoplasm.

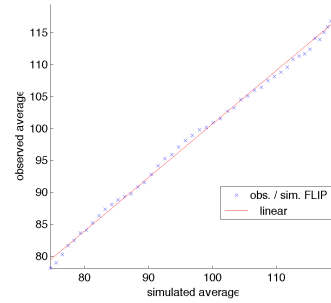


Figure 6.14: Average fluorescence intensity in nucleus.

cytoplasm is higher than 2, which is actually the ratio between the intensities in nucleus and cytoplasm at the first FLIP image, where equilibrium is obtained.

Observe from Figure 6.10 that just like in the FLIP experiment, the bleaching does not have neither a major nor a fast impact on nucleus intensities compared to what is observed in cytoplasm. Thus the membrane shows its protective effect.

By converting the FLIP sequence to gray-scale images and using Chan-Vese segmentation in `/Segmentation/FLIP-average-analysis.m` we find the average intensities of nucleus and cytoplasm respectively for each image. It takes the Zeiss LSM 510 confocal microscope 1.2 seconds to make a full image of the cell, this of course gives rise to inaccurate comparisons. So for simplicity we use the average values from the simulation at time  $t = 1.6n + 1$ ,  $n \in \mathbb{N}$ , corresponding to the moment when image process is midway. These average intensities from the simulation can be seen on Figure 6.11 and 6.12 where they are plotted together with the average intensities from the FLIP images.

Even though the initially simulated intensities are equivalent to the observations, the simulated intensities in cytoplasm drops faster initially. This deviation may be attenuated by further tuning of reaction and diffusion rates in future work, a tuning process that can be automated. On the other hand, we ob-

serve that the simulated nucleus intensities follow the observed intensities from the FLIP sequence very well. An observation that is confirmed, when looking at the correlation plots in Figure 6.13 and 6.14. Thus observe that the FLIP intensities and the computational model seem to be linearly correlated with a correlation coefficient at 0.980 and 0.998 for cytoplasm and nucleus respectively.

## 6.6 Diffusion Rates as Transport Rates

In Section 6.4, we observed that the ratio of average fluorescence intensities in the cytoplasm and nucleus is approximately determined by the ratio of membrane diffusion rates (6.22). In order to confirm this observation a 2D simulation test was set up. The model is equivalent to the previous 2D simulation except that the initial data was:

$$u^0 = \frac{1}{1 + e^{-k(x-x_m)n}} \quad \text{and} \quad v^0 = \frac{1}{1 + e^{k(x-x_m)n}}, \quad (6.24)$$

so that  $c^0 = u^0 + v^0 = 1$ , and the bleaching term is turned off  $b = 0$ . The average values from the simulation can be seen in Figure 6.15.

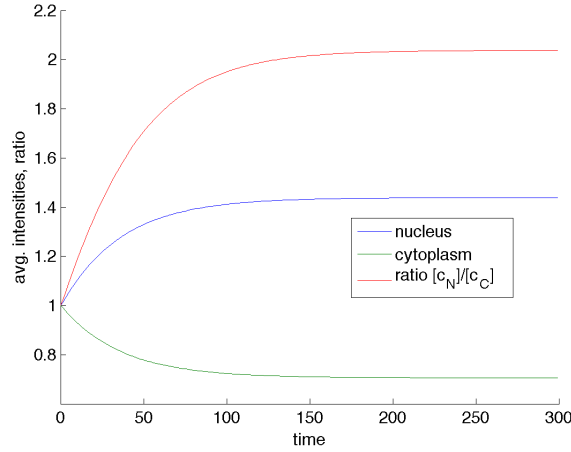


Figure 6.15: Average intensities and ratio over time.

Thus it can be seen that over time, our simulation comes into equilibrium and the average fluorescence intensities inside cytoplasm and nucleus become constant. In Figure 6.15 it is also seen, that the ratio of intensities approaches the ratio of diffusion rates  $\frac{\alpha_M}{\beta_M} \approx 2$ .

From several simulations and the way that the model is constructed, we observe that the speed of transport across the membrane depends on the size of the membrane diffusion rates  $\alpha_M$  and  $\beta_M$ , rather than on the transport rates  $k^\pm$ . This can be explained by the fact that the diffusion rates control the rate at which the eGFP enters the membrane, and thereby how much eGFP that can be transported.

## Chapter 7

# Final Remarks

We describe a computational Reaction-Diffusion Model of FLIP image sequences of eGFP transport between the cytoplasm and nucleus of mammalian cells. Our model performs these simulations in the real cell geometry extracted from microscopy data gathered from a confocal microscope. We found that the transport term describing nucleocytoplasmic exchange of eGFP must include rate coefficients being a function of distance along the normal to the compartment boundaries.

Without the space dependent reaction constants, discontinuity in the FEM system would appear and by the usual FEM, lead to instability. To approximate fluorescence intensity across the nuclear membrane, both the discontinuous finite element method and/or the partition of unity/extended finite element method might be appropriate alternatives to avoid instability [1, 22].

We have proved, positivity and numerical stability of the model. Stability implies that approximate solutions computed by the FLIP model will converge towards a correct solution of the PDE model which itself is an approximation to reality. Also, from Figure 6.11 and 6.12 we found, that our simulation results are highly correlated with experimental observations, however it is still not perfect.

There are a lot of aspects that may lead to inaccurate simulations, as seen in Figure 6.11 and 6.12. Another aspect is that our data is collected from a stationary geometry, whereas, the living cell moves through the FLIP sequence. An aspect that can have a big effect on the average intensities. In order to achieve a more accurate system, we would introduce a bleaching term affecting the intensities of the whole cell. The present model for membrane transport is only the simplest of its kind. The investigation of advanced and nonlinear models will be the subject of future studies. Therefore we plan to apply the reliable FLIP simulator in inverse problem analysis to estimate reaction and diffusion parameters and thus to automatically calibrate the model to experimentally observed images.



## Appendix A

# Paper: Computational Modeling of Fluorescence Loss in Photobleaching

As a result of this project and in corporation with Achim Schroll and Daniel Wüstner the paper *Computational Modeling of Fluorescence Loss in Photobleaching* has been produced. Currently the paper has been submitted to the journal *Computing and Visualization in Science* by Springer and waiting for approval. The full paper can be found on next page.

# Computational Modeling of Fluorescence Loss in Photobleaching

Christian V. Hansen · Hans J. Schroll · Daniel Wüstner

the date of receipt and acceptance should be inserted later

**Abstract** Fluorescence loss in photobleaching (FLIP) is a modern microscopy method for visualization of transport processes in living cells. Although FLIP is widespread, an automated reliable analysis of image data is still lacking. This paper presents a well-posed computational model based on spatially resolved diffusion and transport rates. The model is a reaction-diffusion system, discretized by continuous finite elements. The cell geometry is segmented from FLIP images using an active contours algorithm and the PDE model is subsequently solved in real, two-dimensional geometry. Based on this model, FLIP images are simulated and thus molecular transport in living cells is reliably quantified.

## 1 Introduction

Many human diseases involve or are even caused by malfunction in intracellular protein and membrane trafficking. Examples are Huntington and Parkinson disease, Dubin-Johnson syndrome and various lysosomal storage disorders [1]. Monitoring of intracellular transport in health and disease heavily relies on quantitative fluorescence microscopy. In particular, photobleaching studies of the protein of interest tagged to enhanced

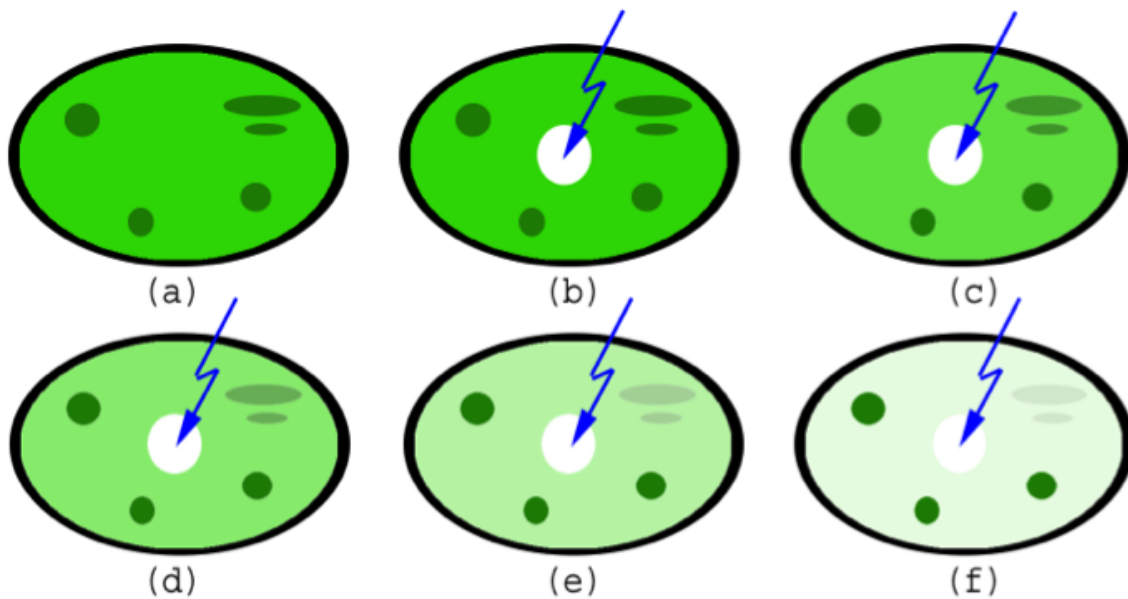
green fluorescent protein (eGFP) is increasingly acknowledged as tool for deciphering transport kinetics and dysfunction [5, 12]. In fluorescence recovery after photobleaching (FRAP) a selected region in the cell is rapidly bleached with a high intensity laser pulse followed by sequential imaging with reduced laser power to determine recovery kinetics due to protein transport into the bleached region. Extensive mathematical modeling has been applied to recover diffusion constants and binding parameters from fluorescence recovery curves [15]. Since the recovery curve in FRAP only depends on time, any spatial information of the acquired image sequence is lost with the consequence, that overfitting often takes place, and the recovered parameters are of low reliability [15]. Fluorescence loss in photobleaching (FLIP) is a variation of FRAP, in which the intense laser bleaching is performed repeatedly in a selected region, while fluorescence loss is monitored outside the bleached area. This bears the potential to include pixel-wise information in the modeling process. Recently, we have presented a pixel-by-pixel fitting of an empirical decay law to diffusion- and binding-limited FLIP image sequences of eGFP and eGFP-tagged Huntingtin protein [17]. Although a step in the right direction, our method did not allow us to model the spatiotemporal protein dynamics based on an underlying physical mechanism.

In this paper, we present a mechanistic computational model of FLIP imaging solving the reaction-diffusion problem in the real cell geometry being extracted from confocal image sequences. The model is proven to preserve positivity. It is well-posed in the sense of Hadamard, and standard discretization methods lead to convergent numerical approximations. In future work, our reliable FLIP simulator can serve as forward solver in an inverse analysis to quantify diffusion and transport rates. We believe that computa-

---

Chr. V. Hansen · H. J. Schroll  
Department of Mathematics and Computer Science, University of Southern Denmark, Campusvej 55, Odense M DK-5230, Denmark  
E-mail: christian@imada.sdu.dk, achim@imada.sdu.dk

D. Wüstner  
Department of Biochemistry and Molecular Biology, University of Southern Denmark, Campusvej 55, Odense M DK-5230, Denmark  
E-mail: wuestner@bmb.sdu.dk



**Fig. 1** Schematic illustration of a FLIP experiment. A cell contains fluorescent molecules (green) (a). A small region is repeatedly bleached by laser light (blue flash) (b-f). Fluorescence begins to fall in the cytoplasm and in some organelles (ellipses) but not in others (circles). This demonstrates diffusive transport of molecules towards the bleached region with recruitment from the elliptical organelle but not from the circular organelles, suggesting that the fluorescent molecules are confined in the latter.

tional modeling as presented here might enhance measurement technologies in the laboratory.

The outline of the paper is as follows: Fluorescence photobleaching technology is explained in Section 2. In Section 3 our reaction–diffusion model is developed. Positivity, well-posedness and convergence of a semi-implicit finite difference method is proven in Section 4. A finite element simulation of a FLIP experiment is presented in Section 5.3. Finally we summarize our findings in Section 6.

## 2 Fluorescence loss in photobleaching

Cellular transport processes are often investigated using fluorescence-tagged molecules and live-cell microscopy [12]. In FLIP experiments, a region of the cell is repeatedly illuminated by an intense laser pulse destroying the fluorescence of the molecule of interest in that area. A pause between the laser pulses allows for some recovery in the bleached region. Repeating this protocol several times creates a sink for the fluorescent molecules in the local environment being in continuous exchange with the bleached region. A decrease of fluorescence of the labeled molecules outside the bleached area allows for assessing continuity between intracellular compartments, and in principal, for measuring the kinetics of recruitment to the bleached region from various cellular

areas. Accordingly, FLIP has the potential to include spatial information in the computational analysis [17].

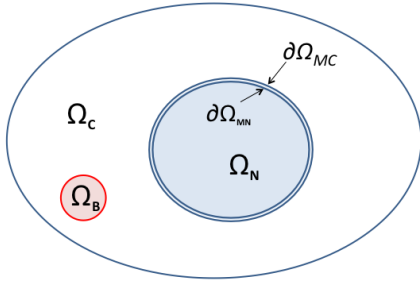
## 3 A reaction–diffusion model in realistic geometry

Living cells are in continuous exchange with their environment and use a variety of transport modes to secure the correct protein and lipid composition of their organelles. Simple Brownian motion and anomalous diffusion of proteins in the cytoplasm, active transport in form of vesicular shuttling along cytoskeleton tracks, budding, fission and fusion of membrane tubules as well as cytoplasmic streaming, all contribute to the complex intracellular traffic network. To describe this complexity in quantitative terms using mathematical models requires simplifying the observed phenomena to the most essential elements. For example, the cytoplasm has long been considered as a homogenous fluid with Newton-viscosity. Accordingly, transport models describing, for example intracellular diffusion of macromolecules contained only a single, spatially invariant diffusion constant. Recent evidence obtained using quantitative fluorescence and electron microscopy, however, has revealed a hierarchically organized entangled network architecture of the cytoplasm. As a consequence, realistic transport models have to account for this structural orga-

nization including space-dependent diffusion constants and complex cell shapes.

### 3.1 Segmentation of FLIP images

The first step toward a realistic simulation is to segment the FLIP image into its nucleus  $\Omega_N$ , the cytoplasm  $\Omega_C$  and the nuclear membrane  $\Omega_M$ . For later reference we denote the boundary of the nuclear membrane as  $\partial\Omega_M$  and split  $\partial\Omega_M$  into two disjoint sets  $\partial\Omega_{MC}$  and  $\partial\Omega_{MN}$ , such that  $\partial\Omega_{MC}$  is the boundary part that faces the cytoplasm and  $\partial\Omega_{MN}$  faces nucleus. The bleaching area is located within the cytoplasm  $\Omega_B \subset \Omega_C$ . In this way, the computational domain  $\Omega$  covers the full geometric complexity of the living cell under observation. See Fig. 2 for a schematic illustration.



**Fig. 2** Schematic illustration of segmented cell.

For segmentation the Chan-Vese active contours algorithm [6] is applied which detects different gray-levels  $c_-$  and  $c_+$  separated by a closed curve  $\Gamma$ . The separation line is represented as a zero-level curve

$$\Gamma = \{\mathbf{x} \in \mathbb{R}^2 | L(\mathbf{x}) = 0\} ,$$

such that  $L < 0$  inside  $\Gamma$ . Note that

$$\mathbf{n} = \frac{\nabla L}{|\nabla L|} \quad (1)$$

is the outward unit normal and  $\kappa = \nabla \cdot \mathbf{n}$  is the signed curvature. The optimal contour is characterized by minimizing the Chan-Vese energy functional

$$\begin{aligned} E(c_-, c_+, \Gamma) = & \mu \int_{\Gamma} d\sigma \\ & + \lambda_- \int_{L < 0} (I(\mathbf{x}) - c_-)^2 d\mathbf{x} \\ & + \lambda_+ \int_{L > 0} (I(\mathbf{x}) - c_+)^2 d\mathbf{x} . \end{aligned} \quad (2)$$

Here  $I(\mathbf{x})$  is the gray-scale FLIP image and  $\mu \geq 0$ ,  $\lambda_{\pm} > 0$  are fixed parameters. The first integral on the right hand side measures the length of the curve and thereby has a smoothing effect. The remaining two integrals are minimized by matching the curve to the boundary of the object as well as the average gray-levels in fore- and background respectively.

The idea of Chan and Vese is to *activate* the curve by introducing an artificial time-like parameter  $\Gamma(t) = \{\mathbf{x}(t) \in \mathbb{R}^2 : L(\mathbf{x}(t), t) = 0\}$  and let the curve *evolve* towards minimal energy. By definition  $L_t = -\nabla L \cdot \mathbf{x}'$ . Evolution towards less energy means  $\mathbf{x}' = -\nabla_{\partial\Omega} E$ , where the shape gradient of the energy (2), see the appendix, is given by

$$\nabla_{\Gamma} E = [2\mu\kappa + \lambda_-(I(\mathbf{x}) - c_-)^2 - \lambda_+(I(\mathbf{x}) - c_+)^2] \cdot \mathbf{n} . \quad (3)$$

Consequently, the level set equation for the Chan-Vese active contour reads

$$L_t = [2\mu\kappa + \lambda_-(I(\mathbf{x}) - c_-)^2 - \lambda_+(I(\mathbf{x}) - c_+)^2] \cdot |\nabla L| .$$

The implementation of the Chan-Vese algorithm described in [10] is a time-stepping method for this evolutionary PDE. It is used to localize boundaries of the cell, nucleus and bleaching area in FLIP images, respectively. The strength of the algorithm is that it can distinguish several disjoint objects in one image. The method does detect objects whose boundaries are not necessarily defined by gradients. Like in *snake-methods* [11] the energy is minimized until the desired boundary is detected. However, the stopping criterion for the Chan-Vese algorithm does not depend on gradients.

The Chan-Vese algorithm applies to segment the cell from the background as well as the nucleus and the bleaching area from the cytoplasm. Both the cell- and the bleaching area are segmented from the first image while the nucleus is taken from the last image of the sequence depicted in Fig. 6.

To simulate transport kinetics across the nuclear membrane using space dependent reaction rates, the nuclear boundary is "blown up" to a small but finite thickness of 9 nm which resembles closely the thickness of the nuclear membrane [18]. From the Chan-Vese method we obtain the boundary of the nucleus represented as a polygon. Each point on this polygon is shifted in both inward- and outward normal direction to form two new polygons  $\partial\Omega_{MN}$  and  $\partial\Omega_{MC}$ , respectively. Finally, the nuclear membrane is bounded by  $\partial\Omega_M = \partial\Omega_{MN} \cup \partial\Omega_{MC}$ .

### 3.2 Transport kinetics across the membrane

The in FLIP images observed fluorescence intensity is a direct indicator for the amount of fluorescence-tagged

molecules. A typical marker is eGFP, whose number density, denoted by  $c = u + v$  is the quantity of interest in our model and subsequent simulation. Typical FLIP images show that the amount of eGFP is highest in the nuclear compartment even in the prebleach image. This suggests a higher amount of eGFP in the nucleus than in the cytoplasm at steady state (see Fig. 6). Therefore, we distinguish between the number density in the cytoplasm  $u$  and in the nucleus  $v$ . Consequently,  $c = u + v$  is a "direct" sum in the sense that  $u(t, \mathbf{x}) = 0$  in the nucleus  $\mathbf{x} \in \Omega_N$  and  $v(t, \mathbf{x}) = 0$  in  $\Omega_C$ . Transport across the membrane is modeled by a first order reversible molecular reaction



where  $k^+$  and  $k^-$  are positive reaction rates. By simple reaction kinetics, the dynamics of (4) is described by a differential system

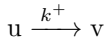
$$\frac{\partial}{\partial t} u(t, \mathbf{x}) = k^-(\mathbf{x})v(t, \mathbf{x}) - k^+(\mathbf{x})u(t, \mathbf{x}) , \quad (5)$$

$$\frac{\partial}{\partial t} v(t, \mathbf{x}) = k^+(\mathbf{x})u(t, \mathbf{x}) - k^-(\mathbf{x})v(t, \mathbf{x}) , \quad (6)$$

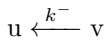
$$\mathbf{x} \in \Omega_M , \quad t > 0.$$

Obviously,  $c = u + v$  is constant and mass is conserved.

To guarantee the complete transformation of  $u$  into  $v$  across the membrane, space dependent reaction rates are needed: Near the nucleus the reaction



dominates  $k^+ \gg 1$ , while the reversed reaction is slow  $k^- \ll 1$ . On the other side of the membrane, close to the cytoplasm



dominates  $k^- \gg 1$  and  $k^+ \ll 1$ . In effect, we introduce stiff reaction rates forcing (5) and (6) into equilibrium

$$k^+(\mathbf{x})u = k^-(\mathbf{x})v .$$

On the boundary  $\partial\Omega_{MN}$   $k^-(\mathbf{x}) = 0$  and  $k^+(\mathbf{x}) \gg 1$  such that  $u = 0$  and all the tagged protein is present as  $v$ . Similarly, on the other side,  $\partial\Omega_{MC}$ ,  $k^+(\mathbf{x}) = 0$  and  $k^-(\mathbf{x}) \gg 1$  such that  $v = 0$  and all protein is transformed into  $u$ . Examples for reaction rates  $k^\pm$  are logistic functions

$$k^\pm(\mathbf{x}) = \frac{1}{\epsilon} \frac{1}{1 + \exp(\pm k \cdot \text{dist}(\mathbf{x} - \mathbf{x}_m))} , \quad (7)$$

$$0 < \epsilon \ll 1 , \quad k \gg 1 .$$

Here  $\mathbf{x}_m$  denotes the orthogonal projection of  $\mathbf{x}$  onto center of the membrane, such that  $\text{dist}(\mathbf{x} - \mathbf{x}_m)$  is the signed distance from the center. This distance is positive in direction of the outward normal (1) pointing from the nucleus into the cytoplasm. Fig. 3 displays  $\epsilon k^+$  for different values of the shape parameter  $k$  and over the typical width of a membrane: 9 nm. Note that the outward normal  $\mathbf{n}$  is positive in negative  $\mathbf{x}$ -direction.

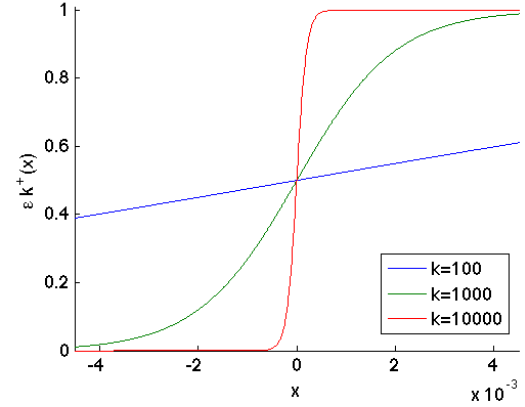


Fig. 3 Reaction rate  $\epsilon k^+$  centered at origin.

### 3.3 Reaction-diffusion dynamics

From [14] we find that the photobleaching may be modeled by the differential equation:

$$\frac{\partial}{\partial t} u(t, \mathbf{x}) = -b \frac{q}{q+1} \cdot u(t, \mathbf{x}) , \quad \mathbf{x} \in \Omega_B , t > 0 , \quad (8)$$

where  $b$  is the intrinsic bleaching rate constant and  $q$  is the equilibrium constant for the reaction between the ground and excited state for a fluorophore. Since the excitation rate constant depends on laser intensity,  $q$  also depend on the intensity. This model allows to study the effect of fluorophore photophysics and eventual fluorescence saturation [14].

Using common notation, our time-dependent PDE model reads:

$$\begin{aligned} u_t &= \nabla \cdot (\alpha \nabla u) + (k^- v - k^+ u) \Big|_{\Omega_M} - \theta b \frac{q}{1+q} u \Big|_{\Omega_B} , \\ v_t &= \nabla \cdot (\beta \nabla v) + (k^+ u - k^- v) \Big|_{\Omega_M} , \end{aligned} \quad (9)$$

$$\mathbf{x} \in \Omega , \quad t > 0 ,$$

where  $\theta$  is a 0–1 step function simulating the high-intensity laser pulses occurring at a frequency given by

the frame rate. Diffusion coefficients are inhomogeneous and piecewise defined

$$(\alpha, \beta) = \begin{cases} (\alpha, \beta)_C & \text{if } \mathbf{x} \in \Omega_C, \\ (\alpha, \beta)_M & \text{if } \mathbf{x} \in \Omega_M, \\ (\alpha, \beta)_N & \text{if } \mathbf{x} \in \Omega_N. \end{cases} \quad (10)$$

To prevent flow across the cell boundary and thereby maintain mass conservation, the model comes with non-flow, Neumann boundary conditions

$$\mathbf{n} \cdot \nabla u = \mathbf{n} \cdot \nabla v = 0, \quad \mathbf{x} \in \partial\Omega. \quad (11)$$

Finally, initial concentrations need to be specified

$$u(0, x) = u^0(x), \quad v(0, x) = v^0(x), \quad x \in \Omega. \quad (12)$$

#### 4 Analysis

For simplicity of presentation we consider a reaction-diffusion system of type (9), but in one spacial variable and with diffusion coefficients scaled to unity

$$\begin{aligned} u_t &= u_{xx} + k^-v - k^+u, \\ v_t &= v_{xx} + k^+u - k^-v, \quad x \in (0, 1), \quad t > 0. \end{aligned} \quad (13)$$

For illustration we consider homogenous Dirichlet and Neumann conditions for  $u$  and  $v$ , respectively

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

$$v_x(t, 0) = v_x(t, 1) = 0.$$

The interested reader can easily replace the Dirichlet condition by a Neumann condition as in (11). Initial data is given as in (12). The analysis uses elements by E. Bohl [3] which originally go back to R.S. Varga [16], L. Collatz [7] and E. Bohl [4].

For the analysis, the spatial derivative is discretized by a central, finite difference approximation. With homogenous Dirichlet conditions the linear operator reads

$$-u_{xx} \approx \mathbf{A}_h \mathbf{u} = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix} \mathbf{u},$$

where the vector  $u$  holds approximations on inner grid points  $u_i \approx u(x_i)$ ,  $x_i = ih$ ,  $i = 1, 2, \dots, m$ , and  $(m+1)h = 1$ . In the case of homogenous Neumann conditions  $v_0 = v_1$  and  $v_m = v_{m+1}$ , the second derivative on inner grid points is approximated by

$$-v_{xx} \approx \mathbf{B}_h \mathbf{v} = \frac{1}{h^2} \begin{pmatrix} 1 & -1 & & \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix} \mathbf{v}.$$

Semi-implicit time stepping results in a linearly coupled, discrete system

$$\begin{aligned} [(1 + \Delta t k^+) \mathbf{I} + \Delta t \mathbf{A}_h] \mathbf{u}^{n+1} &= \mathbf{u}^n + \Delta t k^- \mathbf{v}^n, \\ [(1 + \Delta t k^-) \mathbf{I} + \Delta t \mathbf{B}_h] \mathbf{v}^{n+1} &= \mathbf{v}^n + \Delta t k^+ \mathbf{u}^n. \end{aligned} \quad (14)$$

Naturally, step sizes are positive  $\Delta t$ ,  $h > 0$  and reaction rates are non-negative  $k^\pm \geq 0$ . Again,  $u_i^n$  is thought as approximation to  $u(t_n, x_i)$ . In fact one can prove that the numerical approximation converges to the solution of the PDE system.

##### 4.1 Inverse-monotonicity

The key argument relies on the fact that the finite difference operators  $\mathbf{I} + \Delta t \mathbf{A}_h$  as well as  $\mathbf{I} + \Delta t \mathbf{B}_h$  are inverse-monotone with bounded inverse. In fact  $\mathbf{A}_h$  inherits its inverse-monotonicity from the negative Laplacian, while  $\mathbf{B}_h$  is obviously singular.

**Lemma 1** *With non-negative parameters  $\Delta t \geq 0$ ,  $h > 0$  and reaction rates  $k^\pm \geq 0$  both system matrices  $(1 + \Delta t k^+) \mathbf{I} + \Delta t \mathbf{A}_h$  and  $(1 + \Delta t k^-) \mathbf{I} + \Delta t \mathbf{B}_h$  are regular with non-negative and uniformly bounded inverse.*

An obvious consequence is:

**Corollary 1** *With non-negative initial data  $u^0 \geq 0$  and  $v^0 \geq 0$  and reaction rates  $k^\pm \geq 0$ , approximations computed by (14) remain non-negative for all time  $n\Delta t \geq 0$ .*

The proof of Lemma 1 is based on inverse-monotone  $Z$ -matrices. A  $Z$ -matrix is characterized by its non-positive off-diagonal elements:

$$\mathbf{Z} = (z_{ij}), \quad z_{ij} \leq 0, \quad i \neq j.$$

A matrix  $\mathbf{A}$  is called inverse-monotone if it is invertible with elementwise non-negative inverse  $\mathbf{A}^{-1} \geq 0$ . In this case  $\mathbf{A}\mathbf{e} \geq 0$  implies  $\mathbf{e} \geq 0$ ; i.e. solving the system preserves positivity. An inverse-monotone  $Z$ -matrix is called  $M$ -matrix.

Any strictly positive vector  $\mathbf{p} > 0$  defines a weighted max-norm

$$\|\mathbf{x}\|_p = \max_i \{|x_i|/p_i\}.$$

The corresponding operator norm for a monotone (i.e. elementwise positive) matrix is  $\|\mathbf{A}\|_p = \|\mathbf{A}\mathbf{p}\|_p$ :

**Lemma 2** *For  $0 < \mathbf{p} \in \mathbb{R}^n$  and  $0 \leq \mathbf{A} \in \mathbb{R}^{n \times n}$*

$$\|\mathbf{A}\|_p = \|\mathbf{A}\mathbf{p}\|_p. \quad (15)$$

**Proof** of Lemma 2: Let  $\mathbf{x} \in \mathbb{R}^n$  be arbitrary. As  $\mathbf{A} \geq 0$ ,

$$\frac{|(\mathbf{Ax})_i|}{p_i} \leq \sum_j \left( \frac{a_{ij}p_j}{p_i} \frac{|x_j|}{p_j} \right) \leq \sum_j \frac{a_{ij}p_j}{p_i} \|\mathbf{x}\|_p .$$

Hence,

$$\|\mathbf{Ax}\|_p = \max_i \frac{|(\mathbf{Ax})_i|}{p_i} \leq \max_i \frac{(\mathbf{Ap})_i}{p_i} \|\mathbf{x}\|_p .$$

Again by positivity  $\mathbf{Ap} \geq 0$ ,  $\|\mathbf{Ax}\|_p \leq \|\mathbf{Ap}\|_p \|\mathbf{x}\|_p$  and  $\|\mathbf{A}\|_p \leq \|\mathbf{Ap}\|_p$ . The reversed inequality is obvious, and the proof is complete.  $\square$

**Lemma 3** *A Z-matrix  $\mathbf{A}$  is M-matrix if and only if  $\mathbf{A}$  is semi-positive. That is, there exists  $\mathbf{e} > 0$  with  $\mathbf{Ae} > 0$ .*

**Proof** of Lemma 3: The proof of Lemma 3 (in German) can be found in E. Bohl, Satz 5.1 [3].

1. Let  $\mathbf{A}$  be M-matrix. Then  $\mathbf{e} = \mathbf{A}^{-1}\delta > 0$  and  $\mathbf{Ae} = \delta > 0$ . Here  $\delta = (1, \dots, 1)^T$  denotes the vector consisting of just ones.

2. Let  $\mathbf{e} > 0$  and  $\mathbf{Ae} > 0$ . As  $\mathbf{A}$  is Z-matrix

$$(\mathbf{Ae})_i = a_{ii}e_i - \sum_{j \neq i} |a_{ij}|e_j > 0$$

and hence  $a_{ii} > 0$ . Consequently,  $\mathbf{D} = \text{diag}(a_{ii})$  is regular and

$$\mathbf{P} = \mathbf{D}^{-1}(\mathbf{D} - \mathbf{A}) \geq 0, \quad \mathbf{A} = \mathbf{D}(\mathbf{I} - \mathbf{P}) .$$

Further,  $(\mathbf{I} - \mathbf{P})\mathbf{e} = \mathbf{D}^{-1}\mathbf{Ae} > 0$  and  $\mathbf{Pe} < \mathbf{e}$ . By (15)  $\|\mathbf{P}\|_e < 1$ . By the Banach fixed-point theorem  $\mathbf{I} - \mathbf{P}$  is regular and

$$(\mathbf{I} - \mathbf{P})^{-1} = \sum_{j=0}^{\infty} \mathbf{P}^j \geq 0 .$$

Therefore,

$$\mathbf{A}^{-1} = (\mathbf{I} - \mathbf{P})^{-1}\mathbf{D}^{-1} = \sum_{j=0}^{\infty} \mathbf{P}^j \mathbf{D}^{-1} \geq 0 .$$

The proof is complete.  $\square$

**Lemma 4** *Let  $\mathbf{A}$  be M-matrix and  $\mathbf{D} = \text{diag} \geq 0$ . Then  $\mathbf{A} + \mathbf{D}$  is M-matrix.*

**Proof** of Lemma 4: As  $\mathbf{A}$  is M-matrix, there is  $\mathbf{e} > 0$  such that  $\mathbf{Ae} > 0$ . As  $\mathbf{D}$  is diagonal,  $\mathbf{A} + \mathbf{D}$  is Z-matrix. As  $\mathbf{D} \geq 0$ ,  $(\mathbf{A} + \mathbf{D})\mathbf{e} > \mathbf{De} \geq 0$ . By Lemma 3,  $\mathbf{A} + \mathbf{D}$  is M-matrix.  $\square$

**Proof** of Lemma 1: Note  $\mathbf{A}_h\delta = (1, 0, \dots, 0, 1)/h^2$  and  $\mathbf{B}_h\delta = 0$ . Obviously  $\mathbf{B}_h$  is singular. Both  $\mathbf{I} + \Delta t\mathbf{A}_h$  and  $\mathbf{I} + \Delta t\mathbf{B}_h$  are Z-matrices. Both are semi-positive

$$(\mathbf{I} + \Delta t\mathbf{A}_h)\delta \geq (\mathbf{I} + \Delta t\mathbf{B}_h)\delta \geq \delta > 0 .$$

By Lemma 3  $\mathbf{I} + \Delta t\mathbf{A}_h$  and  $\mathbf{I} + \Delta t\mathbf{B}_h$  are M-matrices. We may add any positive diagonal matrix  $\Delta t\mathbf{kI}$  and the M-property is preserved by Lemma 4. Thus both system matrices are M-matrices, even with non-constant, non-negative reaction rates  $k^\pm$ . It remains to show the uniform bound for the inverse. We have that

$$[(1 + \Delta tk^+)\mathbf{I} + \Delta t\mathbf{A}_h]\delta \geq \delta ,$$

$$[(1 + \Delta tk^-)\mathbf{I} + \Delta t\mathbf{B}_h]\delta \geq \delta .$$

Multiplying by the non-negative inverse, taking norms and applying Lemma 2 the desired bound follows

$$\|[(1 + \Delta tk^+)\mathbf{I} + \Delta t\mathbf{A}_h]^{-1}\|_\delta \leq 1 ,$$

$$\|[(1 + \Delta tk^-)\mathbf{I} + \Delta t\mathbf{B}_h]^{-1}\|_\delta \leq 1$$

and the proof is complete.  $\square$

Note that  $\|\cdot\|_\delta$  is the well-known max-norm. Via the eigensystem of  $\mathbf{A}_h$  one can also show that  $\mathbf{A}_h$  itself is a M-matrix.

## 4.2 Stability and convergence

The uniform bound of the inverse system matrix implies stability of the numerical method with respect to errors in the initial data  $\Delta \mathbf{u}^0$  and  $\Delta \mathbf{v}^0$ . By linearity these errors are governed by the system (14) itself

$$[(1 + \Delta tk^+)\mathbf{I} + \Delta t\mathbf{A}_h]\Delta \mathbf{u}^{n+1} = \Delta \mathbf{u}^n + \Delta tk^- \Delta \mathbf{v}^n ,$$

$$[(1 + \Delta tk^-)\mathbf{I} + \Delta t\mathbf{B}_h]\Delta \mathbf{v}^{n+1} = \Delta \mathbf{v}^n + \Delta tk^+ \Delta \mathbf{u}^n .$$

Uniform boundedness implies

$$\|\Delta \mathbf{u}^{n+1}\|_\delta \leq \|\Delta \mathbf{u}^n\|_\delta + \Delta tk^- \|\Delta \mathbf{v}^n\|_\delta ,$$

$$\|\Delta \mathbf{v}^{n+1}\|_\delta \leq \|\Delta \mathbf{v}^n\|_\delta + \Delta tk^+ \|\Delta \mathbf{u}^n\|_\delta ,$$

and

$$\begin{aligned} & \|\Delta \mathbf{u}^{n+1}\|_\delta + \|\Delta \mathbf{v}^{n+1}\|_\delta \\ & \leq (1 + \Delta tK) (\|\Delta \mathbf{u}^n\|_\delta + \|\Delta \mathbf{v}^n\|_\delta) , \end{aligned}$$

where  $K = \max(k^+, k^-)$ . Finally,

$$\begin{aligned} & \|\Delta \mathbf{u}^n\|_\delta + \|\Delta \mathbf{v}^n\|_\delta \\ & \leq (1 + \Delta tK)^n (\|\Delta \mathbf{u}^0\|_\delta + \|\Delta \mathbf{v}^0\|_\delta) , \end{aligned} \tag{16}$$

with  $(1 + \Delta t K)^n \leq \exp(n \Delta t K) \leq \exp(TK)$ ,  $n \Delta t \leq T$ .

The inequality (16) expresses stability in the sense that errors at later time depend continuously on initial errors. Convergence towards a smooth solution of the reaction–diffusion system (13) follows from the discrete Gronwall lemma.

## 5 Implementation and Simulation

### 5.1 Numerical method

In complex geometries, like segmented FLIP cell images, it is convenient to generate finite element meshes using Gmsh [8]. The Finite Element Method is automated in the FEniCS software [13]. All one needs to specify to the FEniCS compiler is the weak form of our FLIP model (9).

We discretize the time derivative by an backward Euler step. Any higher order but  $L$ -stable method, like f.ex. certain SDIRK schemes [9], are appropriate as well. Let  $\phi$  and  $\psi$  be bilinear Lagrange test–functions for  $u$  and  $v$  respectively. After multiplication and partial integration the weak form reads:

$$\begin{aligned} & \int_{\Omega} \frac{u^{n+1} - u^n}{\Delta t} \phi \, d\mathbf{x} + \int_{\Omega} \alpha \nabla u^{n+1} \cdot \nabla \phi \, d\mathbf{x} \\ & + \int_{\Omega_M} (k^+ u^{n+1} - k^- v^{n+1}) \phi \, d\mathbf{x} \\ & + \int_{\Omega_B} \theta b \frac{q}{q+1} u^{n+1} \phi \, d\mathbf{x} = 0 \quad , \end{aligned} \quad (17)$$

$$\begin{aligned} & \int_{\Omega} \frac{v^{n+1} - v^n}{\Delta t} \psi \, d\mathbf{x} + \int_{\Omega} \beta \nabla v^{n+1} \cdot \nabla \psi \, d\mathbf{x} \\ & + \int_{\Omega_M} (k^- v^{n+1} - k^+ u^{n+1}) \psi \, d\mathbf{x} = 0 \quad . \end{aligned} \quad (18)$$

For faster execution, the bilinear form is pre-assembled. The pulsating laser beam is realized by pre-assembling two systems, with and without the bleaching term and iterating both. To resolve the bleaching effect, the bleaching interval is a multiple of the time step:  $\Delta t_b = n \Delta t$ .

### 5.2 Tuning reaction– and diffusion parameters

To find reasonable reaction–diffusion rates, consider a one–dimensional model problem along the cross–section of a cell membrane of 9nm width.

$$\begin{aligned} u_t &= (\alpha u_x)_x + k^- v - k^+ u \quad , \\ v_t &= (\beta v_x)_x + k^+ u - k^- v \quad , \quad x \in \Omega \quad , \quad t > 0 \quad . \end{aligned}$$

Measuring  $x$  in  $\mu\text{m}$ , the domain of interest is the interval  $\Omega = [0, 0.029]$  where the membrane of width 9 nm is located at  $\Omega_M = [0.01, 0.019]$ , cytoplasm  $\Omega_C = [0, 0.01]$ , and nucleus  $\Omega_N = (0.019, 0.029]$ . As initial data we set

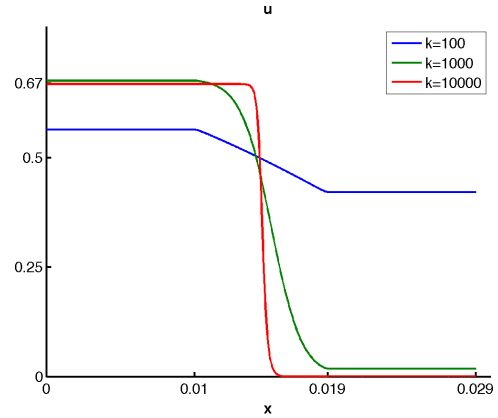
$$u^0 = \begin{cases} 1 & \text{if } x \in \Omega_C \\ 0.5 & \text{if } x \in \Omega_M \\ 0 & \text{if } x \in \Omega_N \end{cases} \quad , \quad v^0 = \begin{cases} 0 & \text{if } x \in \Omega_C \\ 0.5 & \text{if } x \in \Omega_M \\ 1 & \text{if } x \in \Omega_N \end{cases} .$$

Note that  $u^0 + v^0 = 1$ . Diffusion rates are set to

$$\alpha = \begin{cases} 25 & \text{if } x \in \Omega_C \\ 5 \cdot 10^{-4} & \text{if } x \in \Omega_M \\ 10^{-3} & \text{if } x \in \Omega_N \end{cases} \quad , \quad (19)$$

$$\beta = \begin{cases} 10^{-3} & \text{if } x \in \Omega_C \\ 2.5 \cdot 10^{-4} & \text{if } x \in \Omega_M \\ 25 & \text{if } x \in \Omega_N \end{cases} . \quad (20)$$

The stiffness parameter  $\epsilon$  in (7) is set to  $\epsilon = 10^{-4}$ . For different values of the shape coefficient we observe the eGFP number densities as depicted in Figs. 4 and 5 at time  $t = 5$ .

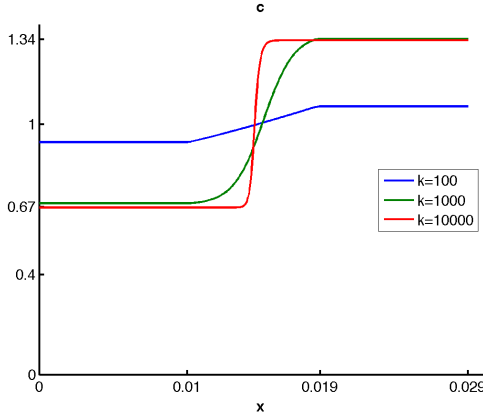


**Fig. 4**  $u$  at time  $t = 5$  sec.

In agreement with Corollary 1, positivity is preserved — a convenient property when computing number densities. From Fig. 4 we conclude that for  $u$  and  $v$  to vanish in the cytoplasm and the nucleus respectively, values of  $k \approx 10^4$  and  $\epsilon \lesssim 10^{-4}$  are sufficient.

The higher amount of total eGFP in the nucleus as seen in Fig. 5, is caused by the trans–membrane kinetics. It is confirmed in our two–dimensional simulation and agrees with typical FLIP images. See Section 5.3. It might be worth noting that the relation of fluorescence intensities in cytoplasm and nucleus agrees with





**Fig. 5**  $c = u + v$  at time  $t = 5$  sec.

the ratio of diffusion rates in the membrane

$$2 = \frac{\alpha_M}{\beta_M} \approx \frac{[c_N]}{[c_C]} . \quad (21)$$

### 5.3 Simulation of FLIP images



**Fig. 11** Finite element mesh on Chan–Vese active contours.

Figure 6 shows a sequence of FLIP images in McArde RH7777 cells. Enhanced green fluorescent protein (eGFP) stably expressed in the cells was repeatedly bleached on temperature controlled ( $35 \pm 1^\circ\text{C}$ ) stage of a Zeiss LSM 510 confocal microscope using the 488-nm line of an Argon laser as excitation source. Bleaching was performed for 5 iterations in the indicated region (circle), while images were acquired with 0.5% laser power to avoid bleaching during image recording. The total frame rate inclusive bleaching was 1.6 sec. Representative frames (every 5th frame) are shown using a

FIRE-LUT for color-coding, in which high and low intensities are yellow/white and violet/blue, respectively.

To simulate the experiment, we compute the FLIP model (7), (9) and (10) on the mesh displayed in Fig. 11. The mesh, consisting of 221558 triangles (68534 located in cytoplasm, 114 in bleaching area, 89077 in membrane, and 63833 in nucleus), was generated by Gmsh [8] on the Chan–Vese contours. The bleaching area is visible in the upper left corner. The model parameters are:  $\epsilon = 10^{-4}$ ,  $k = 10^4$ ,  $bq/(1+q) = 160$  with diffusion rates given by (19) and (20). Initial data as seen in Fig. 7 are logistic functions across the membrane

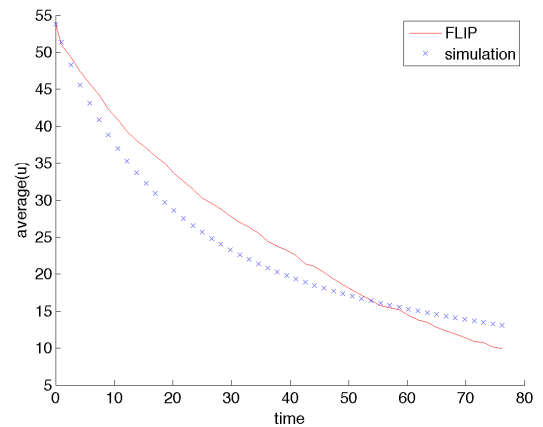
$$u^0 = \frac{53.9}{1 + \exp(-k \cdot \text{dist}(\mathbf{x} - \mathbf{x}_m))} ,$$

$$v^0 = \frac{119.4}{1 + \exp(k \cdot \text{dist}(\mathbf{x} - \mathbf{x}_m))} .$$

The max-values are taken as intensity averages from the initial FLIP image. The bleaching time interval is  $\Delta t_b = 0.4$  sec, the recovery phase lasts for 1.2 sec resulting in a total frame rate of 1.6 sec. The time step is chosen as  $\Delta t = 0.1$  sec.

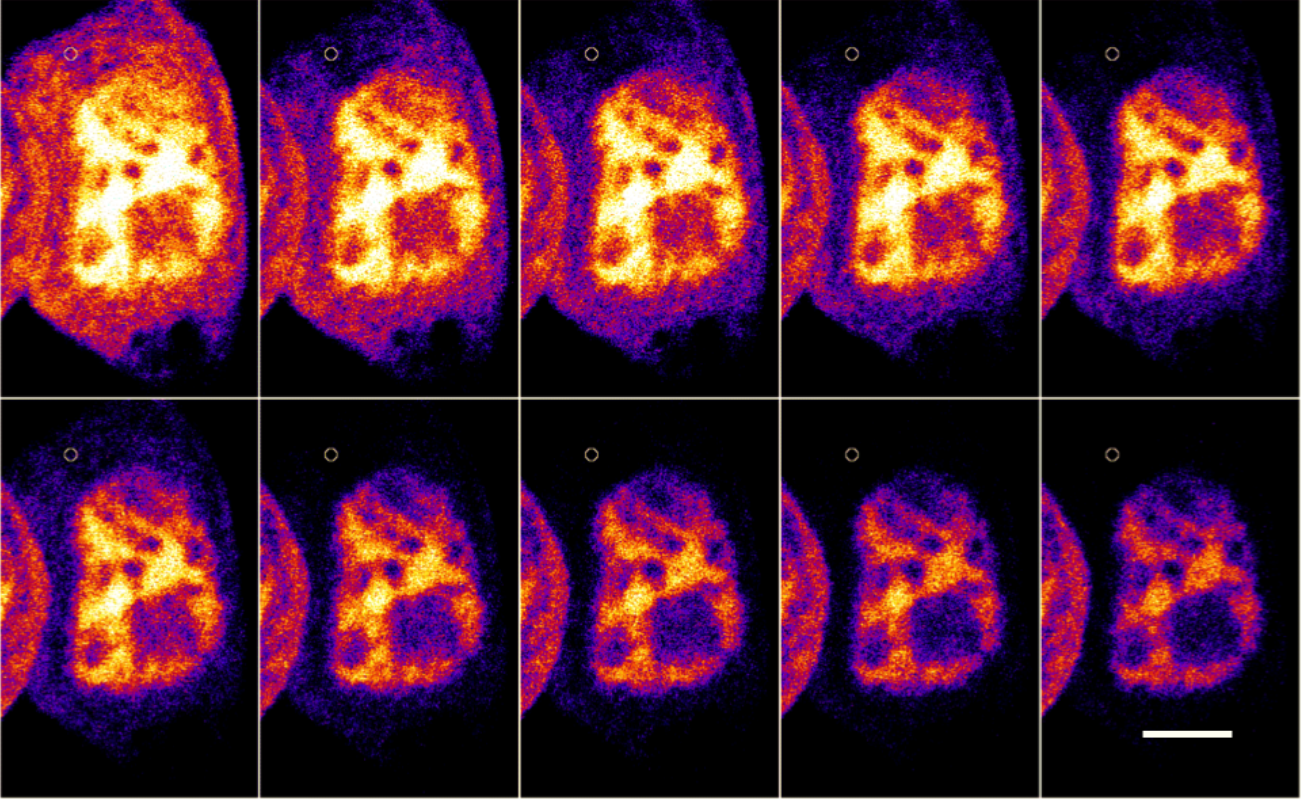
In both the FLIP images and the simulation, we observe a stronger effect of bleaching in the cytoplasm than in the nucleus — the membrane shows its protective effect.

In FLIP images we observe spatially resolved structures especially within the nucleus. Our model however, based on piecewise constant diffusion coefficients approximates averages within the different compartments. When given spatially resolved diffusion parameters, the model will of course develop structures.

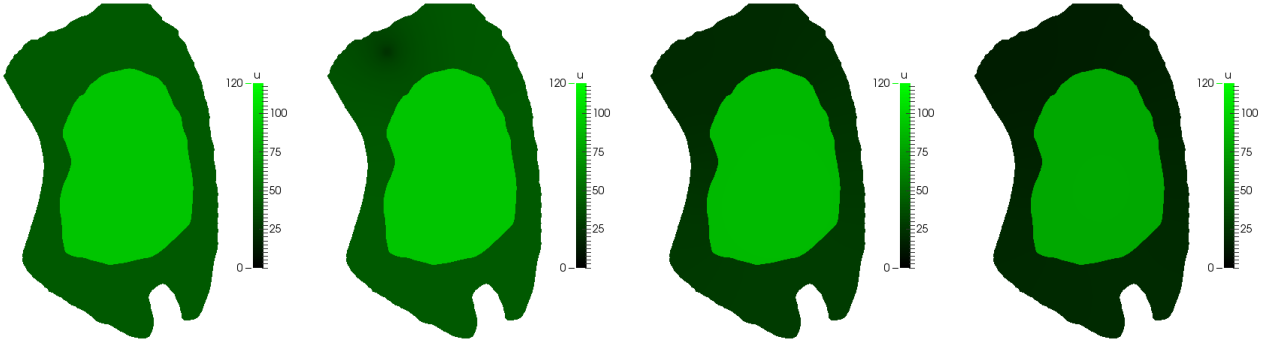


**Fig. 12** Average fluorescence intensity in cytoplasm.

Simulated average intensities are depicted in Figs. 12 and 13. Even though simulated intensities follow the observations, initially the intensity in the cytoplasm



**Fig. 6** Fluorescence loss in photobleaching of eGFP in McArdle RH7777 cells. Representative frames in FIRE-LUT color-coding. White bar 5  $\mu\text{m}$ .



**Fig. 7** before, and ...

**Fig. 8** during first bleach

**Fig. 9**  $t = 17$  sec.

**Fig. 10**  $t = 33$  sec.

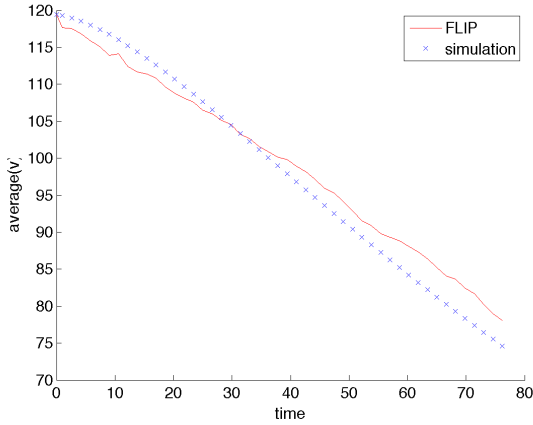
drops faster than observed in FLIP images. This defect may be attenuated by further (automated) tuning of reaction and diffusion rates in future work. Observed FLIP intensities are linearly correlated with the computational model, as seen in Figs 15 and 14. The correlation coefficients for cytoplasm and nucleus are 0.980 and 0.998, respectively.

In Section 5.2 we observed that the ratio of average fluorescence intensities in the cytoplasm and nucleus is approximately determined by the ratio of membrane diffusion rates (21). This observation is confirmed in

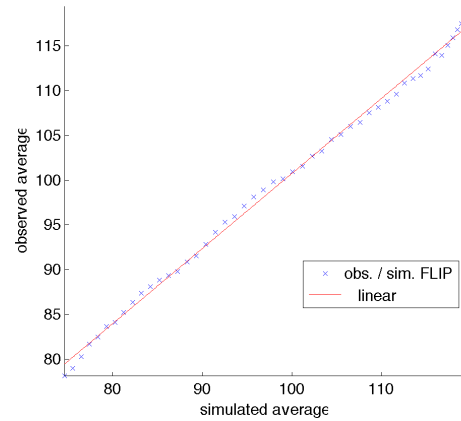
a 2D simulation when bleaching is turned off  $b = 0$ . In Fig. 16 it is seen that the ratio of intensities approaches the ratio of diffusion rates  $\frac{\alpha_M}{\beta_M} = 2$ .

## 6 Conclusion

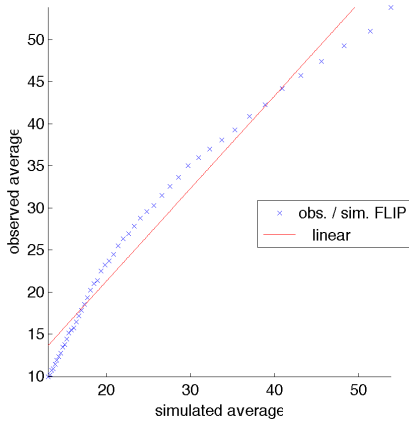
We describe a computational reaction-diffusion model of FLIP image sequences of eGFP transport between the cytoplasm and nucleus of mammalian cells. Our model performs these simulations in the real cell geometry extracted from microscopy data gathered on



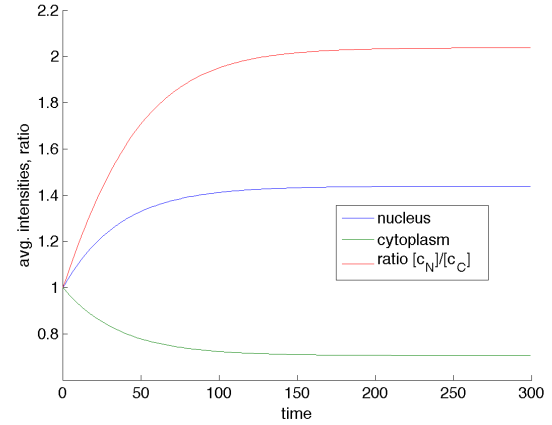
**Fig. 13** Average fluorescence intensity in nucleus.



**Fig. 15** Average fluorescence intensity in nucleus.



**Fig. 14** Average fluorescence intensity in cytoplasm.



**Fig. 16** Average intensities and ratio over time.

a confocal microscope. We found that the transport term describing nucleocytoplasmic exchange of eGFP must include rate coefficients being a function of distance along the normal to the compartment boundaries. Transport of proteins through the nuclear pore complex has been modeled previously as diffusion in an energy landscape. This results naturally in space-dependent transport rates [18]. We prove positivity and numerical stability of the model. Stability implies that approximate solutions computed by the FLIP model will converge towards correct solution of the PDE model which itself is an approximation to reality. Also, we found that our simulation results are highly correlated with experimental observations.

Interestingly, the model simulations suggest that a slowed diffusion of eGFP from the nucleus into the cytoplasm compared to the opposite direction can account for a higher residence of the protein in the nuclear compartment at steady state.

To establish a uniformly second order method in space and time, the backward Euler-step is to be re-

placed by any L-stable scheme, for example the two stage SDIRK method [9]. The present model for membrane transport is only the simplest of its kind. The investigation of advanced and nonlinear models is subject of future studies. We plan to apply the reliable FLIP simulator in inverse problems analysis to estimate reaction and diffusion parameters and thus to automatically calibrate the model to experimentally observed images.

To approximate fluorescence intensity across the nuclear membrane, both the discontinuous finite element method and/or a partition of unity / extended finite element method might be appropriate alternatives [2, 13].

## 7 Appendix: Shape gradients

The Chan-Vese energy (2) consists of two parts: Minimizing the length of the curve has some regularizing effect  $E_{\text{reg}}(\Gamma) = \int_{\Gamma} d\sigma$ . Without loss of generality, we may assume parametrization with respect to arc-length

$$\Gamma = \gamma(\sigma) : [0, l] \rightarrow \mathbb{R}^2$$

such that  $|\gamma'| = 1$  and  $l = \int_{\Gamma} d\sigma = \int_0^l |\gamma'(\sigma)|^2 d\sigma$ . Let  $\omega$  be another closed,  $C^1$ -curve  $\omega : [0, l] \rightarrow \mathbb{R}^2$  such that  $\omega(0) = \omega(l)$  and  $\omega'(0) = \omega'(l)$ . Consider an  $\epsilon$ -variation of the curve  $\Gamma$  given by  $\gamma_\epsilon = \gamma + \epsilon\omega$ . The variation of  $E_{\text{reg}} = \int_0^l |\gamma'(\sigma)|^2 d\sigma$  is defined as

$$\delta E_{\text{reg}}(\gamma, \omega) = \left. \frac{d}{d\epsilon} E_{\text{reg}}(\gamma_\epsilon) \right|_{\epsilon=0}$$

By differentiation we find

$$\begin{aligned} \delta E_{\text{reg}}(\gamma, \omega) &= \left. \frac{d}{d\epsilon} \int_0^l |\gamma'(\sigma) + \epsilon\omega'(\sigma)|^2 d\sigma \right|_{\epsilon=0} \\ &= 2 \int_0^l (\gamma'(\sigma) + \epsilon\omega'(\sigma))\omega'(\sigma) d\sigma \Big|_{\epsilon=0} \\ &= 2 \int_0^l \gamma'(\sigma)\omega'(\sigma) d\sigma . \end{aligned}$$

By partial integration, using periodicity of  $\gamma'$  and  $\omega$

$$\delta E_{\text{reg}}(\gamma, \omega) = -2 \int_0^l \gamma''(\sigma)\omega(\sigma) d\sigma .$$

By definition, the  $L^2$ -shape gradient of  $E_{\text{reg}}$  is

$$\nabla_{\Gamma} E_{\text{reg}}(\gamma) = -2\gamma'' .$$

In arc-length parametrization  $\gamma'' = -\kappa \cdot \mathbf{n}$ , where  $\mathbf{n}$  is the outward unit normal. Finally

$$\nabla_{\Gamma} E_{\text{reg}}(\gamma) = 2\kappa \mathbf{n} . \quad (22)$$

The fitting energy is

$$E_{\text{fit}}(\gamma) = \int_{\text{int}\Gamma} (I(\mathbf{x}) - c_-)^2 d\mathbf{x} + \int_{\text{ext}\Gamma} (I(\mathbf{x}) - c_+)^2 d\mathbf{x} .$$

The integral over the exterior segment equals the integral over the entire image without the interior part

$$\begin{aligned} E_{\text{fit}}(\gamma) &= \int_{\text{int}\Gamma} (I(\mathbf{x}) - c_-)^2 - (I(\mathbf{x}) - c_+)^2 d\mathbf{x} \\ &\quad + \int_{\text{int}\Gamma \cup \text{ext}\Gamma} (I(\mathbf{x}) - c_+)^2 d\mathbf{x} . \end{aligned}$$

The last integral does not depend on the curve  $\Gamma$  and the first one is subject of the next Lemma.

**Lemma 5** *Let  $\gamma$  and  $\omega$  be closed  $C^1$ -curves. Let  $V : \mathbb{R}^2 \rightarrow \mathbb{R}$  be integrable. The variation of*

$$E(\gamma) = \int_{\text{int}\gamma} V(\mathbf{x}) d\mathbf{x}$$

is

$$\delta E(\gamma, \omega) = \int_0^l V(\gamma(\sigma))\mathbf{n}(\sigma) \cdot \omega(\sigma) d\sigma .$$

The proof below shows that for Lemma 5 to hold true, arc-length parametrization is not necessary but convenient. Otherwise the unit normal  $\mathbf{n}$  is to be replaced by the non-unit outward normal. As a result we have that the  $L^2$ -shape gradient is given by

$$\nabla_{\Gamma} E(\gamma) = V(\gamma)\mathbf{n} . \quad (23)$$

**Proof** of Lemma 5:

Let  $P(x_1, x_2) = 0$  and  $Q(x_1, x_2) = \int_0^{x_1} V(\xi, x_2) d\xi$ .

$$E(\gamma) = \int_{\text{int}\gamma} V(\mathbf{x}) d\mathbf{x} = \int_{\text{int}\gamma} \frac{\partial Q}{\partial x_1} - \frac{\partial P}{\partial x_2} dx_1 dx_2 .$$

By Green's formula

$$E(\gamma) = \int_{\gamma} Q d\mathbf{x}_2 .$$

In the last curve integral  $\mathbf{x} = \gamma$ ,  $d\mathbf{x}_2 = \gamma'_2 d\sigma$  and thus

$$E(\gamma) = \int_0^l Q(\gamma(\sigma))\gamma'_2(\sigma) d\sigma .$$

Consider the  $\epsilon$ -variation  $\gamma_\epsilon = \gamma + \epsilon\omega$  and find

$$\begin{aligned} \delta E(\gamma, \omega) &= \left. \frac{d}{d\epsilon} E(\gamma_\epsilon) \right|_{\epsilon=0} \\ &= \left. \frac{d}{d\epsilon} \int_0^l Q(\gamma_\epsilon)(\gamma'_\epsilon)_2 d\sigma \right|_{\epsilon=0} \\ &= \left. \int_0^l \nabla Q(\gamma_\epsilon) \cdot \omega(\gamma'_\epsilon)_2 + Q(\gamma_\epsilon)\omega'_2 d\sigma \right|_{\epsilon=0} \\ &= \int_0^l \nabla Q(\gamma) \cdot \omega\gamma'_2 + Q(\gamma)\omega'_2 d\sigma . \end{aligned}$$

By partial integration and periodicity

$$\begin{aligned} \delta E(\gamma, \omega) &= \int_0^l \nabla Q(\gamma) \cdot \omega\gamma'_2 - \frac{d}{d\sigma} (Q(\gamma))\omega_2 d\sigma \\ &= \int_0^l \nabla Q(\gamma) \cdot \omega\gamma'_2 - \nabla Q(\gamma) \cdot \gamma'\omega_2 d\sigma \\ &= \int_0^l \frac{\partial Q}{\partial x_1} \mathbf{n}\omega d\sigma \\ &= \int_0^l V(\gamma)\mathbf{n}\omega d\sigma . \end{aligned}$$

Here the (unit) normal is given by  $\mathbf{n} = (\gamma'_2, -\gamma'_1)^T$  and our proof is complete.  $\square$

The shape gradient of the Chan-Vese energy (3) follows from (22) and (23) where  $V(\mathbf{x}) = (I(\mathbf{x}) - c_-)^2 - (I(\mathbf{x}) - c_+)^2$ .

## Acknowledgement

The authors want to thank Niels Christian Overgaard from Lund Univeristy for introducing us to level set methods in image segmentation.

## References

1. Aridor M., Hannan L.A.: Traffic Jams II: An Update of Diseases of Intracellular Transport. *Traffic*, 3(11), pp. 781-790 (2002)
2. Babuška I., Banerjee U., and Osborn J.E.: Survey of meshless and generalized finite element methods: A unified approach, *Acta Numerica* 12, 1–125 (2003)
3. Bohl E.: Finite Modelle gewöhnlicher Randwertaufgaben, *Teubner, Stuttgart* (1981)
4. Bohl E.: Monotonie: Lösbarkeit und Numerik bei Operatorgleichungen, *Springer Tracts in Natural Philosophy* 25, Berlin (1974)
5. Braga J., Desterro J.M., and Carmo-Fonseca M.: Intracellular macromolecular mobility measured by fluorescence recovery after photobleaching with confocal laser scanning microscopes, *Mol. Biol. Cell* 15: 4749-4760 (2004)
6. Chan T. and Vese L.: Active Contours Without Edges, *IEEE transactions on image processing* vol. 10, no. 2, pp. 266-277 (2001)
7. Collatz L.: Funktionalanalysis und Numerische Mathematik, *Springer, Berlin* (1964)
8. Geuzaine Ch. and Remacle J.F.: Gmsh, <http://geuz.org/gmsh>
9. Hairer E. and Wanner G.: Solving Ordinary Differential Equations II, *Springer, Heidelberg* (1996)
10. Hansen Ch.V.: Segmentation of Fluorescent Microscopy Images of Living Cells, Bachelor Project, IMADA, SDU (2012)
11. Kass M., Witkin A., Terzopoulos D.: Snakes: Active Contour Models, *Int Journal of Computer Vision*, pp.321-331 (1987)
12. Lippincott-Schwartz J., Altan-Bonnet N., Patterson G.H.: Photobleaching and photoactivation: following protein dynamics in living cells., *Nat. Cell Biol., Suppl.* 7 (2003), S7-14 (2003)
13. Logg A., Mardal K.-A., Wells G.N. et al.: Automated Solution of Differential Equations by the Finite Element Method, *Springer, Heidelberg* (2012)
14. Lund F.W. and Wüstner D.: A comparison of single particle tracking and temporal image correlation spectroscopy for quantitative analysis of endosome motility, *Journal of Microscopy*, Vol. 252, pp. 169-188 (2013)
15. Müller F., D. Mazza D., Stasevich T.J., and McNally J.G.: FRAP and kinetic modeling in the analysis of nuclear protein dynamics: what do we really know? *Curr. Opin. Cell Biol.* 22, pp. 403-411 (2010)
16. Varga R.S.: Matrix iterative analysis, *Prentice-Hall, Englewood Cliffs, N.J.* (1962)
17. Wüstner D., Solanko L.M., Lund F.W., Sage D., Schroll H.J. and Lomholt M.A.: Quantitative Fluorescence Loss in Photobleaching for Analysis of Protein Transport and Aggregation. *BMC Bioinformatics* 2012, 13:296 (2012)
18. Zilman A., Talia S.D., Chait B.T., Rout M.P., Magnasco M.O.: Efficiency, selectivity, and robustness of nucleocytoplasmic transport. *PLoS Comp. Biol.* 3(7), e125 (2007)

# Bibliography

- [1] Babuška I., Banerjee U., and Osborn J.E.: Survey of meshless and generalized finite element methods: A unified approach, *Acta Numerica* 12, 1–125 (2003)
- [2] M. de Berg, O. Cheong, M. van Kreveld, M. Overmars, *Computational Geometry - Algorithms and Applications*, Third Edition, Springer, (2008).
- [3] E. Bohl, *Finite Modelle gewöhnlicher Randwertaufgaben*, Teubner, Stuttgart, (1981).
- [4] E. Bohl and J. Lorenz *Inverse monotonicity and difference schemes of higher order. A summary for two-point boundary value problems*, Birkhäuser Verlag, Basel, *Aequationes Mathematicae* 19: 1-36, (1979).
- [5] A. N. Carvalho, J. A. Cuminato, *Reaction-diffusion problems in cell tissues*, *Journal of Dynamics and Differential Equations*, 9(1):93–131, (1997).
- [6] T. Chan and L. Vese, *Active Contours Without Edges*, *IEEE transactions on image processing* vol. 10, no. 2, , pp. 266-277, (2001).
- [7] J. G. Charney, R. Fjørtoft, J. von Neumann, *Numerical Integration of the Barotropic Vorticity Equation*, *Tellus* 2: 237-254, (1950).
- [8] Y. Chen et al., *Probing Nucleocytoplasmic Transport by Two-Photon Activation of PA-GFP*, *Microsc. Res. Tech.* 69:220-226, (2006).
- [9] J. Crank, *The Mathematics of Diffusion*, second edition, Oxford Science Publications, (1975).
- [10] G. G. Dahlquist, *A special stability problem for linear multistep methods*, Volume 3, Issue 1, pp 27-43, (1963).
- [11] K. Eriksson, D. Estep, P. Hansbo, C. Johnson, *Computational Differential Equations*, Studentlitteratur, (1996).
- [12] C. Geuzaine, J. F. Remacle, *Gmsh Reference Manual* , (2014).
- [13] D. Gilbarg, N. S. Trudinger, *Elliptic Partial Differential Equations of Second Order*, Springer-Verlag Berlin Heidelberg GmbH, (1977).
- [14] <http://geuz.org/gmsh/>
- [15] Ch.V. Hansen, *Segmentation of Fluorescent Microscopy Images of Living Cells*, Bachelor Project, IMADA, SDU, (2012).

- [16] E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II*, Springer, (2010), first print 1991.
- [17] K. Hayashida, *On the uniqueness in Cauchy's problem for elliptic equations*, Publ. Res. Inst. Math. Sci., 2 . pp. 429-449, (1966).
- [18] W. Hundsdorfer, J. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer, (2003).
- [19] C. Johnson, *Numerical solution of partial differential equations by the finite element method*, Studentlitteratur, Lund, (1990).
- [20] M. Kass, A. Witkin, D. Terzopoulos, *Snakes: Active Contour Models*, Int Journal of Computer Vision, pp.321-331, (1987).
- [21] J. Lippincott-Schwartz and G. H. Patterson *Development and Use of Fluorescent Protein Markers in Living Cells*, Science 300, 87 (2003).
- [22] A. Logg, K. A. Mardal, G. N. Wells, *Automated Solution of Differential Equations by the Finite Element Method - The FEniCS Book*, Springer, (2012).
- [23] F. W. Lund and D. Wüstner, *A comparison of single particle tracking and temporal image correlation spectroscopy for quantitative analysis of endosome motility*, Journal of Microscopy, Vol. 252, pp. 169-188, (2013).
- [24] A.a.H. Meir, L.A., *Traffic Jams II: An Update of Diseases of Intracellular Transport*, Traffic, 3 (2002) 781-790.
- [25] R. Peters, *Translocation Through the Nuclear Pore Complex: Selectivity and Speed by Reduction-of-Dimensionality*, Traffic; 6: 421-427, (2006).
- [26] M. H. Protter, H. F. Weinberger, *Maximum Principles in Differential Equations*, Englewood Cliffs, N.J., Prentice-Hall, (1967).
- [27] B. M. Regner, D. Vučinić, C. Domnisoru, T. M. Bartol, M. W. Hetzer, D. M. Tartakovsky, T. J. Sejnowski, (2013). *Anomalous Diffusion of Single Particles in Cytoplasm*. Biophysical Journal 104 (8): 1652-1660.
- [28] M. Shimrat, *Algorithm 112: Position of point relative to polygon*, Communications of the ACM Volume 5 Issue 8, (1962).
- [29] A. Zilman, S.D. Talia, B.T. Chait, M.P. Rout, M.O. Magnasco: Efficiency, selectivity, and robustness of nucleocytoplasmic transport. *PLoS Comp. Biol.* 3(7), e125 (2007)