UiO : **Department of Mathematics**
University of Oslo

An Investigation of Necessary Grid Resolution for Numerical Simulations of Calcium Dynamics in Cardiac Cells

Karoline Horgmo Jæger

Master's Thesis, Spring 2015

UNIVERSITAS OSLOENSIS
MDCCCXI

Abstract

One of the challenges related to modelling calcium dynamics in cardiac cells is the large difference in the length scales involved. The dyad, where important processes take place, is very small compared to the whole cell. Therefore, resolutions fine enough to capture the details of what happens in the dyad result in huge computational problems for whole-cell simulations, and the exact choice of resolution has a substantial effect on the problem size.

In this thesis, we investigate what grid resolution is necessary to capture the details of what happens in the dyad. We study simple mathematical models of calcium dynamics in the dyad and find analytical solutions to some of these simple models. Numerical simulations of the models are carried out for different resolutions using finite difference methods in 1D and 2D and a finite volume method in 3D. The accuracy of the numerical simulations is then studied by comparing the numerical solutions to analytical solutions and fine-grid numerical solutions, and the results suggest necessary resolutions in the nanometre range.

Acknowledgements

First and foremost, I would like to thank my supervisors, Aslak Tveito and Glenn T. Lines. I am deeply grateful for all the help, advise and guidance they have given me throughout this process and for how they so generously have read through the thesis and given me helpful comments during the last months. I would also like to thank Johan Hake and Glenn for their help with the code used for the 3D simulations.

Further thanks go to my fellow master student Karoline, whose friendship, help and support has meant so much to me during the last five years. I would also like to thank my little sister, Pernille, for her engaging and encouraging physiology explanations. Finally, I wish to thank Pernille and my parents for their invaluable love and support and for always being there for me, listening to me and encouraging me.

Contents

List of abbreviations

1D one-dimensional

2D two-dimensional

3D three-dimensional

jSR junctional sarcoplasmic reticulum

LCC L-type calcium channel

nSR network sarcoplasmic reticulum

RyR ryanodine receptor

SERCA sarco/endoplasmic reticulum Ca^{2+}-ATPase

SR sarcoplasmic reticulum

List of figures

List of tables

Chapter 1

Introduction

1.1 The cardiac muscle cell

The heart is the pump of the circulatory system, responsible for pumping blood through the vessels and thereby enabling transportation of substances around the body.

The main mass of the heart is made up by the cardiac muscle cells, and their contraction and relaxation is what makes the heart pump blood. The cells have a tubular shape, each with a length of about 100 μm and a diameter of about 10 μm. Figure 1.1 illustrates their shape and structure.

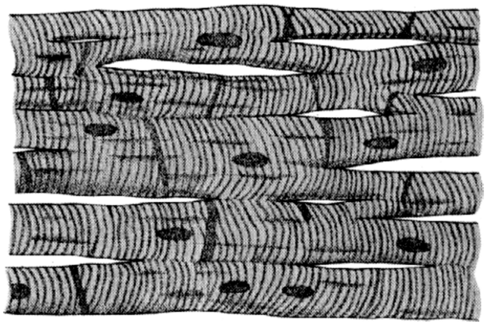

Figure 1.1: Structure of the cardiac muscle cells. The thick vertical lines are boundaries between cells, the thin vertical lines are z-lines and the dark dots are the cell nuclei. The illustration is taken from [4].

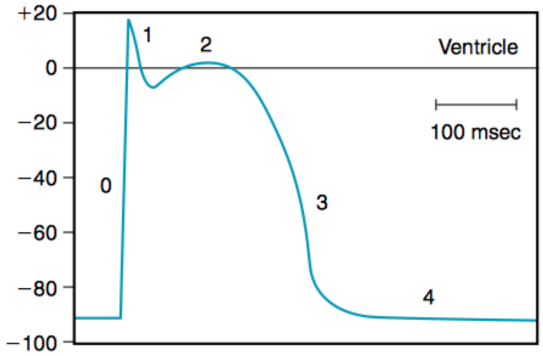

Figure 1.2: Typical action potential (in mV) of a ventricular cardiac muscle cell taken from [16].

The cardiac muscle cells contain a large number of contractile protein fibres (myofilaments), shortening the cells longitudinally during contraction. The myofilaments consist of a repeating sequence of contractile units called sarcomeres. A sarcomere is defined as the region between two so-called z-lines and is approximately 2 μm long.

At the z-lines, the cell is penetrated by so-called t-tubules. These form a network of tubular extensions of the cell membrane into the cell, are open to the extracellular space and contain extracellular fluid. They therefore facilitate effective communication between the intracellular and extracellular space.

The synchronous contraction and relaxation of the cardiac muscle cells are enabled by an electrical signal spreading through the cells, the so-called action potential. In the action potential, the potential across the cell membrane increases from a negative resting potential (depolarisation) and returns to the resting potential again (repolarisation). A typical action potential is illustrated in Figure 1.2. This electrical activity leads to contraction in a process called excitation-contraction coupling. In this process, calcium (Ca^{2+}) plays an important role.

This information about cardiac muscle cells is gathered from the references [9, 18, 19].

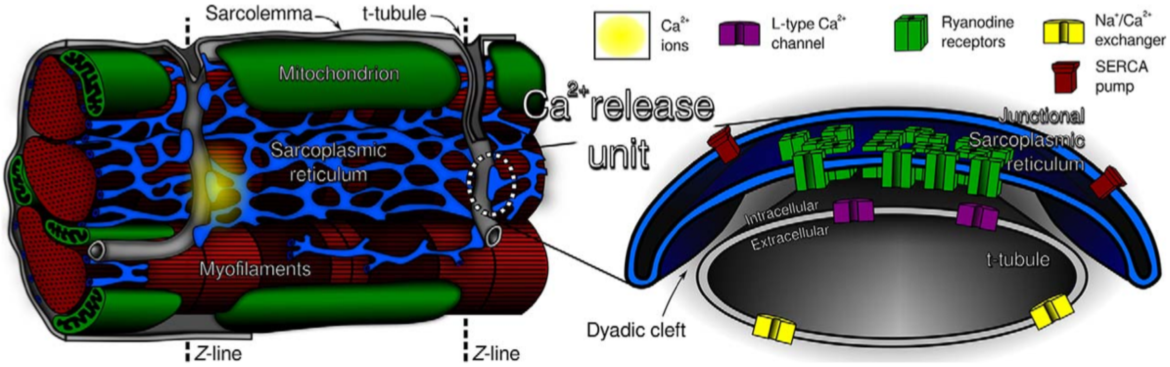

Figure 1.3: Left panel: Illustration of a sarcomere (i.e. the region between two z-lines, see Figure 1.1). The cell membrane (sarcolemma) forms t-tubules at the z-lines. The SR, coloured blue, forms a network throughout the cell and is close to the t-tubule at certain locations. **Right panel:** Illustration of a calcium release unit consisting of the t-tubule, dyadic cleft and jSR. There are RyRs located on the membrane between the jSR and the dyadic cleft and LCCs on the t-tubule. There are also SERCA-pumps on the membrane of the SR and Na$^+$-Ca^{2+} exchangers on the t-tubule. The illustration is taken from [3].

1.2 Calcium dynamics

An increased intracellular (cytosolic) calcium concentration is the signal that initiates contraction in the excitation-contraction coupling in a cardiac muscle cell.

In a normal heartbeat, the depolarisation of the cell membrane activates voltage-gated L-type calcium channels (LCCs) located on the t-tubules, allowing calcium to enter the cell from the extracellular space. The calcium entering the cell through the LCCs does not alone raise the calcium concentration in the cytosol enough to cause contraction. Instead, it activates ryanodine receptors (RyRs) on the membrane of the sarcoplasmic reticulum (SR) and a larger amount of calcium is released from the SR. This process is called calcium-induced calcium release.

The SR is a compartment within the cytosol which serves mainly as a calcium store. The part of the SR that is close to the t-tubules, where the LCCs are located, is called junctional SR (jSR), and the remaining part is called network SR (nSR).

The small gap between the LCCs and the jSR is called the dyadic cleft or

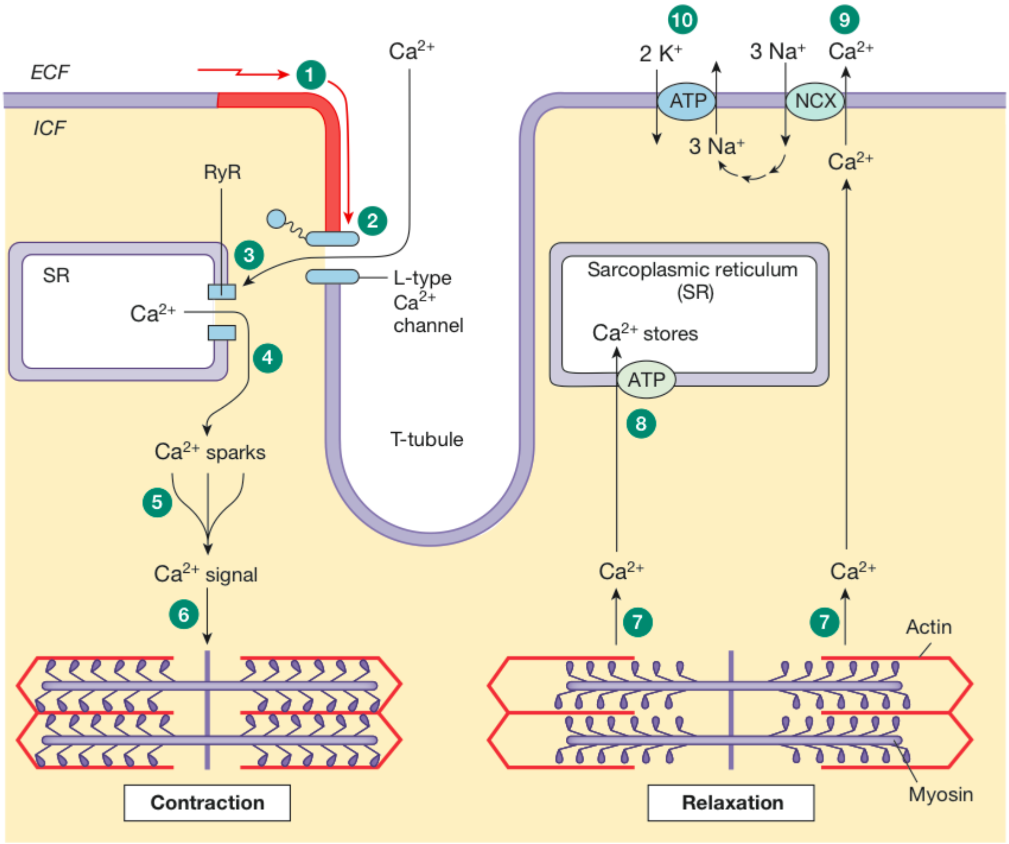

Figure 1.4: Illustration of calcium dynamics in a cardiac muscle cell taken from [20]. Depolarisation of the cell membrane (1) leads to calcium entering the cell through the voltage-gated LCCs (2). The increased calcium concentration activates the RyRs (3) and a larger amount of calcium is released from the SR causing a local increased calcium concentration known as a calcium spark (4). The sum of calcium sparks from several local releases makes up a calcium signal (5) which initiates contraction of the cell (6). The cell relax (7) as calcium is pumped back to the SR by the SERCA-pump (8) and extracted to the extracellular space by the Na$^+$-Ca^{2+} exchanger (9). The Na$^+$-Ca^{2+} exchanger is driven by Na$^+$ gradients maintained by the Na$^+$-K$^+$ pump (10).

the dyad. The gap is approximately 15 nm long and spans an area with a diameter of about 100–500 nm. This is where calcium first enters through the LCCs, diffuses to reach the RyRs and additional calcium is released from the SR. In a typical cardiac muscle cell, there are around 10,000–20,000 dyads.

After calcium is released from the SR, it diffuses out of the dyadic cleft to the rest of the cytosol. It then binds to the myofilaments and initiates contraction.

Both in the SR and in the cytosol, calcium is also bound to large calcium-binding proteins, called calcium buffers. This limits the amount of free calcium and slows down diffusion.

After contraction, the cell returns to its resting state by extracting calcium back to the SR and the extracellular space through, e.g., the SERCA pump and the Na^{+}-Ca^{2+} exchanger (see Figure 1.3).

Calcium's role in the excitation-contraction coupling in cardiac muscle cells is summarized in Figure 1.4, and the information presented in this section is gathered from the references [6, 10, 14, 19].

1.3 The problem

One of the challenges related to modelling calcium dynamics in cardiac muscle cells is the large difference in the length scales involved. The dyad, where calcium-induced calcium release takes place, is approximately 15 nm long and 100–500 nm wide [19]. These lengths are very small compared to the size of a whole cell with a diameter of about 10 μm and a length of about 100 μm [19]. In order to do numerical simulations that capture the details of what happens in the dyad, a fine grid resolution is required. Using this resolution in whole-cell simulations results in huge computational problems.

In the paper by Chai et al. [3], a simulation of calcium dynamics in a sarcomere of size 10 μm × 10 μm × 2 μm was run using a resolution of 3 nm in each spatial direction. An explicit numerical scheme was used, and the total number of floating-point operations for a 1 ms simulation was estimated to be approximately 1.9×10^{17}.

The simulation was carried out on the supercomputer Tianhe-2. Using 10,000 nodes on this computer, where each node was estimated to have a realistic performance of about 150 Gflop/s, a 1 ms simulation would take approximately 2 minutes.

This simulation time would be heavily increased if the resolution was changed from 3 nm to 1 nm. In that case, the total number of floating-point operations would be increased by a factor of $3^5 = 243$, and a 1 ms simulation would take more than 8 hours.

Simulations over a longer time period or over a larger domain would require even more simulation time. For instance, a 100 ms simulation of a whole cell of size 10 μm × 10 μm × 100 μm would require as simulation time of almost five years for a 1 nm resolution. Using a 3 nm resolution, this simulation would require a simulation time of about a week. The paper concludes that a resolution of 3 nm is within reach for whole-cell simulations of subcellular calcium dynamics, but that a 1 nm resolution is still out of reach.

From these estimations from [3] it is evident that the exact choice of grid resolution has a substantial effect on the size of the computational problem to be solved in the numerical simulations. It could therefore be useful to know what grid resolution is actually needed to capture the details of what happens in the dyad.

In this thesis, we address this question by studying some simple models of calcium dynamics in the dyad and investigating the accuracy of numerical simulations of these models for different grid resolutions. After finding an estimated necessary resolution in the dyad, we will carry out the calculations mentioned above to estimate how big the computational problem for a whole-cell simulation would be using that resolution.

1.4 Outline of the thesis

The thesis is structured as follows:

In Chapter 2, the mathematical models of calcium dynamics studied in the thesis are presented.

In Chapter 3, analytical solutions are derived for the equations of some of these models.

In Chapter 4, the numerical methods used to solve the equations of the models numerically are described.

In Chapter 5, results from the numerical simulations are presented, and the accuracy of the numerical simulations is studied by comparing the analytical solutions to the numerical solutions for different grid resolutions.

In Chapter 6, the results from the numerical simulations are summarized and consequenses for the rates in a Markov model of the RyR channel and consequences for whole-cell simulations are discussed.

In Chapter 7, conclusions and some suggestions for further work are presented.

Chapter 2

Mathematical models

In this chapter, we introduce the mathematical models of calcium dynamics studied in this thesis. In Section 2.1, a simple model of the dyad is introduced and in Section 2.2, the model is extended to include reactions with buffers. In Section 2.3, an outline of a whole-cell model is presented.

2.1 The dyad

In our simple models of calcium dynamics in the dyad, we consider a jSR domain in addition to a dyad domain and model how the calcium concentration, denoted by s and c, vary in time and space in these domains. We assume that there is an RyR channel on the boundary connecting the two domains. In addition, the jSR and dyad are connected to the nSR and cytosol, respectively, where the calcium concentration is assumed to be constant. The geometry of the model is shown in Figure 2.1.

Below, we will first introduce the model equations and boundary conditions in a general form, followed by the formulation of the 1D, 2D and 3D problems studied in this thesis.

nSR, $[\mathrm{Ca}^{2+}] = s_0$,	jSR, $s(t, \mathbf{x})$	dyad, $c(t, \mathbf{x})$	cytosol, $[\mathrm{Ca}^{2+}] = c_0$

Figure 2.1: Illustration of the model. The calcium concentration is high in the nSR and jSR and low in the dyad and the cytosol.

2.1.1 Diffusion

Both in the jSR and in the dyad, calcium diffuses from regions with high calcium concentration to regions with low calcium concentration. To derive equations modelling this process, we first consider some arbitrarily chosen region Ω with boundary $\partial\Omega$ in the dyad domain. In this region, we enforce conservation of mass, written as:

$$\frac{d}{dt}\int_{\Omega} c \, dV = \int_{\Omega} R \, dV - \int_{\partial\Omega} \mathbf{J}\cdot\mathbf{n} \, dA. \tag{2.1}$$

Here R is the rate of change of calcium concentration due to some local production, e.g. reactions with buffers, $\mathbf{J}$ is the flux of calcium and $\mathbf{n}$ is the outward unit normal vector.

Assuming that $\mathbf{J}$ is sufficiently smooth, the surface integral can be transformed to a volume integral by applying the divergence theorem:

$$\int_{\partial\Omega} \mathbf{J}\cdot\mathbf{n} \, dA = \int_{\Omega} \nabla\cdot\mathbf{J} \, dV.$$

Assuming that c is sufficiently smooth, integration and differentiation can be interchanged and (2.1) yields

$$\int_{\Omega}\left(\frac{\partial c}{\partial t} - R + \nabla\cdot\mathbf{J}\right) dV = 0.$$

Since Ω is an arbitrarily chosen region, the integrand must be zero, and we have:

$$\frac{\partial c}{\partial t} = R - \nabla\cdot\mathbf{J}. \tag{2.2}$$

Fick's law,

$$\mathbf{J} = -k_c\nabla c, \tag{2.3}$$

states that the flux is proportional to and directed in the opposite direction as the concentration gradient and is a reasonable assumption about the flux into or out of Ω due to diffusion. The parameter k_c is called the diffusion coefficient and is specific for the substance, in this case Ca^{2+}, and the fluid in which it is dissolved, in this case the cytosol in the dyadic cleft.

Inserting (2.3) in (2.2), assuming that k_c is constant and $R = 0$, we end up with the diffusion equation

$$\frac{\partial c}{\partial t} = k_c\nabla^2 c \tag{2.4}$$

for the calcium concentration in the dyad. Similarly, the equation

$$\frac{\partial s}{\partial t} = k_s \nabla^2 s \tag{2.5}$$

models the diffusion in the jSR. This derivation of the diffusion equation is based on the one given in [10].

2.1.2 Boundary conditions

In our models, we assume that there is an RyR channel on the boundary between the jSR and the dyad releasing calcium from the jSR, where the calcium concentration is high, to the dyad, where the calcium concentration is low. The flux through the channel is assumed to be proportional to the concentration difference between the jSR and the dyad outside the channel.

Since the flux is directed from the jSR to the dyad, the outward normal vector on the boundary for the jSR domain is directed in the same direction as the flux, and we can write $\mathbf{J} \cdot \mathbf{n} = \bar{g}(s - c)$ for some positive $\bar{g}$ on the boundary if the channel is open. If the channel is closed, there is no flux, and this can be combined in the expression

$$\mathbf{J} \cdot \mathbf{n} = g(s - c), \tag{2.6}$$

where

$$g(t) = \begin{cases} \bar{g} & \text{if the channel is open at time } t; \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

Combining this with Fick's law, $\mathbf{J} = -k_s \nabla s$, we get the boundary condition

$$-k_s \nabla s \cdot \mathbf{n} = g(s - c),$$

which can be rewritten as

$$k_s \frac{\partial s}{\partial n} = g(c - s). \tag{2.8}$$

In the dyad domain, the outward normal vector on the boundary to the jSR is directed in the opposite direction as the flux, so $\mathbf{J} \cdot \mathbf{n} = -g(s - c)$, and the corresponding boundary condition is

$$k_c \frac{\partial c}{\partial n} = g(s - c). \tag{2.9}$$

In addition, the jSR is connected to the nSR and the dyad is connected to the cytosol. Since the nSR and cytosol represent large compartments relatively far from the RyR channel, we assume that the calcium concentration is constant in these regions. This is incorporated in the models by applying Dirichlet boundary conditions on the respective boundaries.

In 2D and 3D, we let the remaining boundaries, e.g. the upper and lower boundaries in Figure 2.1, simply represent the end of the domain. We assume that no calcium enters or leaves the system through these boundaries and apply no-flux boundary conditions.

2.1.3 Formulation of a 1D problem

Bringing the equations and boundary conditions together and defining the domains

$$\begin{aligned} \Omega_s &= [-L_s, 0], \\ \Omega_c &= [0, L_c] \end{aligned}$$

for the jSR and dyad, respectively, our 1D model is formulated as the equations

$$\frac{\partial s}{\partial t} = k_s \frac{\partial^2 s}{\partial x^2}, \quad x \in \Omega_s, \tag{2.10}$$

$$\frac{\partial c}{\partial t} = k_c \frac{\partial^2 c}{\partial x^2}, \quad x \in \Omega_c \tag{2.11}$$

and the boundary conditions

$$s(t, -L_s) = s_0, \tag{2.12}$$

$$c(t, L_c) = c_0, \tag{2.13}$$

$$k_s \frac{\partial s}{\partial x}(t, 0) = g(t)(c(t,0) - s(t,0)), \tag{2.14}$$

$$k_c \frac{\partial c}{\partial x}(t, 0) = g(t)(c(t,0) - s(t,0)). \tag{2.15}$$

The function $g(t)$ is given by (2.7), and $\bar{g}$ will originally be the constant specified in Table 2.1. However, in order to derive analytical solutions in Section 3.1.1, the constant $\bar{g}$ will be replaced by an explicit function of time.

Similarly, we originally assume that the initial conditions are the constants

$$s(0, x) = s_0, \qquad c(0, x) = c_0, \tag{2.16}$$

but to find analytical solutions in Section 3.1.1, these will be adjusted to be functions of x, and the expressions are found on page 24.

Parameter	Value	Description
L_s	90 nm	Length of the jSR domain in x-direction
L_c	30 nm	Length of the dyad domain in x-direction
L_y	30 nm	Length of jSR and dyad domains in y-direction
s_0	1300 μM	Ca^{2+} concentration in the cytosol domain and initial condition in the dyad
c_0	0.14 μM	Ca^{2+} concentration in the nSR domain and initial condition in the jSR
$\bar{g}$	131 nm/ms	Flux rate through the RyR channel
k_s	$73.3 \cdot 10^3$ nm^2/ms	Diffusion coefficient for Ca^{2+} in the jSR
k_c	$220 \cdot 10^3$ nm^2/ms	Diffusion coefficient for Ca^{2+} in the dyad

Table 2.1: Parameters of the mathematical models of calcium dynamics. The chosen lengths L_s, L_c and L_y are based on the size of a dyad (see e.g. [2]). The rest of the parameters are taken from the paper [5]. Note that other values of the lengths L_s, L_c and L_y are used in the 3D problem (see Table 2.2). Note also that we adjust $\bar{g}$ and the initial conditions to fit analytical solutions for the simulations of the 1D and 2D problems.

2.1.4 Formulation of a 2D problem

The extension from 1D to 2D is done in a straightforward manner, except that we introduce some special Dirichlet boundary conditions on the form (2.19) and (2.20). This is done to introduce variation in the y-direction, enabling us to study how well the numerical methods perform under such variations, in a way so that we are still able to derive analytical solutions.

We extend the jSR and dyad domains to the rectangles

$$\begin{aligned}\Omega_s &= [-L_s, 0] \times [0, L_y],\\ \Omega_c &= [0, L_s] \times [0, L_y].\end{aligned}$$

The 2D versions on the diffusion equations are

$$\frac{\partial s}{\partial t} = k_s \left(\frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} \right), \qquad \mathbf{x} \in \Omega_s, \tag{2.17}$$

$$\frac{\partial c}{\partial t} = k_c \left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} \right), \qquad \mathbf{x} \in \Omega_c, \tag{2.18}$$

and the new Dirichlet boundary conditions are

$$s(t, -L_s, y) = h_s(y) = s_0 + A\cos(By), \qquad y \in [0, L_y], \tag{2.19}$$

$$c(t, L_c, y) = h_c(y) = c_0 + C\cos(Dy), \qquad y \in [0, L_y]. \tag{2.20}$$

For $x = 0$, the boundary conditions are on the familiar form

$$k_s \frac{\partial s}{\partial x}(t, 0, y) = g(t, y)(c(t, 0, y) - s(t, 0, y)), \qquad y \in [0, L_y], \tag{2.21}$$

$$k_c \frac{\partial c}{\partial x}(t, 0, y) = g(t, y)(c(t, 0, y) - s(t, 0, y)), \qquad y \in [0, L_y], \tag{2.22}$$

and we apply no-flux boundary conditions for the boundaries in the y-direction:

$$\frac{\partial s}{\partial y}(t, x, 0) = 0, \qquad \frac{\partial s}{\partial y}(t, x, L_y) = 0, \qquad x \in [-L_s, 0], \tag{2.23}$$

$$\frac{\partial c}{\partial y}(t, x, 0) = 0, \qquad \frac{\partial c}{\partial y}(t, x, L_y) = 0, \qquad x \in [0, L_c]. \tag{2.24}$$

The inital conditions and $\bar{g}$ are adjusted to fit analytical solutions and are specified on page 37.

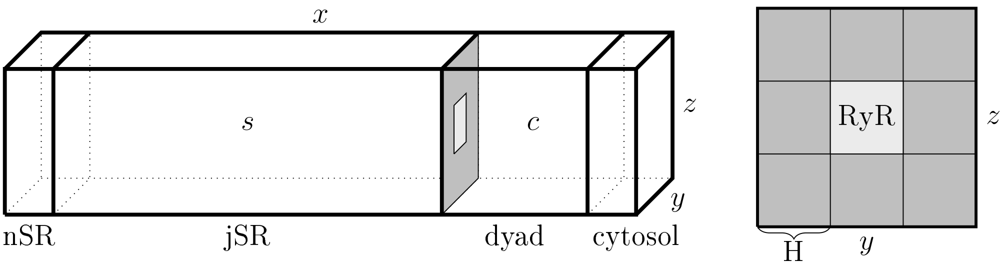

Figure 2.2: Left panel: Geometry of the 3D model. **Right panel:** The boundary between the jSR and the dyad. The lighter area represents the RyR channel.

2.1.5 Formulation of a 3D problem

In the 2D problem, we assumed that there was flux through the RyR channel on the entire boundary between the jSR and the dyad. In the 3D problem, we restrict the channel to a quadratic area on the boundary (see Figure 2.2).

Otherwise, the extention to 3D is straightforward. The jSR and dyad domains are defined as the rectangular boxes

$$\Omega_s = [-L_s, 0] \times [0, L_y] \times [0, L_z],$$

$$\Omega_c = [0, L_c] \times [0, L_y] \times [0, L_z],$$

and the 3D versions of the diffusion equations (2.4) and (2.5) models the calcium concentration in these domains.

Compartments representing the nSR and the cytosol are located to the left of the jSR and to the right of the dyad, respectively, and the concentration in the nSR and the cytosol is assumed to be constant.

On the boundary between the jSR and the dyad, the boundary conditions are given by (2.8) and (2.9) on the RyR-part of the boundary. On the remaining part of the boundary between the jSR and the dyad and on the boundaries in y- and z-direction, we apply no-flux boundary conditions.

Since we do not find analytical solutions in this case, we let $\bar{g}$ be the constant in Table 2.1 and the initial conditions be the constants s_0 and c_0 for the jSR and dyad, respectively.

Parameter	Value	Description
L_s	96 nm	Length of the jSR domain in x-direction
L_c	36 nm	Length of the dyad domain in x-direction
L_y	18 nm	Length of jSR and dyad domains in y-direction
L_z	18 nm	Length of jSR and dyad domains in z-direction
H	6 nm	Length of the sides of the quadratic RyR area (see Figure 2.2)

Table 2.2: Length parameters specific to the 3D model. The lengths L_s, L_c, L_y and L_z are based on the size of a dyad (see e.g. [2]). The value of H, the length of the sides of the RyR channel, is chosen to fit easily with the implementation of the numerical method used in the 3D simulations and so are the values of the other length parameters.

2.2 Buffers

In the model introduced in Section 2.1, the interaction between calcium and calcium buffers is ignored. In this section, we extend the model to include such interactions.

We first consider how to model calcium buffering in general and then formulate a specific model, adding two buffers to the 1D problem in Section 2.1.3. The general derivation of equations is based on the one given in [10] and the specific model is based on the model in [5].

2.2.1 The general case

To derive equations modelling the interaction between calcium and buffers, we apply the law of mass action, which states that given a reaction on the form

$$A + B \quad \xrightarrow{k} \quad C,$$

where k is the rate constant, the rate of the reaction is given by

$$\frac{d[C]}{dt} = k[A][B],$$

where $[X]$ denotes the concentration of the substance X.

The chemical reaction between calcium and a buffer can be represented by the reaction

$$\mathrm{P} + \mathrm{Ca}^{2+} \underset{k_{\text{off}}}{\overset{k_{\text{on}}}{\rightleftarrows}} \mathrm{B}, \tag{2.25}$$

where P represents the buffering protein and B represents calcium bound to the buffer.

Letting B_{tot} denote the total buffer concentration, c the concentration of free calcium and b the concentration of calcium bound to the buffer, the law of mass action states that the rate of change in calcium and buffer concentration due to calcium-buffer reactions is given by

$$\begin{aligned} \frac{db}{dt} &= R(c,b), \\ \frac{dc}{dt} &= -R(c,b), \end{aligned}$$

where

$$R(c,b) = k_{\text{on}}c(B_{\text{tot}} - b) - k_{\text{off}}b.$$

To get a model for the total change in concentration, the $R(c,b)$-term is added to the diffusion equations, and we get the equations

$$\begin{aligned} \frac{\partial c}{\partial t} &= k_c \nabla^2 c - R(c,b), \\ \frac{\partial b}{\partial t} &= k_b \nabla^2 b + R(c,b), \end{aligned}$$

if we assume that the buffer is free to diffuse and that k_c and k_b are the diffusion coefficients for free calcium and calcium bound to buffer, respectively.

Initial conditions for the buffers

To find appropriate initial conditions for the concentration of calcium bound to buffer, we assume that the buffer reaction (2.25) is initially in equilibrium, i.e. that the reaction rates for the forward and backward reactions are the same. This means that

$$k_{\text{on}}c(B_{\text{tot}} - b) = k_{\text{off}}b.$$

Letting c_0 denote the initial concentration of free calcium, the initial concentration of calcium bound to the buffer is then given by:

$$b_0 = \frac{k_{\text{on}}c_0B_{\text{tot}}}{k_{\text{on}}c_0 + k_{\text{off}}}. \tag{2.26}$$

Parameter	Value	Description
k_b	$25 \cdot 10^3$ nm^2/ms	Diffusion coefficient for CMDN
B^c_{tot}	24 μM	Total CMDN concentration in dyad
k^c_{on}	$34 \cdot 10^{-3}$ ms$^{-1}\mu$M^{-1}	On rate for CMDN Ca^{2+} buffering
k^c_{off}	$238 \cdot 10^{-3}$ ms^{-1}	Off rate for CMDN Ca^{2+} buffering
b^c_0	0.471 μM	Initial concentration of Ca^{2+} bound to CMDN in the dyad
B^s_{tot}	$30 \cdot 10^3$ μM	Total CSQN concentration in jSR
k^s_{on}	$102 \cdot 10^{-3}$ ms$^{-1}\mu$M^{-1}	On rate for CSQN Ca^{2+} buffering
k^s_{off}	65 ms^{-1}	Off rate for CSQN Ca^{2+} buffering
b^s_0	$20.1 \cdot 10^3$ μM	Initial concentration of Ca^{2+} bound to CSQN in the jSR

Table 2.3: Parameters for the buffers taken from [5].

2.2.2 Formulation of a 1D problem with buffers

In this section, we will extend the 1D problem in Section 2.1.3 by including two buffers in the model. A buffer called calsequestrin (CSQN) is introduced in the jSR and a buffer called calmodulin (CMDN) is introduced in the dyad. CSQN is the main calcium-binding buffer in the SR [1]. It is a stationary protein located in the jSR, where it among other functions enables storage of large amounts of calcium [1]. CMDN is a mobile calcium-binding protein free to diffuse throughout the cytosol including in the dyad [22].

Letting b_s denote the concentration of calcium bound to CSQN and b_c the concentration of calcium bound to CMDN, we get the following system of equations:

$$\frac{\partial s}{\partial t} = k_s \frac{\partial^2 s}{\partial x^2} - R_s(s, b_s), \qquad x \in [-L_s, 0], \tag{2.27}$$

$$\frac{\partial b_s}{\partial t} = R_s(s, b_s), \qquad x \in [-L_s, 0], \tag{2.28}$$

$$\frac{\partial c}{\partial t} = k_c \frac{\partial^2 c}{\partial x^2} - R_c(c, b_c), \qquad x \in [0, L_c], \tag{2.29}$$

$$\frac{\partial b_c}{\partial t} = k_b \frac{\partial^2 b_c}{\partial x^2} + R_c(c, b_c), \qquad x \in [0, L_c], \tag{2.30}$$

where

$$R_s(s, b_s) = k_{\text{on}}^s s \left(B_{\text{tot}}^s - b_s\right) - k_{\text{off}}^s b_s, \tag{2.31}$$

$$R_c(c, b_c) = k_{\text{on}}^c c \left(B_{\text{tot}}^c - b_c\right) - k_{\text{off}}^c b_c. \tag{2.32}$$

The boundary conditions (2.12) − (2.15) are applied for s and c, where $\bar{g}$ is the constant in Table 2.1. For CMDN we apply the no-flux boundary conditions:

$$\frac{\partial b_c}{\partial x}(t, 0) = 0, \qquad \frac{\partial b_c}{\partial x}(t, L_c) = 0. \tag{2.33}$$

The initial conditions are the constants s_0 and c_0 for s and c, respectively, and the initial conditions for the buffers are calculated from (2.26).

2.3 The cell

In a typical cardiac muscle cell there are around $10,000 - 20,000$ dyads [14]. To model calcium dynamics in the whole cell, the model of one dyad would therefore have to be expanded to include several dyads.

Since calcium is free to diffuse out of the dyadic cleft into the bulk cytosol, these dyads should be connected by a large cytosol compartment. Similarly, the different jSR compartments should be connected by an nSR.

Expanding the model to include several dyads in addition to cytosol and nSR compartments connecting them would obviously imply a much larger domain. In the 3D model of the dyad and jSR presented in Section 2.1.5, the domain is 132 nm × 18 nm × 18 nm. A whole cell of size 100 μm × 10 μm × 10 μm [19], would then have a roughly 10^8 times larger volume.

In this thesis, we only do simulations of calcium dynamics in one dyad. Since the spatial domain of a dyad is much smaller than that of the whole cell, the size of the computational problem to be solved in the simulations is much more manageable. We can therefore easily do simulations using different grid resolutions to investigate what resolution is necessary to get accurate simulations of the dyad. By extension, this resolution would also be necessary for whole-cell simulations because we expect that in order to get accurate simulations of a large system of dyads, we need to use a resolution fine enough to give accurate simulations of one dyad.

Chapter 3

Analytical solutions

The systems of equations involved in models of calcium dynamics are generally too complicated to be solved analytically. Instead, numerical methods are used to find approximations of the solutions, and the accuracy of these numerical solutions will usually depend on the grid resolution used in the simulations. To investigate what resolution is necessary to get accurate simulations of calcium dynamics in the dyad, we will in this thesis use the numerical methods described in Chapter 4 to find numerical solutions of the models from Chapter 2 for different grid resolutions.

In our attempt to find the necessary resolution, it would be very convenient to also have analytical solutions of the equations since that would allow us to determine exactly how close the numerical approximations are to the exact solutions. In this chapter, we therefore derive analytical solutions to the simple 1D and 2D models without buffers introduced above.

Note though that in the derivation of the analytical solutions, we adjust the flux from the jSR to the dyad to fit the analytical solutions. This flux would normally be given by the mathematical model of the RyR channel, e.g. as the boundary conditions (2.8) and (2.9), where $\bar{g}$ is a specified constant. In the solutions derived below, however, we need to adjust the parameter $\bar{g}$ to be a constructed function of t (and y for the 2D problem).

In Chapter 5, we will compare numerical solutions for different resolutions to the analytical solutions derived in this chapter.

3.1 Analytical solutions of the 1D problem

In this section, we derive analytical solutions of the 1D problem described in Section 2.1.3. The solutions of the system depend on whether the RyR channel is open or closed. We will consider the case where the channel is open for $t < t^*$ and then closed for $t \geq t^*$ for some t^*.

3.1.1 Open channel

We first assume that the channel is open. In that case, we find that solutions on the form

$$s(t,x) = a_1 e^{-a_2 t} \sin(a_3(x + L_s)) + a_4 x + a_5, \tag{3.1}$$

$$c(t,x) = b_1 e^{-b_2 t} \sin(b_3(x - L_c)) + b_4 x + b_5 \tag{3.2}$$

fulfil the equations and boundary conditions (2.10)−(2.15) of the 1D problem if the parameters, the initial conditions and $\bar{g}$ are adjusted in a certain way. This is explained in detail below.

Fulfilling the diffusion equations

First, the solutions should satisfy the diffusion equations

$$\frac{\partial s}{\partial t} = k_s \frac{\partial^2 s}{\partial x^2}, \qquad \frac{\partial c}{\partial t} = k_c \frac{\partial^2 c}{\partial x^2}.$$

Inserting s and c on the form (3.1) and (3.2) yields

$$-a_1 a_2 e^{-a_2 t} \sin(a_3(x + L_s)) = -k_s a_1 a_3^2 e^{-a_2 t} \sin(a_3(x + L_s)),$$

$$-b_1 b_2 e^{-b_2 t} \sin(b_3(x - L_c)) = -k_c b_1 b_3^2 e^{-b_2 t} \sin(b_3(x - L_c)),$$

and for this to be fulfilled for all t and x, we must have:

$$a_2 = k_s a_3^2, \tag{3.3}$$

$$b_2 = k_c b_3^2. \tag{3.4}$$

Fulfilling the Dirichlet boundary conditions

Next, we need the solutions to satisfy the Dirichlet boundary conditions

$$s(t, -L_s) = s_0, \qquad c(t, L_c) = c_0.$$

Inserting $x = -L_s$ in (3.1) and $x = L_c$ in (3.2), we find

$$\begin{aligned} s(t, -L_s) &= -L_s a_4 + a_5, \\ c(t, L_c) &= L_c b_4 + b_5, \end{aligned}$$

which means that we require

$$s_0 = -L_s a_4 + a_5, \tag{3.5}$$

$$c_0 = L_c b_4 + b_5. \tag{3.6}$$

Fulfilling the Neumann boundary conditions

The solutions should also fulfil the Neumann boundary conditions

$$\begin{aligned} k_s \frac{\partial s}{\partial x}(t, 0) &= \bar{g}(t)(c(t, 0) - s(t, 0)), \\ k_c \frac{\partial c}{\partial x}(t, 0) &= \bar{g}(t)(c(t, 0) - s(t, 0)). \end{aligned}$$

To find the necessary conditions for these to hold, we first note that the boundary conditions imply

$$k_s \frac{\partial s}{\partial x}(t, 0) = k_c \frac{\partial c}{\partial x}(t, 0).$$

Inserting s and c on the form (3.1) and (3.2) gives

$$k_s(a_1 a_3 e^{-a_2 t} \cos(a_3 L_s) + a_4) = k_c(b_1 b_3 e^{-b_2 t} \cos(-b_3 L_c) + b_4).$$

To make this true for all t and x, we must have:

$$k_s a_1 a_3 \cos(L_s a_3) = k_c b_1 b_3 \cos(L_c b_3), \tag{3.7}$$

$$k_s a_4 = k_c b_4, \tag{3.8}$$

$$a_2 = b_2. \tag{3.9}$$

When these conditions are fulfilled, the Neumann boundary conditions hold if we adjust $\bar{g}(t)$ to be

$$\bar{g}(t) = \frac{k_c \frac{\partial c}{\partial x}(t,0)}{c(t,0) - s(t,0)} = \frac{k_s \frac{\partial s}{\partial x}(t,0)}{c(t,0) - s(t,0)}$$

$$= \frac{k_s(a_1 a_3 e^{-a_2 t}\cos(a_3 L_s) + a_4)}{b_1 e^{-b_2 t}\sin(-b_3 L_c) + b_5 - a_1 e^{-a_2 t}\sin(a_3 L_s) - a_5}. \tag{3.10}$$

This means that in order to find analytical solutions of the 1D problem (2.10) – (2.15) on the form (3.1) and (3.2), we need to construct $\bar{g}$ in the flux through the open RyR channel to be the function (3.10).

Initial conditions

For the initial conditions to match the solutions s and c on the form (3.1) and (3.2), we define the initial conditions as

$$s(0,x) = s_{\text{init}}(x) = a_1 \sin(a_3(x + L_s)) + a_4 x + a_5, \tag{3.11}$$

$$c(0,x) = c_{\text{init}}(x) = b_1 \sin(b_3(x - L_c)) + b_4 x + b_5, \tag{3.12}$$

where we have simply inserted $t = 0$ in s and c on the form (3.1) and (3.2).

Finding the solutions

Bringing everything together, we end up with seven equations (3.3) – (3.9) that must be satisfied. This can be achieved by chosing some values for a_1, a_3 and a_4 and adjusting the rest of the parameters to satisfy the seven equations.

The system (3.3), (3.4) and (3.9) can be solved by

$$a_2 = k_s a_3^2, \qquad b_2 = a_2, \qquad b_3 = \sqrt{\frac{k_s a_3^2}{k_c}}. \tag{3.13}$$

From (3.7) we get

$$b_1 = \frac{k_s a_1 a_3 \cos(L_s a_3)}{k_c b_3 \cos(L_c b_3)}, \tag{3.14}$$

condition (3.8) yields

$$b_4 = \frac{k_s}{k_c} a_4, \tag{3.15}$$

and (3.5) and (3.6) give

$$a_5 = s_0 + L_s a_4, \qquad b_5 = c_0 - L_c b_4. \tag{3.16}$$

If these conditions are satisfied and $\bar{g}$ and the initial conditions are adjusted as explained above, (3.1) and (3.2) are solutions of the 1D problem (2.10) – (2.15).

Special case with constant $\bar{g}$

For the solutions on the form (3.1) and (3.2) to be solutions of the 1D problem, we needed to use a $\bar{g}(t)$ constructed to fit the analytical solutions. This is a drawback since, as metioned above, the value of $\bar{g}$ would normally be given by the mathematical model. A typical choice is that $\bar{g}$ is a specified constant like the one given in Table 2.1. It could therefore be interesting to consider the special case were the parameters in the analytical solutions are adjusted to make $\bar{g}(t)$ a specified constant.

One way to make the function $\bar{g}(t)$ constant is to choose $a_1 = 0$. In that case, (3.14) gives that b_1 is also zero. Inserting this in (3.10) shows that $\bar{g}(t)$ in this case is the constant function:

$$\bar{g}(t) = \frac{k_s a_4}{b_5 - a_5}.$$

By inserting $a_1 = b_1 = 0$ in (3.1) and (3.2), we end up with the linear solutions

$$s(t,x) = a_4 x + a_5,$$
$$c(t,x) = b_4 x + b_5.$$

Using the definition of a_5 and b_5 from (3.16), these solutions can be rewritten as

$$s(x) = a_4(x + L_s) + s_0, \tag{3.17}$$
$$c(x) = b_4(x - L_c) + c_0. \tag{3.18}$$

If we have some value of $\bar{g}$ specified by the model, we can now determine the values of a_4 and b_4 and find the resulting solutions. We want to find a_4 and b_4 so that

$$\frac{k_s a_4}{b_5 - a_5} = \bar{g}.$$

Inserting (3.15) – (3.16), we get

$$\frac{k_s a_4}{c_0 - \frac{k_s}{k_c} L_c a_4 - s_0 - L_s a_4} = \bar{g},$$

and solving for a_4 gives

$$a_4 = \frac{k_c \bar{g}(c_0 - s_0)}{k_s k_c + L_c k_s \bar{g} + L_s k_c \bar{g}}. \tag{3.19}$$

From (3.15) we now have that b_4 is given by:

$$b_4 = \frac{k_s}{k_c} a_4 = \frac{k_s \bar{g}(c_0 - s_0)}{k_s k_c + L_c k_s \bar{g} + L_s k_c \bar{g}}. \tag{3.20}$$

This means that in the case of an open channel with a constant $\bar{g}(t) = \bar{g}$, we get solutions on the form (3.17) and (3.18), where a_4 and b_4 are given by (3.19) and (3.20).

The jump $s(t, 0) - c(t, 0)$ is then given by

$$s(0) - c(0) = \frac{k_c k_s (s_0 - c_0)}{k_s k_c + L_c k_s \bar{g} + L_s k_c \bar{g}}.$$

In the way we have constructed the solutions and initial conditions above, the solutions (3.17) and (3.18) will in this case be specified as initial conditions and as time goes by, the solutions will stay the same. Since nothing happens, these solutions for a constant $\bar{g}(t)$ are not so suitable to test the accuracy of the numerical simulations.

The solutions could be of interest for other initial conditions, though, because solutions of a system like the 1D problem (2.10) – (2.15) for an open channel with a constant $\bar{g}$ could be expected to eventually reach steady-state solutions that do not change with time. Since (3.17) and (3.18) are solutions satisfying the equations and boundary conditions in this case, we expect these solutions to be the steady-state solutions of the system for other initial conditions as well. In that case, we do not know the solutions for small values of t, but expect the solutions to approach (3.17) and (3.18) as $t \to \infty$. This means that for an open channel with a constant $\bar{g}$ and some chosen initial conditions, e.g. the constants s_0 and c_0, our derived analytical solutions serve as expected steady-state solutions of the system. In Chapter 5, this will be investigated by numerical simulations.

Parameter	Value
t^*	2.0 ms
a_1	-310 μM
a_3	-0.0072 nm^{-1}
a_4	-1.97 μM/nm

Table 3.1: Choice of values for the free parameters used in the adjusted solutions. The remaining parameter values are given by (3.13) – (3.16) and in Table 2.1.

Choice of free parameters

In the solutions constructed above, we ended up with three free parameters a_1, a_3 and a_4. Although any choice of values for these parameters would satisfy the equations and boundary conditions, in our attempt to study the accuracy of numerical simulations of calcium dynamics, we wish to choose values so that the analytical solutions are close to the solutions of the original mathematical model.

In the simulations presented in Chapter 5, we assign a_4 the value specified by (3.19), where the parameters, including the constant $\bar{g}$, are the ones given in Table 2.1. Since the solutions $s(x,t)$ and $c(x,t)$ given by (3.1) and (3.2) approach

$$\begin{aligned} s(t,x) &= a_4 x + a_5, \\ c(t,x) &= b_4 x + b_5 \end{aligned}$$

as $t \to \infty$, this assures that the steady-state solutions are the same as they would have been if we had used the constant $\bar{g}$ instead of the constructed $\bar{g}(t)$.

The choice of values for a_1 and a_3 were done by doing simulations with different values, trying to find some appropriate a_1 and a_3 that gave somewhat reasonable solutions, e.g. by ensuring that $\bar{g}(t)$ was positive for all $t \geq 0$ and that the initial conditions were almost constant. The chosen values are given in Table 3.1.

The time to close the channel, t^*, was chosen as the time where the solutions in the simulations seemed to have reached steady state.

3.1.2 Closed channel

We will now derive solutions of the 1D problem when the channel is closed, i.e. when there are no-flux boundary conditions on the boundary between the jSR and the dyad.

To find the analytical solutions in this case, we use Fourier's method, which allows us to adjust the solutions to satisfy the initial conditions of our choice. We can therefore find the solutions for a closed channel, given that the channel has been open until $t = t^*$. Introductions to Fourier's method for solving partial differential equations are given in e.g. [15, 24].

Since there is no connection between the two domains in this case, the equations for s and c can be solved separately, and we will solve the equation for s, the calcium concentration in the jSR, first.

Homogeneous Dirichlet boundary condition

To use Fourier's method, we initially consider a homogeneous Dirichlet boundary condition. This means that we are looking for solutions of the problem:

$$\frac{\partial s}{\partial t} = k_s \frac{\partial^2 s}{\partial x^2}, \quad x \in [-L_s, 0], \tag{3.21}$$

$$s(t, -L_s) = 0, \tag{3.22}$$

$$\frac{\partial s}{\partial x}(t, 0) = 0. \tag{3.23}$$

Afterwards, the constant s_0 can be added to the solution to fulfil the boundary condition $s(t, -L_s) = s_0$. Since the derivative of a constant is zero, the diffusion equation and no-flux boundary condition will still be satisfied.

Separation of variables

As a first step to find a solution of (3.21) − (3.23), we assume that the solution can be separated into its t and x dependency in the sense that [24]:

$$s(t, x) = T(t)X(x).$$

If we put this into equation (3.21) and assume that neither $X(x)$ nor $T(t)$ is zero, we find

$$T'(t)X(x) = k_s T(t)X''(x) \quad \Rightarrow \quad \frac{T'(t)}{T(t)} = k_s \frac{X''(x)}{X(x)} = -\lambda,$$

for some constant λ.

This results in the two ordinary differential equations:

$$X''(x) + \frac{\lambda}{k_s} X(x) = 0, \tag{3.24}$$

$$T'(t) + \lambda T(t) = 0. \tag{3.25}$$

Boundary condition (3.22) is now

$$X(-L_s) = 0, \tag{3.26}$$

and boundary condition (3.23) is

$$X'(0) = 0. \tag{3.27}$$

Finding $X(x)$

The solution of equation (3.24) depends on the sign of λ. If we assume $\lambda \leq 0$, the trivial solution $X(x) = 0$ is the only possible solution. We therefore assume $\lambda > 0$. The solution of equation (3.24) is then on the form:

$$X(x) = C \cos\left(\sqrt{\tfrac{\lambda}{k_s}} x\right) + D \sin\left(\sqrt{\tfrac{\lambda}{k_s}} x\right).$$

Boundary condition (3.27) now implies $D = 0$, and then boundary condition (3.26) implies

$$C \cos\left(-\sqrt{\tfrac{\lambda}{k_s}} L_s\right) = 0 \quad \Rightarrow \quad \sqrt{\frac{\lambda}{k_s}} L_s = (2k-1)\frac{\pi}{2} \quad \Rightarrow \quad \frac{\lambda}{k_s} = \left(\frac{(2k-1)\pi}{2L_s}\right)^2$$

for any integer k.

We have now found a family of possible values of λ;

$$\lambda_k = \left(\frac{(2k-1)\pi}{2L_s}\right)^2 k_s$$

and corresponding functions;

$$X_k(x) = \cos\left(\frac{(2k-1)\pi}{2L_s} x\right).$$

Since we have no conditions for the constant C, we can choose any coefficient for the $X_k(x)$-functions.

Finding the particular solutions

Equation (3.25) has a solution on the form

$$T(t) = Ee^{-\lambda t},$$

so any

$$T_k(t) = e^{-\left(\frac{(2k-1)\pi}{2L_s}\right)^2 k_s t}$$

with any coefficient will satisfy the system (3.24) – (3.27).

Multiplying $T_k(t)$ and $X_k(x)$, we find the particular solutions

$$s_k(t,x) = T_k(t)X_k(x) = e^{-\left(\frac{(2k-1)\pi}{2L_s}\right)^2 k_s t} \cos\left(\frac{(2k-1)\pi}{2L_s}x\right),$$

all fulfilling (3.21) – (3.23). Since the system is linear and homogenous, we see that any linear combination of these particular solutions will satisfy the system. This is called the principle of superposition [13].

Adjusting the solution to an initial condition

The particular solutions can now be used to find a solution that fulfils the initial condition

$$s(t^*,x) = f_s(x).$$

To find this solution, we first assume that the function $f_s(x)$ can be expressed as a Fourier series on the form

$$s(t^*,x) = f_s(x) = s_0 + \sum_{m=1}^{\infty} \alpha_m \cos\left(\frac{(2m-1)\pi}{2L_s}x\right), \tag{3.28}$$

where α_m are some appropriate constants.

In that case, we see by insertion that the solution

$$s(t,x) = s_0 + \sum_{m=1}^{\infty} \alpha_m e^{-\left(\frac{(2m-1)\pi}{2L_s}\right)^2 k_s (t-t^*)} \cos\left(\frac{(2m-1)\pi}{2L_s}x\right) \tag{3.29}$$

satisfies both inital condition (3.28), equation (3.21), boundary condition (3.23) and the boundary condition $s(t,-L_s) = s_0$.

In order to find appropriate values for α_m, we first note that for $m,n \geq 1$:

$$\int_{-L_s}^{0} \cos\left(\frac{(2m-1)\pi}{2L_s}x\right) \cos\left(\frac{(2n-1)\pi}{2L_s}x\right) dx = \begin{cases} L_s/2 & \text{if } m = n, \\ 0 & \text{if } m \neq n. \end{cases}$$

We can now use (3.28) to calculate

$$\begin{aligned}\int_{-L_s}^{0} (f_s(x) - s_0) \cos\left(\tfrac{(2k-1)\pi}{2L_s}x\right) dx &= \sum_{m=1}^{\infty} \alpha_m \int_{-L_s}^{0} \cos\left(\tfrac{(2m-1)\pi}{2L_s}x\right) \cos\left(\tfrac{(2k-1)\pi}{2L_s}x\right) dx \\ &= \alpha_k \int_{-L_s}^{0} \cos\left(\tfrac{(2k-1)\pi}{2L_s}x\right) \cos\left(\tfrac{(2k-1)\pi}{2L_s}x\right) dx \\ &= \alpha_k \frac{L_s}{2}.\end{aligned}$$

This can be used as a formula for α_m:

$$\alpha_m = \frac{2}{L_s} \int_{-L_s}^{0} (f_s(x) - s_0) \cos\left(\tfrac{(2m-1)\pi}{2L_s}x\right) dx. \tag{3.30}$$

A solution for c

Using the same arguments, it can be shown that the c-solution is given by

$$c(t,x) = c_0 + \sum_{m=1}^{\infty} \beta_m e^{-\left(\frac{(2m-1)\pi}{2L_c}\right)^2 k_c (t-t^*)} \cos\left(\tfrac{(2m-1)\pi}{2L_c}x\right), \tag{3.31}$$

where

$$\beta_m = \frac{2}{L_c} \int_{0}^{L_c} (f_c(x) - c_0) \cos\left(\tfrac{(2m-1)\pi}{2L_c}x\right) dx \tag{3.32}$$

and $f_c(x)$ is the function defining the initial condition $c(t^*, x) = f_c(x)$.

Coefficients for our special case

We will now connect the two solutions s and c by assuming that the channel has been open until $t = t^*$.

Assuming that the solutions are on the form

$$\begin{aligned} s(t,x) &= a_1 e^{-a_2 t} \sin(a_3(x + L_s)) + a_4 x + a_5, \\ c(t,x) &= b_1 e^{-b_2 t} \sin(b_3(x - L_c)) + b_4 x + b_5 \end{aligned}$$

for an open channel, we can use the initial conditions

$$\begin{aligned} f_s(x) = s(t^*, x) &= a_1 e^{-a_2 t^*} \sin(a_3(x + L_s)) + a_4 x + a_5, \\ f_c(x) = c(t^*, x) &= b_1 e^{-b_2 t^*} \sin(b_3(x - L_c)) + b_4 x + b_5 \end{aligned}$$

to find solutions for after the channel is closed.

The coefficients are calculated using (3.30) and (3.32), and we find:

$$\begin{aligned}\alpha_m &= \frac{2}{L_s}\int_{-L_s}^{0} (f_s(x) - s_0)\cos\left(\tfrac{(2m-1)\pi}{2L_s}x\right)dx \\ &= -\frac{8L_s a_1 a_3 e^{-a_2 t^*}\cos(a_3 L_s)}{4L_s^2 a_3^2 - (2m-1)^2\pi^2} + \frac{8L_s a_4}{(2m-1)^2\pi^2},\end{aligned} \tag{3.33}$$

$$\begin{aligned}\beta_m &= \frac{2}{L_c}\int_{0}^{L_c} (f_c(x) - c_0)\cos\left(\tfrac{(2m-1)\pi}{2L_c}x\right)dx \\ &= \frac{8L_c b_1 b_3 e^{-b_2 t^*}\cos(b_3 L_c)}{4L_c^2 b_3^2 - (2m-1)^2\pi^2} - \frac{8L_c b_4}{(2m-1)^2\pi^2}.\end{aligned} \tag{3.34}$$

This means that if the channel has been open until t^* and the solutions are on the form (3.1) and (3.2) for an open channel, then the solutions are on the form (3.29) and (3.31) where α_m and β_m are given by (3.33) and (3.34) after the channel is closed.

Partial sum approximations

The analytical solutions (3.29) and (3.31) are represented as infinite sums. In Table 3.2, we observe how the partial sums

$$s_M(t,x) = s_0 + \sum_{m=1}^{M} \alpha_m e^{-\left(\frac{(2m-1)\pi}{2L_s}\right)^2 k_s(t-t^*)}\cos\left(\tfrac{(2m-1)\pi}{2L_s}x\right), \tag{3.35}$$

$$c_M(t,x) = c_0 + \sum_{m=1}^{M} \beta_m e^{-\left(\frac{(2m-1)\pi}{2L_c}\right)^2 k_c(t-t^*)}\cos\left(\tfrac{(2m-1)\pi}{2L_c}x\right) \tag{3.36}$$

change when we increase M and thus include more of the terms in the sums.

The difference is defined as

$$D_M = \frac{\|s_M - s_{M-1}\|}{\|s_M\|} + \frac{\|c_M - c_{M-1}\|}{\|c_M\|}, \tag{3.37}$$

where s_M and c_M are the partial sums (3.35) and (3.36) at time $t = 2.001$ ms when $t^* = 2.0$ ms.

M	D_M
2	8.480e-03
4	5.721e-04
6	4.642e-05
8	2.448e-06
10	7.322e-08
12	1.174e-09
14	9.792e-12
16	4.180e-14
18	9.030e-17
20	9.803e-20

Table 3.2: Observing how the partial sums (3.35) and (3.36) change with M. The parameters are given in Table 2.1 and Table 3.1, and the difference D_M is calculated from (3.37).

We calculate the difference analytically using the norm

$$\|u\| = \sqrt{\int_{\Omega_u} u^2 dx},$$

where Ω_u is Ω_s and Ω_c for s and c, respectively.

Studying Table 3.2, we observe that the difference between the approximations and thereby the effect of including more terms becomes small quite quickly. We conclude that we can compute the solutions to desired accuracy by adding a sufficient number of terms. In the calculations and plots presented in Chapter 5, the analytical solutions are given by (3.35) and (3.36) with $M = 20$.

The special case with constant $\bar{g}$

After finding analytical solutions for an open channel above, we considered the special case where the parameters were chosen so that $\bar{g}(t)$ was a constant specified by the model. In that case, we found that solutions on the form (3.17) and (3.18) were expected steady-state solutions of the 1D problem with an open channel and some chosen initial conditions.

To make $\bar{g}$ constant, we let $a_1 = b_1 = 0$. For the solutions for after the channel is closed, this means that we get

$$\alpha_m = \frac{8L_s a_4}{(2m-1)^2\pi^2}, \tag{3.38}$$

$$\beta_m = -\frac{8L_c b_4}{(2m-1)^2\pi^2}, \tag{3.39}$$

where a_4 and b_4 are given by (3.19) and (3.20).

Assuming that the solutions have reached steady state when the channel is closed, the solutions (3.35) and (3.36) where α_m and β_m are given by (3.38) and (3.39) can then be used as analytical solutions for after the channel is closed for the original 1D problem with a constant $\bar{g}$.

3.2 Analytical solutions of the 2D problem

In this section, we derive analytical solutions to the 2D problem described in Section 2.1.4. We again consider the case where the channel is open until $t = t^*$ and then closed.

3.2.1 Open channel

Analogously to the 1D problem, we find that solutions on the form

$$s(t,x,y) = a_1 e^{-a_2 t}\sin\left(a_3(x+L_s)\right) + a_4 x + a_5 + a_6\cosh(a_7 x)\cos(a_8 y), \tag{3.40}$$

$$c(t,x,y) = b_1 e^{-b_2 t}\sin\left(b_3(x-L_c)\right) + b_4 x + b_5 + b_6\cosh(b_7 x)\cos(b_8 y) \tag{3.41}$$

fulfil the equations and boundary conditions (2.17)−(2.24) of the 2D problem for an open channel if the parameters, initial conditions and $\bar{g}$ are adjusted as explained below.

Fulfilling the diffusion equations

First, the solutions must satisfy the diffusion equations

$$\frac{\partial s}{\partial t} = k_s\left(\frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2}\right), \tag{3.42}$$

$$\frac{\partial c}{\partial t} = k_c\left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2}\right). \tag{3.43}$$

Differentiating s on the form (3.40), we find

$$\begin{aligned}
\frac{\partial s}{\partial t} =& - a_1 a_2 e^{-a_2 t} \sin\left(a_3(x+L_s)\right),\\
\frac{\partial^2 s}{\partial x^2} =& - a_1 a_3^2 e^{a_2 t} \sin\left(a_3(x+L_s)\right) + a_6 a_7^2 \cosh(a_7 x)\cos(a_8 y),\\
\frac{\partial^2 s}{\partial y^2} =& - a_6 a_8^2 \cosh(a_7 x)\cos(a_8 y),
\end{aligned}$$

and inserting this in (3.42) yields:

$$-a_1(a_2 - k_s a_3^2)e^{-a_2 t}\sin\left(a_3(x+L_s)\right) = k_s a_6(a_7^2 - a_8^2)\cosh(a_7 x)\cos(a_8 y).$$

For this to be satisfied for all x and t, we need

$$a_2 = k_s a_3^2, \tag{3.44}$$

$$a_7^2 = a_8^2. \tag{3.45}$$

From (3.43) we similarly get the conditions

$$b_2 = k_c b_3^2, \tag{3.46}$$

$$b_7^2 = b_8^2. \tag{3.47}$$

Fulfilling the boundary conditions on the y-boundary

Next, the solutions should satisfy the no-flux boundary conditions

$$\frac{\partial s}{\partial y}(t,x,0) = \frac{\partial s}{\partial y}(t,x,Ly) = \frac{\partial c}{\partial y}(t,x,0) = \frac{\partial c}{\partial y}(t,x,Ly) = 0.$$

Differentiating s on the form (3.40), we find

$$\frac{\partial s}{\partial y} = -a_6 a_8 \cosh(a_7 x)\sin(a_8 y).$$

Inserting $y = 0$, we see that the boundary condition $\frac{\partial s}{\partial y}(t,x,0) = 0$ is fulfilled, but to make sure $\frac{\partial s}{\partial y}(t,x,L_y) = 0$, we let

$$a_8 = \frac{k_1 \pi}{L_y} \tag{3.48}$$

for some integer k_1.

Similarly, $\frac{\partial c}{\partial y}(t,x,0) = 0$ is fulfilled for any choice of parameters, while $\frac{\partial c}{\partial y}(t,x,L_y) = 0$ requires

$$b_8 = \frac{k_2\pi}{L_y} \tag{3.49}$$

for some integer k_2.

Fulfilling the Dirichlet boundary conditions on the x-boundary

The solutions must also satisfy the Dirichlet boundary conditions

$$\begin{aligned} s(t,-L_s,y) &= s_0 + A\cos(By), \\ c(t,L_c,y) &= c_0 + C\cos(Dy). \end{aligned}$$

Inserting $x = -L_s$ in s on the form (3.40) gives

$$s(t,-L_s,y) = -a_4L_s + a_5 + a_6\cosh(a_7L_s)\cos(a_8y).$$

For this to equal $s_0 + A\cos(By)$, we need

$$-a_4L_s + a_5 = s_0, \tag{3.50}$$

$$a_6\cosh(a_7L_s) = A, \tag{3.51}$$

$$a_8 = B. \tag{3.52}$$

For c we similarly get the conditions

$$b_4L_c + b_5 = c_0, \tag{3.53}$$

$$b_6\cosh(b_7L_c) = C, \tag{3.54}$$

$$b_8 = D. \tag{3.55}$$

Fulfilling the Neumann boundary conditions on the x-boundary

Finally, the solutions should satisfy the Neumann boundary conditions

$$\begin{aligned} k_s\frac{\partial s}{\partial x}(t,0,y) &= \bar{g}(t,y)(c(t,0,y) - s(t,0,y)), \\ k_c\frac{\partial c}{\partial x}(t,0,y) &= \bar{g}(t,y)(c(t,0,y) - s(t,0,y)). \end{aligned}$$

We again note that these imply

$$k_s \frac{\partial s}{\partial x}(t,0,y) = k_c \frac{\partial c}{\partial x}(t,0,y).$$

Inserting s and c on the form (3.40) and (3.41), we get

$$k_s \left(a_1 a_3 e^{-a_2 t} \cos\left(a_3 L_s\right) + a_4 \right) = k_c \left(b_1 b_3 e^{-b_2 t} \cos\left(b_3 L_c\right) + b_4 \right),$$

and for this to be satisfied for all x and t, we must have:

$$k_s a_1 a_3 \cos(L_s a_3) = k_c b_1 b_3 \cos(L_c b_3), \tag{3.56}$$

$$a_2 = b_2, \tag{3.57}$$

$$k_s a_4 = k_c b_4. \tag{3.58}$$

When these conditions are satisfied, the Neumann boundary conditions hold if we adjust $\bar{g}(t,y)$ to be

$$\begin{aligned}\bar{g}(t,y) &= \frac{k_c \frac{\partial c}{\partial x}(t,0,y)}{c(t,0,y) - s(t,0,y)} = \frac{k_s \frac{\partial s}{\partial x}(t,0,y)}{c(t,0,y) - s(t,0,y)} \\ &= \frac{k_s \left(a_1 a_3 e^{-a_2 t} \cos\left(a_3 L_s\right) + a_4\right)}{b_1 e^{-b_2 t} \sin\left(-b_3 L_c\right) + b_5 + b_6 \cos(b_8 y) - a_1 e^{-a_2 t} \sin\left(a_3 L_s\right) - a_5 - a_6 \cos(a_8 y)}.\end{aligned} \tag{3.59}$$

This means that $\bar{g}$ has to be a constructed function, now given by (3.59), in order for (3.40) and (3.41) to be solutions of the 2D problem (2.17) – (2.24).

Initial conditions

The initial conditions are specified by

$$\begin{aligned}s(0,x,y) &= s_{\text{init}}(x,y) \\ &= a_1 \sin\left(a_3(x + L_s)\right) + a_4 x + a_5 + a_6 \cosh(a_7 x) \cos(a_8 y),\end{aligned} \tag{3.60}$$

$$\begin{aligned}c(0,x,y) &= c_{\text{init}}(x,y) \\ &= b_1 \sin\left(b_3(x - L_c)\right) + b_4 x + b_5 + b_6 \cosh(b_7 x) \cos(b_8 y),\end{aligned} \tag{3.61}$$

where we have simply inserted $t = 0$ in s and c on the form (3.40) and (3.41).

Finding the solutions

Bringing everything together, we end up with 15 equations (3.44) – (3.58) that must be fulfilled. This can be achieved by choosing some numbers a_1, a_3, a_4, A and C and some integers k_1 and k_2 and adjusting the rest of the parameters to satisfy the 15 equations.

From (3.48), (3.49), (3.52) and (3.55) we get

$$a_8 = B = \frac{k_1 \pi}{L_y}, \qquad b_8 = D = \frac{k_2 \pi}{L_y}. \tag{3.62}$$

The conditions (3.45) and (3.47) are satisfied if

$$a_7 = a_8, \qquad b_7 = b_8, \tag{3.63}$$

and (3.51) and (3.54) results in

$$a_6 = \frac{A}{\cosh(a_7 L_s)}, \qquad b_6 = \frac{C}{\cosh(b_7 L_c)}. \tag{3.64}$$

The system (3.44), (3.46) and (3.57) can be solved by

$$a_2 = k_s a_3^2, \qquad b_2 = a_2, \qquad b_3 = \sqrt{\frac{k_s a_3^2}{k_c}}, \tag{3.65}$$

condition (3.58) gives

$$b_4 = \frac{k_s}{k_c} a_4, \tag{3.66}$$

and (3.56) results in

$$b_1 = \frac{k_s a_1 a_3 \cos(L_s a_3)}{k_c b_3 \cos(L_c b_3)}. \tag{3.67}$$

Finally, (3.50) and (3.53) is solved by

$$a_5 = s_0 + L_s a_4, \qquad b_5 = c_0 - L_c b_4. \tag{3.68}$$

We see that we get the same conditions for $a_1 - a_5$ and $b_1 - b_5$ as we did for the solutions of the 1D problem and some new conditions for the new parameters introduced in the 2D solutions. If these conditions are fufilled and $\bar{g}$ and the initial conditions are adjusted as explained above, (3.40) and (3.41) are solutions of the 2D problem (2.17) – (2.24).

Parameter	Value
A	100 μM
C	-10 μM
k_1	2
k_2	2

Table 3.3: Parameters used in the 2D simulations. The rest of the parameters are given in Table 2.1 and Table 3.1 and by (3.13) – (3.16) and (3.62) – (3.68).

Choice of free parameters

In the 2D solutions constructed above, we ended up with seven free parameters a_1, a_3, a_4, A, C, k_1 and k_2. In the simulations presented in Chapter 5, we use the same values of a_1, a_3 and a_4 as for the 1D simulations. We also use the same value of t^*. The values are specified in Table 3.1.

The amplitudes A and C in the Dirichlet boundary conditions are chosen to be 100 μM and -10 μM, respectively, motivated by the magnitude of the decrease and increase in calcium concentration due to release through the RyR channel observed in the 1D simulations. The integers k_1 and k_2 in the Dirichlet boundary conditions are both chosen to be 2, representing a decreased or increased concentration in the middle of the boundary similar to the RyR boundary in the 3D problem.

3.2.2 Closed channel

To find solutions of the 2D problem (2.17)–(2.24) when the channel is closed, we again use Fourier's method. We solve the problem in two steps:

- First, we find solutions $s_E(x, y)$ and $c_E(x, y)$ of the steady-state problems
$$\frac{\partial s_E}{\partial x^2} + \frac{\partial s_E}{\partial y^2} = 0,$$
$$\frac{\partial c_E}{\partial x^2} + \frac{\partial c_E}{\partial y^2} = 0,$$
with the correct boundary conditions (2.19) – (2.24).
- Then, we find solutions $s_H(t, x, y)$ and $c_H(t, x, y)$ of the full equations
$$\frac{\partial s_H}{\partial t} = k_s \left(\frac{\partial s_H}{\partial x^2} + \frac{\partial s_H}{\partial y^2} \right),$$
$$\frac{\partial c_H}{\partial t} = k_c \left(\frac{\partial c_H}{\partial x^2} + \frac{\partial c_H}{\partial y^2} \right),$$
with the correct version of the no-flux boundary conditions (2.21) – (2.24) and the homogeneous version of the Dirichlet boundary conditions (2.19) and (2.20).
- The solutions
$$s(x, y, t) = s_E(x, y) + s_H(t, x, y),$$
$$c(x, y, t) = c_E(x, y) + c_H(t, x, y)$$
will then satisfy the full system (2.17) – (2.24) with the correct boundary conditions.

We will start by finding the solutions s_E and s_H.

Steady-state solution for s

The steady-state solution s_E satisfies the equation
$$\frac{\partial s_E}{\partial x^2} + \frac{\partial s_E}{\partial y^2} = 0 \tag{3.69}$$

and the boundary conditions

$$s_E(-L_s, y) = h_s(y) = s_0 + A\cos(By), \tag{3.70}$$

$$\frac{\partial s_E}{\partial x}(0, y) = 0, \tag{3.71}$$

$$\frac{\partial s_E}{\partial y}(x, 0) = \frac{\partial s_E}{\partial y}(x, Ly) = 0. \tag{3.72}$$

We assume that we can find a solution of this system that can be separated into its x and y dependency in the sense that [24]

$$s_E(x, y) = X(x)Y(y). \tag{3.73}$$

Putting this into equation (3.69), we get the two ordinary differential equations:

$$X''(x) - \rho X(x) = 0,$$

$$Y''(y) + \rho Y(y) = 0.$$

Using the same approach as in Section 3.1.2, we find that

$$\rho_n = \left(\frac{n\pi}{L_y}\right)^2, \quad Y_n(y) = \cos\left(\frac{n\pi}{L_y}y\right), \quad X_n(x) = \cosh\left(\frac{n\pi}{L_y}x\right), \quad n = 0, 1, 2...$$

is a family of solutions that satisfies the equation (3.69) and the homogenous Neumann boundary conditions (3.71) – (3.72).

Using assumption (3.73), we find the particular solutions

$$s_{E,n}(x, y) = \cosh\left(\frac{n\pi}{L_y}x\right)\cos\left(\frac{n\pi}{L_y}y\right),$$

and by the principle of superposition,

$$s_E(x, y) = \gamma_0 + \sum_{n=1}^{\infty} \gamma_n \cosh\left(\frac{n\pi}{L_y}x\right)\cos\left(\frac{n\pi}{L_y}y\right)$$

fulfils equation (3.69) and the three boundary conditions (3.71) – (3.72) for any choice of the coefficients γ_n. What remains is to find an appropriate choice of values for γ_n so that the Dirichlet boundary condition (3.70) is also satisfied.

Inserting $x = -L_s$ in s_E, we get

$$s_E(-L_s, y) = \gamma_0 + \sum_{n=1}^{\infty} \gamma_n \cosh\left(\frac{n\pi L_s}{L_y}\right) \cos\left(\frac{n\pi}{L_y} y\right),$$

and we need to choose the values of γ_n so that this equals

$$h_s(y) = s_0 + A\cos(By) = s_0 + A\cos\left(\frac{k_1\pi}{L_y} y\right).$$

This is achieved by simply letting

$$\begin{aligned}\gamma_0 &= s_0,\\ \gamma_{k_1} &= \frac{A}{\cosh\left(\frac{k_1\pi L_s}{L_y}\right)},\\ \gamma_n &= 0, \qquad \text{for all other } n.\end{aligned}$$

The resulting steady-state solution is

$$s_E(x, y) = s_0 + a_6 \cosh(a_7 x) \cos(a_8 y),$$

where we have used the definition of a_6, a_7 and a_8 from (3.62) – (3.64).

Homogeneous solution for s

The next step is to find a solution $s_H(t, x, y)$ of the full equation

$$\frac{\partial s_H}{\partial t} = k_s \left(\frac{\partial s_H}{\partial x^2} + \frac{\partial s_H}{\partial y^2}\right) \tag{3.74}$$

with the correct boundary conditions

$$\frac{\partial s_H}{\partial x}(t, 0, y) = 0, \tag{3.75}$$

$$\frac{\partial s_H}{\partial y}(t, x, 0) = \frac{\partial s_H}{\partial y}(t, x, Ly) = 0 \tag{3.76}$$

and the homogenous version of the Dirichlet boundary condition

$$s_H(t, -L_s, y) = 0. \tag{3.77}$$

Using separation of variables and the principle of superposition, we find that the solution

$$s_H(t,x,y) = \sum_{m=1}^{\infty}\sum_{n=0}^{\infty} \alpha_{mn} e^{-\lambda_{mn}(t-t^*)} \cos\left(\tfrac{(2m-1)\pi}{2L_s}x\right) \cos\left(\tfrac{n\pi}{L_y}y\right), \tag{3.78}$$

where

$$\lambda_{mn} = k_s\left(\left(\tfrac{(2m-1)\pi}{2L_s}\right)^2 + \left(\tfrac{n\pi}{L_y}\right)^2\right),$$

is a solution of the system (3.74) – (3.77).

To make this solution satisfy our initial condition, we again assume that the initial condition $s_H(t^*,x,y) = f_s(x,y)$ can be expressed as a Fourier series:

$$s_H(t^*,x,y) = f_s(x,y) = \sum_{m=1}^{\infty}\sum_{n=0}^{\infty} \alpha_{mn} \cos\left(\tfrac{(2m-1)\pi}{2L_s}x\right) \cos\left(\tfrac{n\pi}{L_y}y\right). \tag{3.79}$$

To find appropriate values of α_{mn}, we first recall that for $m,n>0$, we have:

$$\int_{-L_s}^{0} \cos\left(\tfrac{(2m-1)\pi}{2L_s}x\right) \cos\left(\tfrac{(2n-1)\pi}{2L_s}x\right) dx = \begin{cases} L_s/2 & \text{if } m=n, \\ 0 & \text{if } m \neq n. \end{cases} \tag{3.80}$$

We also note that

$$\int_{0}^{L_y} \cos\left(\tfrac{m\pi}{L_y}y\right) \cos\left(\tfrac{n\pi}{L_y}y\right) dy = \begin{cases} L_y & \text{if } m=n=0, \\ L_y/2 & \text{if } m=n\neq 0, \\ 0 & \text{if } m \neq n. \end{cases} \tag{3.81}$$

Using (3.79) – (3.81), we find

$$\int_{0}^{L_y}\int_{-L_s}^{0} f_s(x,y)\cos\left(\tfrac{(2k-1)\pi}{2L_s}x\right) dxdy = \sum_{n=0}^{\infty} \alpha_{kn}\frac{L_s}{2}\int_0^{L_y}\cos\left(\tfrac{n\pi}{L_y}y\right) dy = \alpha_{k0}\frac{L_sL_y}{2},$$

which gives

$$\alpha_{m0} = \frac{2}{L_sL_y}\int_{0}^{L_y}\int_{-L_s}^{0} f_s(x,y)\cos\left(\tfrac{(2m-1)\pi}{2L_s}x\right) dxdy. \tag{3.82}$$

If we now assume $l \neq 0$, we similarly find

$$\int_{0}^{L_y}\int_{-L_s}^{0} f_s(x,y)\cos\left(\tfrac{(2k-1)\pi}{2L_s}x\right)\cos\left(\tfrac{l\pi}{L_y}y\right) dxdy$$
$$= \sum_{n=0}^{\infty} \alpha_{kn}\frac{L_s}{2}\int_0^{Ly}\cos\left(\tfrac{l\pi}{L_y}y\right)\cos\left(\tfrac{n\pi}{L_y}y\right) dy = \alpha_{kl}\frac{L_sL_y}{4},$$

which gives

$$\alpha_{mn} = \frac{4}{L_s L_y} \int_0^{L_y} \int_{-L_s}^{0} f_s(x,y) \cos\left(\tfrac{(2m-1)\pi}{2L_s} x\right) \cos\left(\tfrac{n\pi}{L_y} y\right) dx dy \tag{3.83}$$

for $n \geq 1$.

We are now interested in finding a solution for a closed channel given that the channel has been open until $t = t^*$ and that the solution for an open channel is on the form studied in the previous section, i.e

$$s(t,x,y) = a_1 e^{-a_2 t} \sin\left(a_3(x + L_s)\right) + a_4 x + a_5 + a_6 \cosh(a_7 x) \cos(a_8 y).$$

This means that the initial condition in our case is:

$$s(t^*,x,y) = a_1 e^{-a_2 t^*} \sin\left(a_3(x + L_s)\right) + a_4 x + a_5 + a_6 \cosh(a_7 x) \cos(a_8 y).$$

Since we have assumed

$$s(t,x,y) = s_H(t,x,y) + s_E(x,y),$$

the initial condition $f_s(x,y)$ for $s_H(t,x,y)$ is found by:

$$\begin{aligned} f_s(x,y) &= s(t^*,x,y) - s_E(x,y) \\ &= a_1 e^{-a_2 t^*} \sin\left(a_3(x + L_s)\right) + a_4 x + a_5 + a_6 \cosh(a_7 x) \cos(a_8 y) \\ &\quad - s_0 - a_6 \cosh(a_7 x) \cos(a_8 y) \\ &= a_1 e^{-a_2 t^*} \sin\left(a_3(x + L_s)\right) + a_4 x + a_5 - s_0. \end{aligned}$$

The coefficients can now be calculated using (3.82) and (3.83). We find

$$\begin{aligned} \alpha_{m0} &= -\frac{8L_s a_1 a_3 e^{-a_2 t^*} \cos(a_3 L_s)}{4L_s^2 a_3^2 - (2m-1)^2\pi^2} + \frac{8L_s a_4}{(2m-1)^2 \pi^2}, \\ \alpha_{mn} &= 0 \qquad \text{for } n \geq 1 \end{aligned}$$

and end up with the solution

$$s_H(t,x,y) = \sum_{m=1}^{\infty} \alpha_{m0} e^{-\lambda_{m0}(t-t^*)} \cos\left(\tfrac{(2m-1)\pi}{2L_s} x\right),$$

which consists of the same components and coefficients as the sum in the solution of the 1D problem.

Final solutions

In the end, we add $s(t,x,y) = s_E(x,y) + s_H(t,x,y)$ and find the final solution

$$s(t,x,y) = s_0 + a_6 \cosh(a_7 x)\cos(a_8 y) + \sum_{m=1}^{\infty} \alpha_m e^{-\lambda_m (t-t^*)} \cos\left(\tfrac{(2m-1)\pi}{2L_s} x\right), \tag{3.84}$$

where

$$\alpha_m = -\frac{8L_s a_1 a_3 e^{-a_2 t^*}\cos(a_3 L_s)}{4L_s^2 a_3^2 - (2m-1)^2\pi^2} + \frac{8L_s a_4}{(2m-1)^2\pi^2}$$

and

$$\lambda_m = k_s \left(\tfrac{(2m-1)\pi}{2L_s}\right)^2.$$

Similarly, we can derive the solution

$$c(t,x,y) = c_0 + b_6 \cosh(b_7 x)\cos(b_8 y) + \sum_{m=1}^{\infty} \beta_m e^{-\eta_m (t-t^*)} \cos\left(\tfrac{(2m-1)\pi}{2L_c} x\right), \tag{3.85}$$

where

$$\beta_m = \frac{8L_c b_1 b_3 e^{-b_2 t^*}\cos(b_3 L_c)}{4L_c^2 b_3^2 - (2m-1)^2\pi^2} - \frac{8L_c b_4}{(2m-1)^2\pi^2}$$

and

$$\eta_m = k_c \left(\tfrac{(2m-1)\pi}{2L_c}\right)^2$$

for the concentration in the dyad.

To summarize, this means that the functions (3.84) and (3.85) are solutions of the 2D problem for a closed channel, given that the channel has been open and the solutions on the form (3.40) and (3.41) until $t = t^*$.

Analogously to the 1D case, we assume that we can represent these solutions to desired accuracy using a finite number of terms in the infinite Fourier sums. In the simulations presented in Chapter 5, we include the terms for $m = 1, ..., 20$ in the representations of the analytical solutions.

Chapter 4

Numerical methods

In this chapter, we describe the numerical methods used in the simulations of calcium dynamics. The 1D and 2D problems are solved using finite difference methods, and the 3D problem is solved using a finite volume method.

4.1 Finite difference method

In the finite difference method, we introduce a grid of discrete points in time and space and seek approximations of the function values of the solution in these discrete points [11].

At the grid points, the derivatives are approximated by linear combinations of the function values in the nearby points. These approximations are called finite differences and can be found using a finite number of terms in Taylor series expansions.

To define the finite differences used in our calculations, we consider some four times continuously differentiable function $f(x)$. The Taylor series expansion of f can be expressed as [24]

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f^{(3)}(x) + \frac{h^4}{24}f^{(4)}(x+\xi_1) \quad (4.1)$$

for some $h > 0$ and $\xi_1 \in [0, h]$.

Similarly, we have

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f^{(3)}(x) + \frac{h^4}{24}f^{(4)}(x-\xi_2) \quad (4.2)$$

for some for some $h > 0$ and $\xi_2 \in [0, h]$.

The finite differences are found by rewriting these expressions and eliminating some of the last terms.

Rearranging (4.1) gives

$$\begin{aligned} f'(x) &= \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h) \\ &\approx \frac{f(x+h) - f(x)}{h}, \end{aligned} \tag{4.3}$$

and rearranging (4.2) gives

$$\begin{aligned} f'(x) &= \frac{f(x) - f(x-h)}{h} + \mathcal{O}(h) \\ &\approx \frac{f(x) - f(x-h)}{h}. \end{aligned} \tag{4.4}$$

Subtracting (4.2) from (4.1), we get

$$\begin{aligned} f'(x) &= \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2) \\ &\approx \frac{f(x+h) - f(x-h)}{2h}, \end{aligned} \tag{4.5}$$

and adding (4.1) and (4.2) gives

$$\begin{aligned} f''(x) &= \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \mathcal{O}(h^2) \\ &\approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \end{aligned} \tag{4.6}$$

We will now use these finite differences to derive schemes to solve the 1D and 2D problems.

4.1.1 Grid for the 1D finite difference schemes

In order to set up a finite difference scheme for the 1D models described in Chapter 2, we first introduce the following grid in time and space;

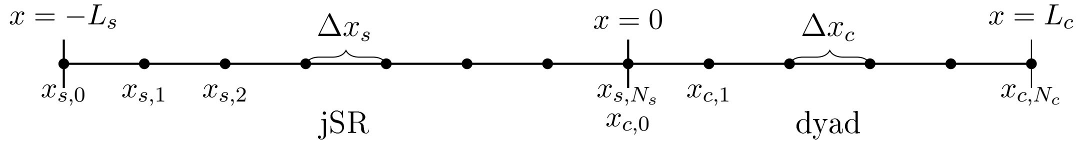

Figure 4.1: Illustration of the spatial 1D grid used in the finite difference schemes. We seek approximations of the solutions s and c in the grid points marked with black dots for every time step t_n.

$$\begin{aligned} t_n &= n\Delta t, && n = 0, 1, ..., \\ x_{s,i} &= -L_s + i\Delta x_s && i = 0, 1, ..., N_s, \\ x_{c,i} &= i\Delta x_c && i = 0, 1, ..., N_c, \end{aligned}$$

where

$$\Delta x_s = \frac{L_s}{N_s}, \qquad \Delta x_c = \frac{L_c}{N_c}.$$

We are trying to find approximations of the solutions s and c in these discrete points. These solutions can be collected in the vectors

$$\mathbf{s}^n = \begin{pmatrix} s_0^n \\ \vdots \\ s_{N_s}^n \end{pmatrix}, \qquad \mathbf{c}^n = \begin{pmatrix} c_0^n \\ \vdots \\ c_{N_c}^n \end{pmatrix}, \qquad n = 0, 1, ...,$$

where s_i^n denotes the numerical approximation of $s(t_n, x_{s,i})$ and c_i^n denotes the numerical approximation of $c(t_n, x_{c,i})$.

Since the solutions are known on the Dirichlet part of the boundary, the unknowns can be collected in the truncated vectors:

$$\hat{\mathbf{s}}^n = \begin{pmatrix} s_1^n \\ \vdots \\ s_{N_s}^n \end{pmatrix}, \qquad \hat{\mathbf{c}}^n = \begin{pmatrix} c_0^n \\ \vdots \\ c_{N_c-1}^n \end{pmatrix}, \qquad n = 0, 1, \ldots$$

The inital conditions can easily be discretized by setting

$$\begin{aligned} s_i^0 &= s_{\text{init}}(x_{s,i}), \quad i = 0, ..., N_s, \\ c_i^0 &= c_{\text{init}}(x_{c,i}), \quad i = 0, ..., N_c. \end{aligned}$$

To calculate s_i^n and c_i^n for $n \geq 1$, we use schemes derived by replacing derivatives with finite differences.

4.1.2 1D explicit finite difference scheme

We first derive an explicit finite difference scheme used to calculate approximations of s, the calcium concentration in the jSR.

General scheme

By replacing the derivatives in the diffusion equation

$$\frac{\partial s}{\partial t} = k_s \frac{\partial^2 s}{\partial x^2}$$

at some point $(t_n,\ x_{s,i})$ with the finite differences (4.3) and (4.6), we get

$$\frac{s_i^{n+1} - s_i^n}{\Delta t} = k_s \frac{s_{i+1}^n - 2s_i^n + s_{i-1}^n}{\Delta x_s^2}.$$

Solving for the unknowns s_i^{n+1}, we get the explicit formula

$$s_i^{n+1} = r_s s_{i-1}^n + (1 - 2r_s) s_i^n + r_s s_{i+1}^n,$$

where

$$r_s = k_s \frac{\Delta t}{\Delta x_s^2}. \tag{4.7}$$

Dirichlet boundary condition

The Dirichlet boundary condition $s(t, -L_s) = s_0$ can be discretized by

$$s_0^n = s_0, \quad n = 1, 2, \ldots$$

By inserting this in the general scheme for $i = 1$, we get the special formula:

$$s_1^{n+1} = (1 - 2r_s) s_1^n + r_s s_2^n + r_s s_0. \tag{4.8}$$

Neumann boundary condition

For the Neumann boundary condition

$$k_s \frac{\partial s}{\partial x}(t, 0) = g(t)(c(t, 0) - s(t, 0)),$$

we can replace the derivative with the centred difference (4.5). This gives

$$k_s \frac{s^n_{N_s+1} - s^n_{N_s-1}}{2\Delta x_s} = g(t_n)(c^n_0 - s^n_{N_s}),$$

which can be rewritten as:

$$s^n_{N_s+1} = s^n_{N_s-1} + \frac{2\Delta x_s}{k_s} g(t_n) \left(c^n_0 - s^n_{N_s}\right). \tag{4.9}$$

Inserting this in the general scheme gives a special formula for $i = N_s$:

$$s^{n+1}_{N_s} = 2r_s s^n_{N_s-1} + (1-2r_s)s^n_{N_s} + \frac{2\Delta t}{\Delta x_s} g(t_n) \left(c^n_0 - s^n_{N_s}\right).$$

Full scheme on matrix form

If we combine the schemes for the boundary and the inner points, we can write the full scheme as a matrix problem,

$$\hat{\mathbf{s}}^{n+1} = A_s \hat{\mathbf{s}}^n + \mathbf{d}^n_s,$$

to be calculated for every time step $n = 0, 1, ...$

The matrix $A_s \in \mathbb{R}^{N_s \times N_s}$ and the vector $\mathbf{d}^n_s \in \mathbb{R}^{N_s}$ are given by

$$A_s = \begin{pmatrix} 1-2r_s & r_s & 0 & \cdots & 0 \\ r_s & 1-2r_s & r_s & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & r_s & 1-2r_s & r_s \\ 0 & \cdots & 0 & 2r_s & 1-2r_s \end{pmatrix}, \quad \mathbf{d}^n_s = \begin{pmatrix} r_s s_0 \\ 0 \\ \vdots \\ 0 \\ \frac{2\Delta t}{\Delta x_s} g(t_n) \left(c^n_0 - s^n_{N_s}\right) \end{pmatrix}.$$

Using the same approach to discretize the equation and boundary conditions for the calcium concentration in the dyad, we get the matrix problem

$$\hat{\mathbf{c}}^{n+1} = A_c \hat{\mathbf{c}}^n + \mathbf{d}^n_c,$$

where $A_c \in \mathbb{R}^{N_c \times N_c}$ and $\mathbf{d}^n_c \in \mathbb{R}^{N_c}$ are given by

$$A_c = \begin{pmatrix} 1-2r_c & 2r_c & 0 & \cdots & 0 \\ r_c & 1-2r_c & r_c & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & r_c & 1-2r_c & r_c \\ 0 & \cdots & 0 & r_c & 1-2r_c \end{pmatrix}, \quad \mathbf{d}^n_c = \begin{pmatrix} -\frac{2\Delta t}{\Delta x_c} g(t_n)(c^n_0 - s^n_{N_s}) \\ 0 \\ \vdots \\ 0 \\ r_c c_0 \end{pmatrix}$$

and

$$r_c = k_c \frac{\Delta t}{\Delta x_c^2}. \tag{4.10}$$

Stability condition for the explicit 1D finite difference scheme

For the explicit finite difference scheme to be stable, the discretization parameters need to satisfy a stability condition. This condition is set up to ensure that small errors in the initial conditions do not grow exponentially for each time step.

Assuming that $\mathbf{u}^n$ is the finite difference solution at time step n of the scheme with initial condition $\mathbf{u}^0$ and that $\bar{\mathbf{u}}^n$ is the solution of the same scheme with initial condition $\mathbf{u}^0 + \mathbf{e}^0$, we want to make sure the error $\mathbf{e}^n = \bar{\mathbf{u}}^n - \mathbf{u}^n$ satisfies

$$\|\mathbf{e}^n\| \leq C\|\mathbf{e}^0\| \tag{4.11}$$

for some vector norm $\|\cdot\|$ and some positive constant C independent of n, Δt and Δx. We will now consider how to make sure this condition is fulfilled for our coupled system using an approach described in [21].

We begin by writing the scheme as one large system on the form

$$\mathbf{u}^{n+1} = A\mathbf{u}^n + \mathbf{d}, \tag{4.12}$$

where

$$\mathbf{u}^n = \begin{pmatrix} \hat{\mathbf{s}}^n \\ \hat{\mathbf{c}}^n \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} r_s s_0 \\ 0 \\ \vdots \\ 0 \\ r_c c_0 \end{pmatrix},$$

$$A = \begin{pmatrix}
1-2r_s & r_s & 0 & & \cdots & 0 & \cdots & & \cdots & 0 \\
r_s & 1-2r_s & r_s & & & & & & & \vdots \\
 & \ddots & \ddots & \ddots & & & & & & \\
0 & & r_s & 1-2r_s & r_s & & & & & \\
\vdots & & & 2r_s & 1-2r_s-p_s^n & p_s^n & & \ddots & & \vdots \\
 & & \ddots & & p_c^n & 1-2r_c-p_c^n & 2r_c & & & \\
\vdots & & & & & r_c & 1-2r_c & r_c & & 0 \\
 & & & & & & \ddots & \ddots & \ddots & \\
 & & & & & & & r_c & 1-2r_c & r_c \\
0 & \cdots & & \cdots & 0 & \cdots & & 0 & r_c & 1-2r_c
\end{pmatrix},$$

and

$$p_s^n = \frac{2\Delta t}{\Delta x_s} g(t_n), \qquad p_c^n = \frac{2\Delta t}{\Delta x_c} g(t_n).$$

Similarly, the solution $\bar{\mathbf{u}}^{n+1}$ can be expressed as

$$\bar{\mathbf{u}}^{n+1} = A\bar{\mathbf{u}}^n + \mathbf{d}. \tag{4.13}$$

Subtracting (4.12) from (4.13), we get

$$\mathbf{e}^{n+1} = A\mathbf{e}^n,$$

which implies that

$$\mathbf{e}^n = A^n\mathbf{e}^0.$$

Consequently, we have

$$\|\mathbf{e}^n\| = \|A^n\mathbf{e}^0\| \leq \|A^n\|\|\mathbf{e}^0\| \leq \|A\|^n\|\mathbf{e}^0\|$$

for a submultiplicative matrix norm consistent with the vector norm, and we see that the stability condition (4.11) is satisfied if

$$\|A\| \leq 1. \tag{4.14}$$

Choosing the vector norm

$$\|\mathbf{u}\|_\infty = \max_i |x_i|$$

and matrix norm

$$\|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{i,j}|,$$

we can easily verify whether the condition (4.14) is satisfied by considering the row sums $R_i = \sum_{j=1}^{n} |a_{i,j}|$ of A. Here $a_{i,j}$ denotes the element in row i and column j of the matrix A.

For the matrix of our scheme, we have

$$\begin{aligned}
R_i &= |1-2r_s| + |r_s|, && i = 1,\\
R_i &= |r_s| + |1-2r_s| + |r_s|, && i = 2, ..., N_s - 1,\\
R_i &= |2r_s| + |1-2r_s-p_s^n| + |p_s^n|, && i = N_s,\\
R_i &= |p_c^n| + |1-2r_c-p_c^n| + |2r_c|, && i = N_s + 1,\\
R_i &= |r_c| + |1-2r_c| + |r_c|, && i = N_s + 2, ..., N_s + N_c - 1,\\
R_i &= |1-2r_c| + |r_c|, && i = N_s + N_c,
\end{aligned}$$

where all the values r_s, r_c, p_s^n and p_c^n are nonnegative since

$$r_s = k_s\frac{\Delta t}{\Delta x_s^2}, \qquad p_s^n = \frac{2\Delta t}{\Delta x_s}g(t_n),$$

$$r_c = k_c\frac{\Delta t}{\Delta x_c^2}, \qquad p_c^n = \frac{2\Delta t}{\Delta x_c}g(t_n),$$

and we assume that $g(t)$ is nonnegative for all $t \geq 0$.

Considering these expressions, we find that $R_i \leq 1$ for all i if Δt satisfies the condition

$$\Delta t \leq \min(\Delta t_1, \Delta t_2), \tag{4.15}$$

where

$$\Delta t_1 = \frac{1}{2}\frac{\Delta x_s^2}{k_s + \Delta x_s \bar{g}}, \tag{4.16}$$

$$\Delta t_2 = \frac{1}{2}\frac{\Delta x_c^2}{k_c + \Delta x_c \bar{g}} \tag{4.17}$$

and $\bar{g}$ is the maximum value of $g(t)$.

Considering, for instance, R_{N_s}, we first note that (4.15) and (4.16) imply that

$$2r_s + p_s^n \leq 1 \tag{4.18}$$

for all $n = 0, 1, 2...$ This means that

$$R_{N_s} = 2r_s + 1 - 2r_s - p_s^n + p_s^n = 1 \leq 1.$$

If (4.18) did not hold, i.e. if $2r_s + p_s^n > 1$, then

$$R_{N_s} = 2r_s - 1 + 2r_s + p_s^n + p_s^n = 2(2r_s + p_s^n) - 1 > 1.$$

Analogously, we can show that the remaining row sums, R_i, are all smaller than or equal to one if the stability condition (4.15) holds. As a result, we have $\|A\|_\infty \leq 1$, and the finite difference scheme is stable.

4.1.3 1D explicit finite difference scheme including buffers

Including buffers in the model as described in Section 2.2.2, we get the following system of equations:

$$\frac{\partial s}{\partial t} = k_s \frac{\partial^2 s}{\partial x^2} - R_s(s, b_s), \qquad x \in [-L_s, 0], \tag{4.19}$$

$$\frac{\partial b_s}{\partial t} = R_s(s, b_s), \qquad x \in [-L_s, 0], \tag{4.20}$$

$$\frac{\partial c}{\partial t} = k_c \frac{\partial^2 c}{\partial x^2} - R_c(c, b_c), \qquad x \in [0, L_c], \tag{4.21}$$

$$\frac{\partial b_c}{\partial t} = k_b \frac{\partial^2 b_c}{\partial x^2} + R_c(c, b_c), \qquad x \in [0, L_c]. \tag{4.22}$$

We get the two new unknowns b_s and b_c, and the numerical approximations of these solutions can be gathered in the vectors

$$\mathbf{b}_c^n = \begin{pmatrix} b_{c,0}^n \\ \vdots \\ b_{c,N_c}^n \end{pmatrix}, \qquad \mathbf{b}_s^n = \begin{pmatrix} b_{s,0}^n \\ \vdots \\ b_{s,N_s}^n \end{pmatrix},$$

where $b_{c,i}^n$ denotes the numerical approximation of $b_c(t_n, x_{c,i})$ and $b_{s,i}^n$ denotes the numerical approximation of $b_s(t_n, x_{s,i})$.

Schemes for s and c

The reaction terms in the equations for s and c are incorporated in the schemes by adding reaction vectors $\hat{\mathbf{q}}_s^n$ and $\hat{\mathbf{q}}_c^n$ to the schemes. We define

$$\mathbf{q}_s^n = \Delta t \begin{pmatrix} R_s(s_0^n, b_{s,0}^n) \\ \vdots \\ R_s(s_{N_s}^n, b_{s,N_s}^n) \end{pmatrix}, \quad \mathbf{q}_c^n = \Delta t \begin{pmatrix} R_c(c_0^n, b_{c,0}^n) \\ \vdots \\ R_c(c_{N_c}^n, b_{c,N_c}^n) \end{pmatrix}$$

and let $\hat{\mathbf{q}}_s^n$ be the vector $\mathbf{q}_s^n$ except the first entry and $\hat{\mathbf{q}}_c^n$ be the vector $\mathbf{q}_c^n$ except the last entry.

The schemes for s and c can now be written as:

$$\begin{aligned} \hat{\mathbf{s}}^{n+1} &= A_s \hat{\mathbf{s}}^n + \mathbf{d}_s^n - \hat{\mathbf{q}}_s^n, \\ \hat{\mathbf{c}}^{n+1} &= A_c \hat{\mathbf{c}}^n + \mathbf{d}_c^n - \hat{\mathbf{q}}_c^n. \end{aligned}$$

Schemes for the buffers

The scheme for the concentration of calcium bound to buffer in the dyad is very similar, except that there are no-flux boundary conditions on the boundary. We get the scheme

$$\mathbf{b}_c^{n+1} = A_b \mathbf{b}_c^n + \mathbf{q}_c^n,$$

where the matrix $A_b \in \mathbb{R}^{(N_c+1)\times(N_c+1)}$ is given by

$$A_b = \begin{pmatrix} 1-2r_b & 2r_b & 0 & \cdots & 0 \\ r_b & 1-2r_b & r_b & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & r_b & 1-2r_b & r_b \\ 0 & \cdots & 0 & 2r_b & 1-2r_b \end{pmatrix}$$

and

$$r_b = k_b \frac{\Delta t}{\Delta x_c^2}.$$

In the equation (4.20) for the concentration of calcium bound to buffer in the jSR, we do not have a diffusion term. We discretize the time derivative using the forward difference (4.3) and get the scheme:

$$\mathbf{b}_s^{n+1} = \mathbf{b}_s^n + \mathbf{q}_s^n.$$

Stability condition for the scheme including buffers

Above we found a stability condition for the finite difference scheme for the 1D problem without buffers by demanding that a small error in the initial conditions should remain small for all time steps. This was done by writing the scheme as a linear system on matrix form and finding conditions so that $\|A\| \leq 1$. Since the finite difference scheme for the 1D problem including buffers includes the nonlinear terms $R_s(s_i^n, b_{s,i}^n)$ and $R_c(c_i^n, b_{c,i}^n)$, we can not find stability conditions for this scheme using the same approach.

Instead, we can try to find conditions to make sure the values of the numerical solutions are bounded to a limited interval for all time steps, thus ensuring that we e.g. do not get negative concentrations or concentrations larger than some bound. This approach to determine invariant regions for numerical solutions of finite difference schemes is described in [24].

Below we will show that the numerical solutions of the explicit finite difference scheme including buffers will remain in the invariant regions

$$0 \leq s_i^n \leq s^*, \qquad i = 0, ..., N_s, \tag{4.23}$$

$$0 \leq b_{s,i}^n \leq b_s^*, \qquad i = 0, ..., N_s, \tag{4.24}$$

$$0 \leq c_i^n \leq c^*, \qquad i = 0, ..., N_c, \tag{4.25}$$

$$0 \leq b_{c,i}^n \leq b_c^*, \qquad i = 0, ..., N_c, \tag{4.26}$$

where

$$s^* = c^*, \tag{4.27}$$

$$b_s^* = \frac{k_{\text{on}}^s s^* B_{\text{tot}}^s}{k_{\text{on}}^s s^* + k_{\text{off}}^s} < B_{\text{tot}}^s, \tag{4.28}$$

$$b_c^* = \frac{k_{\text{on}}^c c^* B_{\text{tot}}^c}{k_{\text{on}}^c c^* + k_{\text{off}}^c} < B_{\text{tot}}^c, \tag{4.29}$$

for all time steps if s_i^0, $b_{s,i}^0$, c_i^0 and $b_{c,i}^0$ are in these regions and Δt satisfies the condition

$$\Delta t \leq \min(\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4), \tag{4.30}$$

where

$$\Delta t_1 = \frac{1}{2}\frac{\Delta x_s^2}{k_s + \Delta x_s \bar{g} + \Delta x_s^2 k_{\text{on}}^s B_{\text{tot}}^s/2}, \tag{4.31}$$

$$\Delta t_2 = \frac{1}{k_{\text{on}}^s s^* + k_{\text{off}}^s}, \tag{4.32}$$

$$\Delta t_3 = \frac{1}{2}\frac{\Delta x_c^2}{k_c + \Delta x_c \bar{g} + \Delta x_c^2 k_{\text{on}}^c B_{\text{tot}}^c/2}, \tag{4.33}$$

$$\Delta t_4 = \frac{1}{2}\frac{\Delta x_c^2}{k_b + \Delta x_c^2 (k_{\text{on}}^c c^* + k_{\text{off}}^c)/2}. \tag{4.34}$$

We begin by writing the scheme for the numerical concentration of free calcium and calcium bound to buffer in the jSR on component form

$$\begin{aligned}
s_1^{n+1} &= (1-2r_s)s_1^n + r_s s_2^n + r_s s_0 - \Delta t R_s(s_1^n, b_{s,1}^n), \\
s_i^{n+1} &= r_s s_{i-1}^n + (1-2r_s)s_i^n + r_s s_{i+1}^n - \Delta t R_s(s_i^n, b_{s,i}^n), && i = 2, ..., N_s - 1, \\
s_{N_s}^{n+1} &= 2r_s s_{N_s-1}^n + (1-2r_s)s_{N_s}^n + p_s^n\left(c_0^n - s_{N_s}^n\right) - \Delta t R_s(s_{N_s}^n, b_{s,N_s}^n), \\
b_{s,i}^{n+1} &= b_{s,i}^n + \Delta t R_s(s_i^n, b_{s,i}^n), && i = 0, ..., N_s,
\end{aligned}$$

where

$$\begin{aligned}
R_s(s, b_s) &= k_{\text{on}}^s s\,(B_{\text{tot}}^s - b_s) - k_{\text{off}}^s b_s, \\
R_c(c, b_c) &= k_{\text{on}}^c c\,(B_{\text{tot}}^c - b_c) - k_{\text{off}}^c b_c.
\end{aligned}$$

Similarly, the scheme for the calcium concentration in the dyad is

$$\begin{aligned}
c_0^{n+1} &= (1-2r_c)c_0^n + 2r_c c_1^n - p_c^n\left(c_0^n - s_{N_s}^n\right) - \Delta t R_c(c_0^n, b_{c,0}^n), \\
c_i^{n+1} &= r_c c_{i-1}^n + (1-2r_c)c_i^n + r_c c_{i+1}^n - \Delta t R_c(c_i^n, b_{c,i}^n), && i = 1, ..., N_c - 2, \\
c_{N_c-1}^{n+1} &= r_c c_{N_c-2}^n + (1-2r_c)c_{N_c-1}^n + r_c c_0 - \Delta t R_c(c_{N_c-1}^n, b_{c,N_c-1}^n),
\end{aligned}$$

and the scheme for the concentration of calcium bound to buffer in the dyad is

$$\begin{aligned}
b_{c,0}^{n+1} &= (1-2r_b)b_{c,0}^n + 2r_b b_{c,1}^n + \Delta t R_c(c_0^n, b_{c,0}^n), \\
b_{c,i}^{n+1} &= r_b b_{c,i-1}^n + (1-2r_b)b_{c,i}^n + r_b b_{c,i+1}^n + \Delta t R_c(c_i^n, b_{c,i}^n), \quad i = 1, ..., N_c - 1, \\
b_{c,N_c}^{n+1} &= 2r_b b_{c,N_c-1}^n + (1-2r_b)b_{c,N_c}^n + \Delta t R_c(c_{N_c}^n, b_{c,N_c}^n).
\end{aligned}$$

These expressions for the solutions in the next time step can be expressed as functions of the numerical solutions in the previous time step. Considering, for instance, the scheme for the calcium concentration in the inner points of the jSR domain, we can write:

$$\begin{aligned} &s_i^{n+1} = S(s_{i-1}^n, s_i^n, s_{i+1}^n, b_{s,i}^n), \qquad i = 2, ..., N_s - 1, \\ &S(s_-, s, s_+, b) = r_s s_- + (1 - 2r_s)s + r_s s_+ - \Delta t R_s(s, b). \end{aligned}$$

We are now interested in ensuring that the function S is an increasing function with respect to all the variables. This is true if the derivative of S with respect to each variable is positive. Differentiating S and using (4.30) and (4.31), we find

$$\begin{aligned} &\frac{\partial S}{\partial s_-} = \frac{\partial S}{\partial s_+} = r_s > 0, \\ &\frac{\partial S}{\partial b} = \Delta t(k_{\text{on}}^s s + k_{\text{off}}) > 0, \\ &\frac{\partial S}{\partial s} = 1 - 2r_s - \Delta t k_{\text{on}}^s (B_{\text{tot}}^s - b) > 1 - 2r_s - \Delta t k_{\text{on}}^s B_{\text{tot}}^s \geq 0, \end{aligned}$$

for $s_-, s, s_+ \in (0, s^*)$ and $b \in (0, b_s^*)$. So, if (4.30) holds, S is an increasing function with respect to all the variables $s_-, s, s_+ \in [0, s^*]$ and $b \in [0, b_s^*]$.

Writing the expressions for s_1^{n+1} and $s_{N_s}^{n+1}$ as the functions

$$\begin{aligned} &s_1^{n+1} = S_1(s_1^n, s_2^n, b_{s,1}^n), \\ &S_1(s, s_+, b) = (1 - 2r_s)s + r_s s_+ + r_s s_0 - \Delta t R_s(s, b), \end{aligned}$$

$$\begin{aligned} &s_{N_s}^{n+1} = S_N(s_{N_s-1}^n, s_{N_s}^n, c_0^n, b_{s,N_s}^n), \\ &S_N(s_-, s, c, b) = 2r_s s_- + (1 - 2r_s)s + p_s^n\,(c - s) - \Delta t R_s(s, b) \end{aligned}$$

and differentiating with respect to each variable, we can similarly show that S_1 is an increasing function with respect to $s, s_+ \in [0, s^*]$, $b \in [0, b_s^*]$ and S_N is an increasing function with respect to $s_-, s \in [0, s^*]$, $c \in [0, c^*]$ and $b \in [0, b_s^*]$.

Analogously, the numerical concentration of calcium bound to buffer in the jSR for the next time step can be expressed as

$$\begin{aligned} &b_{s,i}^{n+1} = B(s_i^n, b_{s,i}^n), \\ &B(s, b) = b + \Delta t R_s(s, b). \end{aligned}$$

Differentiating B and using (4.28), (4.30) and (4.32), we find

$$\begin{aligned}\frac{\partial B}{\partial s} &= \Delta t k_{\text{on}}^{s}(B_{\text{tot}}^{s} - b) > 0,\\ \frac{\partial B}{\partial b} &= 1 - \Delta t(k_{\text{on}}^{s} s + k_{\text{off}}^{s}) > 1 - \Delta t(k_{\text{on}}^{s} s^{*} + k_{\text{off}}) \geq 0\end{aligned}$$

for $s \in (0, s^*)$ and $b \in (0, b_s^*)$, so B is an increasing function with respect to the variables $s \in [0, s^*]$ and $b \in [0, b_s^*]$.

Likewise, we can show that the condition (4.30) also ensures that the remaining expressions for the numerical solutions for the next time step can be expressed as increasing functions of the numerical solutions in the previous time step. We will now use this fact to show that (4.23) – (4.26) is an invariant region for the numerical solutions if (4.30) holds.

First, we assume that the solutions s_i^n, c_i^n, $b_{s,i}^n$ and $b_{c,i}^n$ from the previous time step are in the region (4.23) – (4.26). Considering the calcium concentration in the jSR, the numerical solutions at the next time step are given by:

$$\begin{aligned}s_1^{n+1} &= S_1(s_1^n, s_2^n, b_{s,1}^n), &&\\ s_i^{n+1} &= S(s_{i-1}^n, s_i^n, s_{i+1}^n, b_{s,i}^n), && i = 2, ..., N_s - 1\\ s_{N_s}^{n+1} &= S_N(s_{N_s-1}^n, s_{N_s}^n, c_0^n, b_{s,N_s}^n). &&\end{aligned}$$

Since S_1, S and S_N are increasing functions, (4.23) – (4.29) imply that

$$\begin{aligned}s_1^{n+1} &\geq S_1(0, 0, 0) = 0, &&\\ s_i^{n+1} &\geq S(0, 0, 0, 0) = 0, && i = 2, ..., N_s - 1,\\ s_{N_s}^{n+1} &\geq S_N(0, 0, 0, 0) = 0 &&\end{aligned}$$

and

$$\begin{aligned}s_1^{n+1} &\leq S_1(s^*, s^*, b_s^*) = s^*, &&\\ s_i^{n+1} &\leq S(s^*, s^*, s^*, b_s^*) = s^*, && i = 2, ..., N_s - 1,\\ s_{N_s}^{n+1} &\leq S_N(s^*, s^*, c^*, b_s^*) = s^*. &&\end{aligned}$$

Similarly, for $b_{s,i}^{n+1}$, we have that

$$b_{s,i}^{n+1} = B(s_i^n, b_{s,i}^n),$$

and since B is an increasing function, it follows from (4.23) – (4.29) that

$$\begin{aligned}b_{s,i}^{n+1} &\geq B(0, 0) = 0, && i = 0, ..., N_s,\\ b_{s,i}^{n+1} &\leq B(s^*, b_s^*) = b_s^*, && i = 0, ..., N_s.\end{aligned}$$

This means that we have

$$\begin{aligned} 0 \le s_i^{n+1} &\le s^*, & i &= 0, ..., N_s, \\ 0 \le b_{s,i}^{n+1} &\le b_s^*, & i &= 0, ..., N_s, \end{aligned}$$

and using similar arguments for the dyad concentrations, we find that

$$\begin{aligned} 0 \le c_i^{n+1} &\le c^*, & i &= 0, ..., N_c, \\ 0 \le b_{c,i}^{n+1} &\le b_c^*, & i &= 0, ..., N_c. \end{aligned}$$

By induction this means that if the initial numerical concentrations, s_i^0, $b_{s,i}^0$, c_i^0 and $b_{c,i}^0$, are in the region (4.23) – (4.26) and Δt fulfils the condition (4.30), then the numerical solutions will remain in the region (4.23) – (4.26) for all time steps. Given the initial conditions in our model (see Section 2.2.2), we can e.g. choose $s^* = c^* = s_0$.

4.1.4 1D implicit finite difference scheme

In the schemes derived so far, the solution vectors for new time steps are found directly by an explicit formula. A disadvantage with these schemes is that the discretization parameters have to fulfil stability conditions, requiring small time steps.

We will now derive an implicit finite difference scheme where a system of linear equations has to be solved for every time step. This results in more computations for each time step, but we avoid the stability conditions.

We consider the 1D problem without buffers described in Section 2.1.3. To derive an implicit scheme, we discretize the time derivative in the diffusion equations using the backward difference (4.4) instead of the forward difference (4.3).

For s we get

$$\frac{s_i^{n+1} - s_i^n}{\Delta t} = k_s \frac{s_{i+1}^{n+1} - 2s_i^{n+1} + s_{i-1}^{n+1}}{\Delta x_s^2},$$

which can be rewritten to the implicit system

$$-r_s s_{i-1}^{n+1} + (1 + 2r_s)s_i^{n+1} - r_s s_{i+1}^{n+1} = s_i^n.$$

Using the centred difference (4.5) to discretize the derivative in the Neumann boundary condition, we get

$$s_{N_s+1}^{n+1} = s_{N_s-1}^{n+1} + \frac{2\Delta x_s}{k_s} g(t_{n+1}) \left(c_0^{n+1} - s_{N_s}^{n+1}\right).$$

Inserting this in the general scheme yields

$$-2r_s s_{N_s-1}^{n+1} + \left(1 + 2r_s + p_s^{n+1}\right) s_N^{n+1} - p_s^{n+1} c_0^{n+1} = s_{N_s}^n,$$

where

$$p_s^{n+1} = \frac{2\Delta t}{\Delta x_s} g(t_{n+1}).$$

This equation involves the unknown c_0^{n+1}, and the system for $\hat{\mathbf{c}}^{n+1}$ will likewise involve $s_{N_s}^{n+1}$. To find the solutions $\hat{\mathbf{s}}^{n+1}$ and $\hat{\mathbf{c}}^{n+1}$, we therefore combine the unknowns into a single vector $\mathbf{u}^{n+1}$ defined by

$$\mathbf{u}^n = \left(s_1^n, \cdots, s_{N_s}^n, c_0^n, \cdots, c_{N_c-1}^n\right)^T. \tag{4.35}$$

Combining the schemes for the boundaries and inner points for both s and c, we end up with the system

$$A\mathbf{u}^{n+1} = \mathbf{u}^n + \mathbf{d} \tag{4.36}$$

to be solved for every time step $n = 0, 1, ...$

The matrix $A \in \mathbb{R}^{(N_s+N_c)\times(N_s+N_c)}$ is given by

$$A = \begin{pmatrix}
q_s & -r_s & 0 & & \cdots & 0 & \cdots & & \cdots & 0 \\
-r_s & q_s & -r_s & & & & & & & \vdots \\
 & \ddots & \ddots & \ddots & & & & & & \\
0 & & -r_s & q_s & -r_s & & & & & \\
\vdots & & & -2r_s & q_s + p_s^{n+1} & -p_s^{n+1} & & \ddots & & \vdots \\
 & & \ddots & & -p_c^{n+1} & q_c + p_c^{n+1} & -2r_c & & & \\
\vdots & & & & & -r_c & q_c & -r_c & & 0 \\
 & & & & & & \ddots & \ddots & \ddots & \\
 & & & & & & & -r_c & q_c & -r_c \\
0 & \cdots & & \cdots & 0 & \cdots & & 0 & -r_c & q_c
\end{pmatrix} \tag{4.37}$$

and the vector $\mathbf{d} \in \mathbb{R}^{(N_s+N_c)}$ is given by

$$\mathbf{d} = \begin{pmatrix} r_s s_0 \\ 0 \\ \vdots \\ 0 \\ r_c c_0 \end{pmatrix}, \tag{4.38}$$

where we use the notation

$$\begin{aligned} q_s &= 1 + 2r_s, \\ q_c &= 1 + 2r_c, \\ p_s^{n+1} &= \frac{2\Delta t}{\Delta x_s} g(t_{n+1}), \\ p_c^{n+1} &= \frac{2\Delta t}{\Delta x_c} g(t_{n+1}). \end{aligned}$$

Stability condition for the implicit 1D finite difference scheme

For the explicit finite difference schemes to be stable, we needed Δt to satisfy stability conditions. For the scheme for the 1D problem without buffers, this was derived by writing the scheme on the form $\mathbf{u}^{n+1} = A\mathbf{u}^n + \mathbf{d}$ and enforcing $\|A\| \leq 1$ (see page 52-54).

To check whether the implicit scheme is stable, we write it on the similar form

$$\mathbf{u}^{n+1} = A^{-1}\mathbf{u}^n + A^{-1}\mathbf{d},$$

where the vector $\mathbf{u}^n$ is given by (4.35), the matrix A is given by (4.37) and the vector $\mathbf{d}$ is given by (4.38). Using the same argument as for the explicit scheme, the implicit scheme is stable if $\|A^{-1}\|_\infty \leq 1$.

Considering the matrix A given by (4.37) , we observe that A is strictly diagonally dominant, i.e. that [7]

$$|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|,$$

for all i, where $a_{i,j}$ denotes the element in row i and column j of A. From [26] we have that in that case, $\|A^{-1}\|_\infty$ satisfies

$$\|A^{-1}\|_\infty < \frac{1}{\alpha},$$

where

$$\alpha = \min_i (|a_{i,i}| - \sum_{j \neq i} |a_{i,j}|).$$

For our matrix A, we have $\alpha = 1$ (see (4.37)), so $\|A^{-1}\|_\infty \leq 1$, and the implicit scheme is unconditionally stable. Nevertheless, the accuracy of the numerical solution will depend on the value of Δt, so we still need to choose a Δt that is small enough to give accurate simulations.

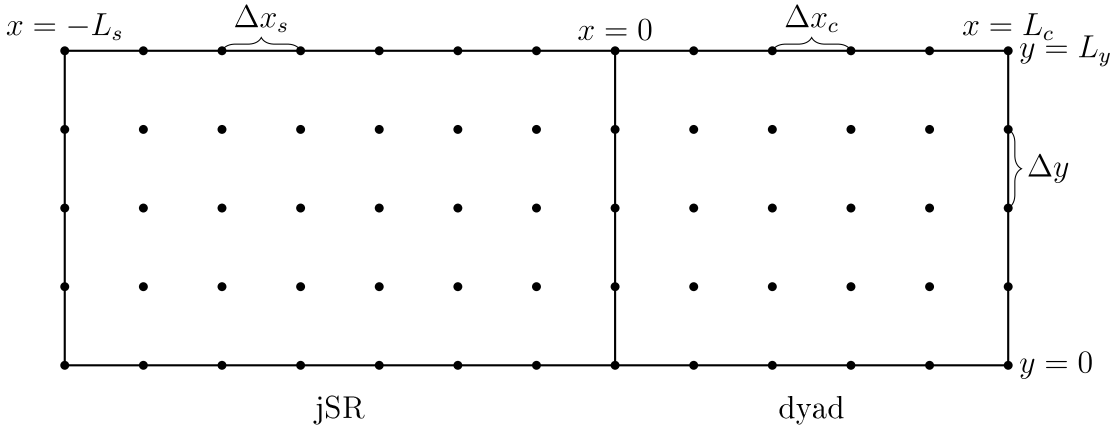

Figure 4.2: Illustration of the spatial 2D grid used in the finite difference scheme. We seek approximations of the solutions s and c in the grid points marked with black dots for every time step t_n.

4.1.5 2D explicit finite difference scheme

Grid and initial conditions

To set up a finite difference scheme for the 2D problem, we first extend the grid to two spacial dimensions;

$$\begin{aligned} t_n &= n\Delta t, & n &= 0, 1, ..., \\ x_{s,i} &= -L_s + i\Delta x_s, & i &= 0, 1, ..., N_s, \\ x_{c,i} &= i\Delta x_c, & i &= 0, 1, ..., N_c, \\ y_j &= j\Delta y, & j &= 0, 1, ..., N_y, \end{aligned}$$

where

$$\Delta x_s = \frac{L_s}{N_s}, \qquad \Delta x_c = \frac{L_c}{N_c}, \qquad \Delta y = \frac{L_y}{N_y}.$$

We let $s^n_{i,j}$ be the numerical approximation of $s(t_n, x_{s,i}, y_j)$ and $c^n_{i,j}$ be the numerical approximation of $c(t_n, x_{c,i}, y_j)$.

The initial conditions are discretized:

$$s^0_{i,j} = s_{\text{init}}(x_{s,i}, y_j), \qquad c^0_{i,j} = c_{\text{init}}(x_{c,i}, y_j).$$

Like for the 1D case, the values of $s^n_{i,j}$ and $c^n_{i,j}$ for $n \geq 1$ are calculated using schemes derived by replacing derivatives with finite differences. In the 2D case, we only consider an explicit scheme, and we start by considering the scheme for s, the calcium concentration in the jSR.

General scheme

Replacing the derivatives in the diffusion equation

$$\frac{\partial s}{\partial t} = k_s \left(\frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} \right)$$

with the finite differences (4.3) and (4.6), we get

$$\frac{s_{i,j}^{n+1} - s_{i,j}^n}{\Delta t} = k_s \left(\frac{s_{i+1,j}^n - 2s_{i,j}^n + s_{i-1,j}^n}{\Delta x_s^2} + \frac{s_{i,j+1}^n - 2s_{i,j}^n + s_{i,j-1}^n}{\Delta y^2} \right),$$

and solving for $s_{i,j}^{n+1}$ gives

$$s_{i,j}^{n+1} = r_s^x s_{i-1,j}^n + r_s^y s_{i,j-1}^n + (1 - 2r_s^x - 2r_s^y) s_{i,j}^n + r_s^y s_{i,j+1}^n + r_s^x s_{i+1,j}^n,$$

where

$$r_s^x = \frac{k_s \Delta t}{\Delta x_s^2}, \qquad r_s^y = \frac{k_s \Delta t}{\Delta y^2}.$$

Boundary conditions

The Dirichlet boundary condition is discretized like it was in the 1D case and directly inserted in the scheme for $i = 1$.

The derivatives in the Neumann boundary conditions are replaced by the centred difference (4.5). For instance, the no-flux boundary condition

$$\frac{\partial s}{\partial y}(t, x, 0) = 0$$

yields

$$\frac{s_{i,1}^n - s_{i,-1}^n}{2\Delta y} = 0 \quad \Rightarrow \quad s_{i,-1}^n = s_{i,1}^n.$$

The resulting expressions for the values in the points outside the grid are inserted in the general scheme to get special formulas for the values at the Neumann boundaries.

Full scheme on matrix form

To write the scheme on matrix form, we introduce a solution vector

$$\hat{\mathbf{s}}^n = \left(s_{1,0}^n, \, \cdots, \, s_{1,N_y}^n, \, s_{2,0}^n, \, \cdots, \, s_{2,N_y}^n, \, \cdots, \, \cdots, \, s_{N_s,0}^n, \, \cdots, \, s_{N_s,N_y}^n \right)^T \tag{4.39}$$

containing all the unknown s-values at time t_n.

The scheme can now be written as

$$\hat{\mathbf{s}}^{n+1} = A_s \hat{\mathbf{s}}^n + \mathbf{d}_s^n,$$

where the matrix $A_s \in \mathbb{R}^{N_s(N_y+1)\times N_s(N_y+1)}$ is given by

$$A_s = \left(\begin{array}{ccccccccccccccccc}
\multicolumn{5}{c}{\overbrace{u_s \quad 2r_s^y \quad 0 \quad \cdots \quad 0}^{N_y+1}} & r_s^x & 0 & & & \cdots & & & & \cdots & & & 0 \\
r_s^y & u_s & r_s^y & & & & r_s^x & & & & & & & & & & \\
0 & \ddots & \ddots & \ddots & & & & \ddots & & & & & & & & & \\
\vdots & & r_s^y & u_s & r_s^y & & & & r_s^x & & & & & & & & \vdots \\
0 & & & 2r_s^y & u_s & 0 & & & & r_s^x & & & & & & & \\
r_s^x & & & & 0 & u_s & 2r_s^y & & & & r_s^x & & & & & & \\
0 & r_s^x & & & & r_s^y & u_s & r_s^y & & & & r_s^x & & & & & \\
& & \ddots & & & & \ddots & \ddots & \ddots & & & \ddots & & & & & \vdots \\
& & & r_s^x & & & & r_s^y & u_s & r_s^y & & & & r_s^x & & & \\
\vdots & & & & r_s^x & & & & 2r_s^y & u_s & 0 & & & & r_s^x & & \\
& & & & & \ddots & & & & \ddots & \ddots & \ddots & & & & \ddots & 0 \\
& & & & & & r_s^x & & & & \ddots & \ddots & \ddots & & & & r_s^x \\
& & & & & & & 2r_s^x & & & & 0 & u_s & 2r_s^y & & & 0 \\
\vdots & & & & & & & & 2r_s^x & & & & r_s^y & u_s & r_s^y & & \vdots \\
& & & & & & & & & \ddots & & & & \ddots & \ddots & \ddots & 0 \\
& & & & & & & & & & 2r_s^y & & & & r_s^y & u_s & r_s^y \\
0 & & & \cdots & & & & \cdots & & & 0 & 2r_s^x & 0 & \cdots & 0 & 2r_s^y & u_s
\end{array}\right),$$

the vector $\mathbf{d}_s^n \in \mathbb{R}^{N_s(N_y+1)}$ is given by

$$\mathbf{d}_s^n = \begin{pmatrix} r_s^x h_s(y_0) \\ \vdots \\ r_s^x h_s(y_{N_y}) \\ 0 \\ \vdots \\ 0 \\ \frac{2\Delta t}{\Delta x_s} g(t_n, y_0)\left(c_{0,0}^n - s_{N_s,0}^n\right) \\ \vdots \\ \frac{2\Delta t}{\Delta x_s} g(t_n, y_{N_y})\left(c_{0,N_y}^n - s_{N_s,N_y}^n\right) \end{pmatrix}$$

and $u_s = 1 - 2r_s^x - 2r_s^y$.

A scheme for c, the calcium concentration in the dyad, is derived similarly.

Stability conditions for the explicit 2D finite difference scheme

Like for the 1D case, we can derive stability conditions for the 2D finite difference scheme for the coupled system by collecting all the unknowns for each time step into a single vector and writing the coupled system on matrix form.

Using arguments similar to the 1D case (see page 52-54), we get the stability condition

$$\Delta t \leq \min(\Delta t_1, \Delta t_2), \tag{4.40}$$

where

$$\begin{aligned}\Delta t_1 &= \frac{1}{2}\frac{\Delta x_s^2 \Delta y^2}{k_s(\Delta x_s^2 + \Delta y^2) + \bar{g}\Delta x_s \Delta y^2},\\ \Delta t_2 &= \frac{1}{2}\frac{\Delta x_c^2 \Delta y^2}{k_c(\Delta x_c^2 + \Delta y^2) + \bar{g}\Delta x_c \Delta y^2}\end{aligned}$$

and $\bar{g}$ is the maximum value of $g(t, y)$ for $t \geq 0$ and $y \in [0, L_y]$.

If we let $k_{\max}$ be the maximum of k_s and k_c, and $\Delta x_{\min}$ be the minimum of Δx_s, Δx_c and Δy, the condition (4.40) is satisfied if

$$\Delta t \leq \frac{1}{4}\frac{(\Delta x_{\min})^2}{k_{\max} + \bar{g}\Delta x_{\min}/2}. \tag{4.41}$$

4.1.6 Implementation

The 1D and 2D finite difference schemes are implemented in Matlab. In this section, some of the details of the implementation are presented.

Set up grid in time and space

The 1D grid consists of uniformly spaced points in each of the two domains. In the code, the grid is set up by:

```
% Set up grid in space
Ns = round(N*Ls/(Ls+Lc));      % number of intervals in jSR grid
Nc = round(N*Lc/(Ls+Lc));      % number of intervals in dyad grid
x_s = linspace(-Ls,0,Ns+1)';   % jSR grid
x_c = linspace(0,Lc,Nc+1)';    % dyad grid
```

Here the parameter N denotes the total number of intervals in the spatial grid. The number of intervals in each domain, Ns and Nc, is calculated so that the distance between points is approximately the same in the two domains. The parameters Ls and Lc are the lengths of the jSR and the dyad, respectively.

In 2D, we also set up a vector for the y-values and set up a 2D grid by:

```
y = linspace(0,Ly,Ny+1)';      % y-values of the grid
[xs, ys] = meshgrid(x_s,y);    % 2D jSR grid
[xc, yc] = meshgrid(x_c,y);    % 2D dyad grid
```

The time step parameter, dt, is set up so that it fulfils the stability conditions of the schemes.

Apply initial conditions

In the solvers for 1D problem without buffers, the initial conditions are applied by:

```
% Initial conditions
s = s_init(x_s);
c = c_init(x_c);
```

The functions s_init(x_s) and c_init(x_c) take the grid vectors x_s and x_c as input and return vectors containing the values of the initial conditions in the grid points.

For the implicit solver, the vectors s and c are combined to a single unknown vector u on the form (4.35) by:

```
u = [s(2:Ns+1); c(1:Nc)];    % Combined vector
```

When buffers are included, s and c are extended to store both the concentration of free calcium and the concentration of calcium bound to buffer in every point, and the initial conditions are applied to each of the four concentrations in the same manner as above.

In 2D, the functions for the initial conditions take matrices as input and return the matrices containing the initial conditions in the grid points. The matrices are then transformed to vectors on the form (4.39).

Set up matrices and vectors

For the 1D explicit solver, the matrices A_s and A_c are set up by:

```
% Set up the matrices
As = spdiags((1-2*rs)*ones(Ns,1), 0, Ns, Ns) ...
    + spdiags(rs*ones(Ns,1), 1, Ns, Ns) ...
    + spdiags([rs*ones(Ns-2,1); 2*rs], -1, Ns, Ns);

Ac = spdiags((1-2*rc)*ones(Nc,1), 0, Nc, Nc) ...
    + spdiags([0; 2*rc; rc*ones(Nc-2,1)], 1, Nc, Nc) ...
    + spdiags(rc*ones(Nc-1,1), -1, Nc, Nc);
```

The values rs and rc represents r_s and r_c given by (4.7) and (4.10).

The **d**-vectors are set up by:

```
% Set up the d-vectors
ds = zeros(Ns,1);
dc = zeros(Nc,1);
ds(1)  = rs*s0;
dc(Nc)  = rc*c0;
```

The last element in ds and the first element in dc both vary with time and are updated for every time step.

The matrix A_b for the solver including buffers and the matrices and **d**-vectors for the implicit 1D solver and the explicit 2D solver are set up in a similar manner.

Update solutions for every time step

In the explicit 1D solvers, the **d**-vectors are updated for every time step using the first value in c, the last value in s and a function g(t), which computes the value of the function $g(t)$ given by (2.7). In addition, the solution vectors are updated for every time step. This is repeated until the current time, `t`, reaches the specified time to end the simulation, `Tmax`. The code for the problem without buffers is:

```
t = 0;
while t < Tmax
    % Update d-vector
    ds(Ns) = 2*dt/dx_s*g(t)*(c(1)-s(Ns+1));
    dc(1) = -2*dt/dx_c*g(t)*(c(1)-s(Ns+1));

    % Calculate s and c for next time step
    s(2:Ns+1) = As*s(2:Ns+1) + ds;
    c(1:Nc) = Ac*c(1:Nc) + dc;

    % Update t
    t = t + dt;
end
```

In the 1D explicit solver including buffers and the 2D explicit solver, we use similar loops to calculate new solutions for every time step.

In the implicit 1D solver, we need to solve a system of equations for each time step. In addition, some of the values in the matrix A change with time and are updated for every time step. This is done in the loop:

```
t = 0;
while t < Tmax
    % Update t
    t = t + dt;

    % Update values in the matrix
    A(Ns, Ns) = 1 + 2*rs + 2*dt/dx_s*g(t);
    A(Ns, Ns+1) = -2*dt/dx_s*g(t);
    A(Ns+1, Ns+1) = 1 + 2*rc + 2*dt/dx_c*g(t);
    A(Ns+1, Ns) = -2*dt/dx_c*g(t);

    % Solve linear system of equations
    u = A\(u+d);
end

% Rearrange solution to vectors s and c
s = [s0; u(1:Ns)];
c = [u(Ns+1:Ns+Nc); c0];
```

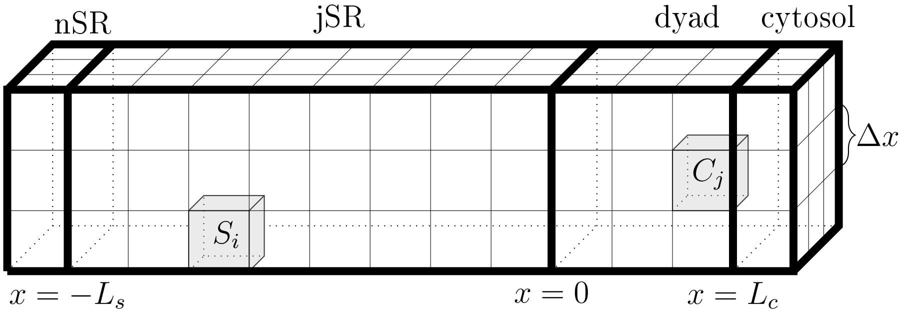

Figure 4.3: Illustration of the spatial 3D grid used in the finite volume method. The grid consists of a number of non-overlapping cube-shaped cells with sides of length Δx. Some arbitrarily chosen cells S_i and C_j are marked in the illustration. The calcium concentration is assumed to be piecewise constant for each cell for each time step. The cells in the region where $x \leq -L_s$ represent the nSR. In these cells the concentration is s_0. The cells in the region where $x \geq L_c$ represent the cytosol, and in these cells the concentration is c_0.

4.2 Finite volume method

For the 3D simulations, a code developed for the paper [3] was used. The code is an implementation of a finite volume method, and the finite volume scheme used to solve the 3D problem described in Section 2.1.5 is described below.

4.2.1 3D finite volume scheme

In the finite volume method, the domain is divided into a finite number of cells and conservation of mass is enforced in each of the cells [12, 27].

To define a finite volume scheme for the 3D problem in Section 2.1.5, we divide the domains Ω_s and Ω_c into non-overlapping cube-shaped cells with sides of length Δx. An illustration of the grid is given in Figure 4.3.

The cells in the jSR domain are denoted by

$$S_i, \quad i = 1, ..., N_s N_y N_z,$$

and the cells in the dyad domain are denoted by

$$C_i, \quad i = 1, ..., N_c N_y N_z,$$

where

$$N_s = \frac{L_s}{\Delta x}, \quad N_c = \frac{L_c}{\Delta x}, \quad N_y = \frac{L_y}{\Delta x}, \quad N_z = \frac{L_z}{\Delta x}.$$

The concentration is assumed to be piecewise constant in the sense that

$$\begin{aligned} s(t,x,y,z) &= s_i(t) \quad \text{for } (x,y,z) \in S_i, \\ c(t,x,y,z) &= c_i(t) \quad \text{for } (x,y,z) \in C_i, \end{aligned}$$

and we seek the solutions

$$\begin{aligned} &s_i, \quad i = 1, ..., N_s N_y N_z \\ &c_i, \quad i = 1, ..., N_c N_y N_z, \end{aligned}$$

for each discrete time step.

To the left of the jSR domain, there are cells representing the nSR. In these cells, we force the concentration to be s_0. Similarly, there are cells to the right of the dyad domain, representing the cytosol, where the concentration is forced to be c_0.

To derive a scheme for the diffusion equations (2.4) and (2.5), we enforce conservation of mass directly on the cells. Recall from Section 2.1.1 that conservation of mass in a region Ω of the dyad domain can be written as

$$\frac{d}{dt}\int_\Omega c\, dV = -\int_{\partial\Omega} \mathbf{J}\cdot\mathbf{n}\, dA$$

if we assume that there is no local production of calcium, e.g. reactions with buffers. Here c is the calcium concentration, $\partial\Omega$ is the boundary of Ω, $\mathbf{J}$ is the flux of calcium and $\mathbf{n}$ is the outward unit normal vector.

Enforcing conservation of mass on a cell C_i in the dyad domain gives

$$\frac{d}{dt}\int_{C_i} c\, dV = -\int_{\partial C_i} \mathbf{J}\cdot\mathbf{n}\, dA,$$

where ∂C_i is the boundary of cell C_i.

Since we assume that the calcium concentration is constant in each cell, we can move the concentration out of the integral and get

$$\int_{C_i} c\, dV = \int_{C_i} c_i\, dV = c_i \int_{C_i} 1\, dV = c_i \Delta x^3,$$

since the volume of a cell is Δx^3.

This gives:

$$\frac{dc_i}{dt} = -\frac{1}{\Delta x^3}\int_{\partial C_i} \mathbf{J}\cdot\mathbf{n}\,dA.$$

Since each cell has six boundary faces, we can write

$$\frac{dc_i}{dt} = -\frac{1}{\Delta x^3}\sum_{j=1}^{6}\int_{F_{ij}} \mathbf{J}\cdot\mathbf{n}_{ij}\,dA = \sum_{j=1}^{6} D_{ij}, \tag{4.42}$$

where $F_{ij},\ j = 1, ..., 6$ denotes each of the boundary faces of cell C_i, $\mathbf{n}_{ij}$ denotes the outward unit normal vector of F_{ij} and

$$D_{ij} = -\frac{1}{\Delta x^3}\int_{F_{ij}} \mathbf{J}\cdot\mathbf{n}_{ij}\,dA. \tag{4.43}$$

Conservation of mass on a cell S_i in the jSR domain similarly gives

$$\frac{ds_i}{dt} = \sum_{j=1}^{6} E_{ij}, \tag{4.44}$$

where

$$E_{ij} = -\frac{1}{\Delta x^3}\int_{F_{ij}} \mathbf{J}\cdot\mathbf{n}_{ij}\,dA.$$

and F_{ij}, $j = 1, ..., 6$ is each of the boundary faces of S_i.

In our 3D problem, there are four types of boundary faces:

1. Internal boundaries

2. Boundaries to cells with known concentration (nSR, cytosol)

3. No-flux boundaries

4. Boundaries covered by the RyR channel

We will now explain how the contributions D_{ij} and E_{ij} are expressed for each of these boundary types.

1. Internal boundary

Assuming that the boundary face F_{ij} is an internal boundary between the cells C_i and C_k within the dyad domain, the flux through the boundary is governed by Fick's law; $\mathbf{J} = -k_c \nabla c$ (see Section 2.1.1). We get:

$$\mathbf{J} \cdot \mathbf{n}_{ij} = -k_c \nabla c \cdot \mathbf{n}_{ij}.$$

We use the centred finite difference (4.5) with $h = \Delta x/2$ to approximate the derivative on the boundary and get:

$$\nabla c \cdot \mathbf{n}_{ij} = \frac{c_k - c_i}{\Delta x}.$$

Inserting this in (4.43) yields:

$$\begin{aligned} D_{ij} &= -\frac{1}{\Delta x^3} \int_{F_{ij}} \mathbf{J} \cdot \mathbf{n}_{ij} \, dA = -\frac{1}{\Delta x^3} \int_{F_{ij}} -k_c \frac{c_k - c_i}{\Delta x} \, dA = k_c \frac{c_k - c_i}{\Delta x^4} \int_{F_{ij}} 1 \, dA \\ &= k_c \frac{c_k - c_i}{\Delta x^2}. \end{aligned}$$

For an internal boundary face F_{ij} between the cells S_i and S_k in the jSR domain, we similarly have

$$E_{ij} = k_s \frac{s_k - s_i}{\Delta x^2}.$$

2. Boundary to cell with known concentration

For a boundary face F_{ij} between a cell in the dyad domain and a cell in the cytosol domain, the flux through the boundary is again assumed to be governed by Fick's law, and we approximate the derivative with the finite difference (4.5). We get exactly the same expression as we did for internal boundaries within the dyad domain, except that the concentration c_k is known to be c_0, so

$$D_{ij} = k_c \frac{c_0 - c_i}{\Delta x^2}.$$

Similarly, we have

$$E_{ij} = k_s \frac{s_0 - s_i}{\Delta x^2}$$

for boundary faces F_{ij} between cells in the jSR and nSR domains.

3. No-flux boundary

On a boundary face with a no-flux boundary condition, we have $\mathbf{J} \cdot \mathbf{n}_{ij} = 0$, so

$$D_{ij} = 0,$$

$$E_{ij} = 0.$$

4. Boundary covered by the RyR channel

To find the contribution from a boundary face covered by the RyR channel, we recall from Section 2.1.2 that on the part of the boundary between the jSR and the dyad that is covered by the channel, we have

$$\mathbf{J} \cdot \mathbf{n} = g(t)(s - c) \quad \text{for the jSR},$$

$$\mathbf{J} \cdot \mathbf{n} = g(t)(c - s) \quad \text{for the dyad.}$$

Considering a cell C_i with a boundary face F_{ij} against the cell S_k , we get

$$\begin{aligned} D_{ij} &= -\frac{1}{\Delta x^3} \int_{F_{ij}} \mathbf{J} \cdot \mathbf{n}_{ij} \, dA = -\frac{1}{\Delta x^3} \int_{F_{ij}} g(t)(c_i - s_k) \, dA \\ &= -g(t) \frac{c_i - s_k}{\Delta x^3} \int_{F_{ij}} 1 \, dA = \frac{1}{\Delta x} g(t)(s_k - c_i), \end{aligned}$$

assuming that the channel covers the entire boundary face F_{ij}.

For a cell S_i with a boundary face F_{ij} against the cell C_k, we similarly get:

$$E_{ij} = -\frac{1}{\Delta x} g(t)(s_i - c_k).$$

Final scheme

To get the final scheme for the solution in the cells, the contributions from the different boundary faces are added together in the sums in (4.42) and (4.44). The time derivative in the resulting equations is discretized using the forward finite difference (4.3), and we get an explicit numerical scheme.

Chapter 5

Numerical simulations

In this chapter, we present results from numerical simulations of the mathematical models described in Chapter 2. To study the accuracy of the simulations for different grid resolutions, errors of the numerical approximations are computed for different resolutions and collected in convergence tables. In addition, plots of the numerical and analytical solutions are presented to get a visual impression of how the solutions behave and how close the numerical solutions are to the analytical ones.

Results from the following simulations will be presented:

(I) First, we consider simulations of the 1D problem where $\bar{g}(t)$ is constructed as described in Section 3.1.1. In this case, we have analytical solutions both for when the channel is open and for when the channel is closed. The solutions produced by the numerical simulations for different grid resolutions are compared to these analytical solutions.

(II) We then present results from simulations of the 1D problem with a constant $\bar{g}$ and constant initial conditions. In this case, we only have analytical solutions for steady state and for when the channel is closed after steady state is reached, but the results are more in agreement with the original mathematical model.

(III) Next, we present results from simulations of the 1D problem with a constant $\bar{g}$ including buffers. We do not have analytical solutions in this case, so we just use plots of simulations for different resolutions to see if any difference is visible.

(IV) Afterwards, we present results from simulations of the 2D problem with a constructed $\bar{g}(t, y)$. In this case, we have analytical solutions both for when the channel is open and for when the channel is closed.

(V) Finally, we present results from simulations of the 3D problem with a constant $\bar{g}$. In this case, we do not have any analytical solutions, but to get an impression of the accuracy of the simulations for different grid resolutions, we compare the numerical solutions for coarse grids with the solutions produced by simulations on finer grids.

5.1 Numerical codes

To run the 1D simulations, we use Matlab implementations of the finite difference schemes described in Section 4.1. For the 1D problem without buffers, we do simulations using both an explicit and an implicit solver, and for the problem with buffers, we only use an explicit solver.

Similarly, the 2D simulations are carried out using a Matlab implementation of the explicit finite difference scheme described in Section 4.1.5. The Matlab implementations used in the 1D and 2D simulations are written for this thesis, and some of the details of the implementation are given in Section 4.1.6.

For the 3D simulations, we use a code developed for the paper [3] by the authors of that paper. The code is an implementation of a finite volume method described in Section 4.2.

5.2 Simulations of the 1D problem with a constructed $\bar{g}(t)$

We start by presenting results from simulations of the 1D problem described in Section 2.1.3, were the flux from the jSR to the dyad is constructed to fit analytical solutions as explained in Section 3.1.1. This means that $\bar{g}(t)$ is given by the expression (3.10) found on page 24. The initial conditions are also adjusted to fit the analytical solutions and are given by (3.11) and (3.12).

The parameters used in the simulations are found in Table 2.1 and Table 3.1.

5.2.1 Results case I

In Figure 5.1, we show the analytical and numerical solutions in the jSR domain at some different points in time. Figure 5.2 shows the corresponding solutions in the dyad domain. In Figure 5.3, we show more detailed plots of the solutions at $t = 0.201$ ms, where some difference between the analytical and numerical solutions is visible. To get an overview of the whole domain, the solutions from the jSR and the dyad at $t = 2.0$ ms are plotted together in Figure 5.4. Figure 5.5 shows how the concentration in the midpoint of each domain change with time.

In the simulations, the RyR channel is open until $t = t^* = 2.0$ ms and then closed. By considering Figure 5.1, Figure 5.2 and Figure 5.5, we observe that when the channel is open, the calcium concentration in the dyad increases and the calcium concentration in the jSR decreases. This is reasonable since there should be calcium flowing out of the jSR into the dyad when the channel is open. The concentration changes most extensively close to the boundary between the domains, at $x = 0$, where the channel is located and remains constant at the Dirichlet boundaries $x = -L_s$ and $x = L_c$.

When the channel is closed, the concentration in the dyad decreases and the concentration in the jSR increases until the constant solutions c_0 and s_0 are reached. This is also reasonable; since there is no longer any connection between the two domains, we would expect diffusion to even out the concentration differences in each domain. Calcium is diffusing out of the dyad into the bulk cytosol and from the nSR to the jSR.

For the simulations recorded in the plots, we use $\Delta x_s = \Delta x_c = 15$ nm. We observe that the numerical solutions are very close to the analytical ones, except at $t = 2.001$ ms, where we can see a small difference between the numerical and analytical solutions in the dyad domain (see Figure 5.3).

5.2.2 Convergence tables

The errors of the numerical simulations of the 1D problem are collected in convergence tables. In these tables, we compare the numerical approximations of the solutions at some point in time to the analytical solutions.

In Table 5.1 and Table 5.2, the numerical solutions at time $t = 0.001$ ms are compared to the analytical solutions at that point in time. In Table 5.3 and Table 5.4, we consider the solutions at $t = 2.001$ ms, i.e. 0.001 ms after the channel is closed.

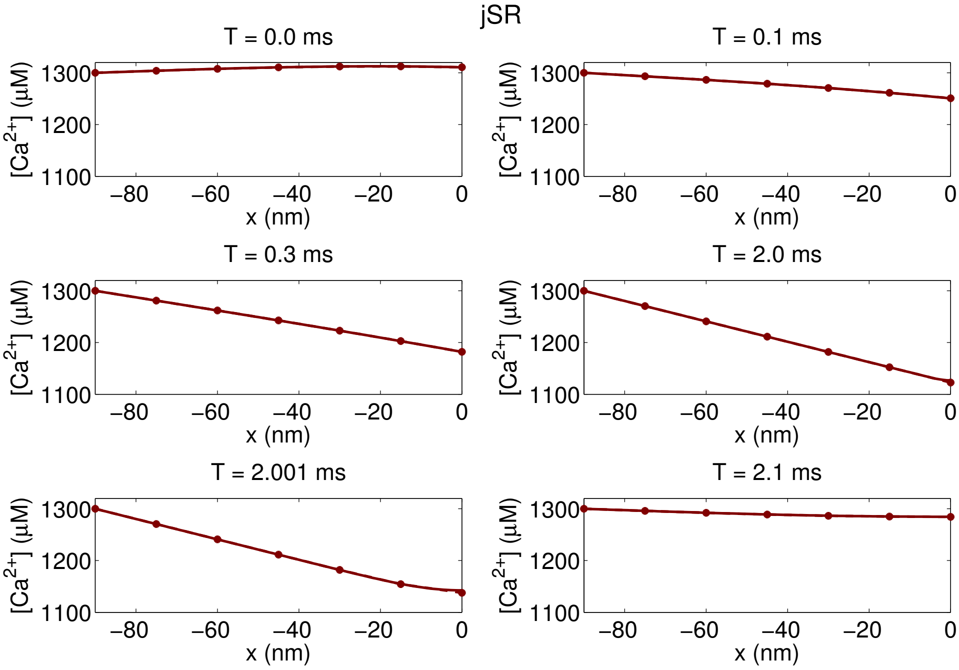

Figure 5.1: Analytical and numerical solutions of the 1D problem with a constructed $\bar{g}(t)$ in the jSR at some different points in time. The analytical solutions are plotted with a solid line, and the numerical solutions are plotted as dots. The numerical solutions are generated using the explicit solver with $\Delta x = 15$ nm.

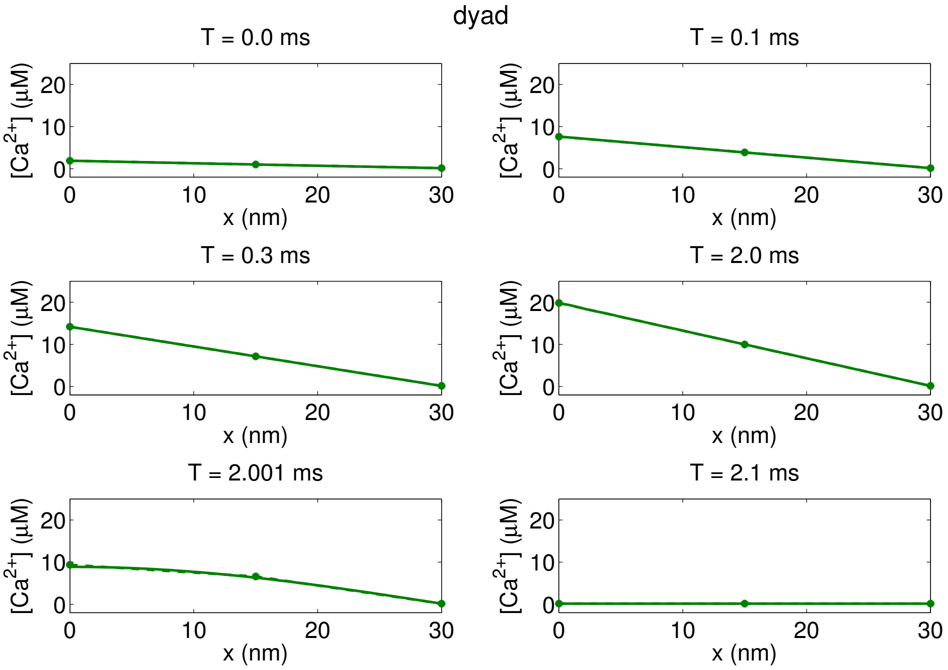

Figure 5.2: Analytical and numerical solutions of the 1D problem with a constructed $\bar{g}(t)$ in the dyad at some different points in time. The analytical solutions are plotted with a solid line, and the numerical solutions are plotted as dots. The numerical solutions are generated using the explicit solver with $\Delta x = 15$ nm.

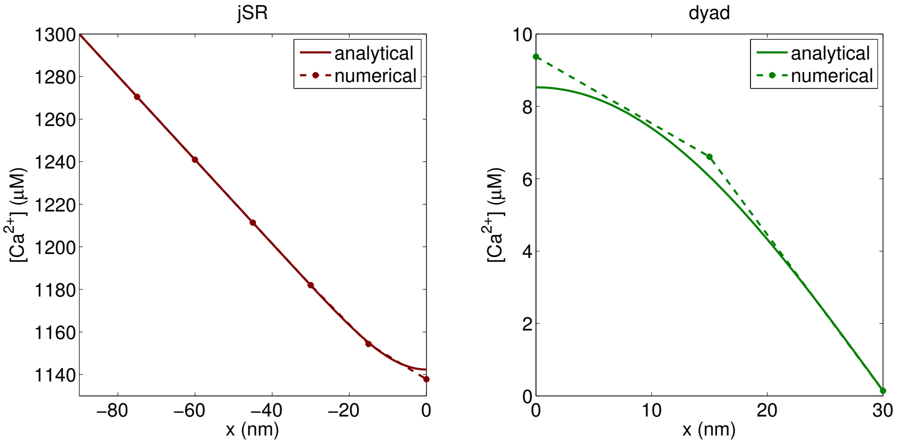

Figure 5.3: Analytical and numerical solutions of the 1D problem with a constructed $\bar{g}(t)$ at $t = 2.001$ ms. The numerical solutions are generated using the explicit solver with $\Delta x = 15$ nm.

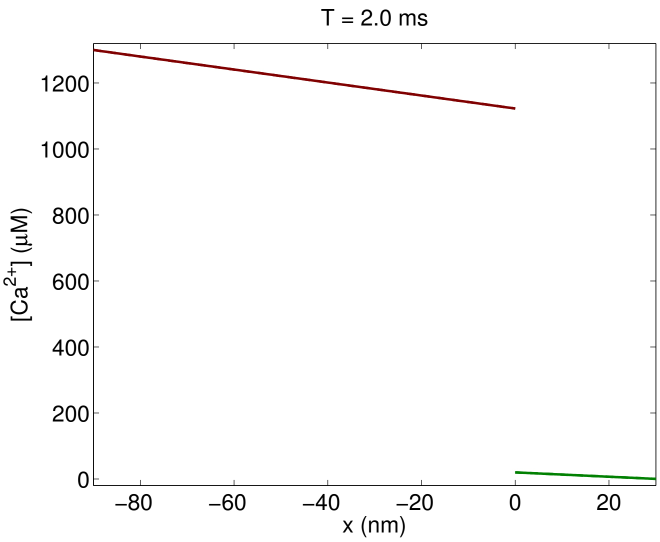

Figure 5.4: Analytical and numerical solutions of the 1D problem in both domains at $t = 2.0$ ms. The numerical solution for $\Delta x = 15$ nm is plotted as a dashed line, but is indistinguishable from the analytical one.

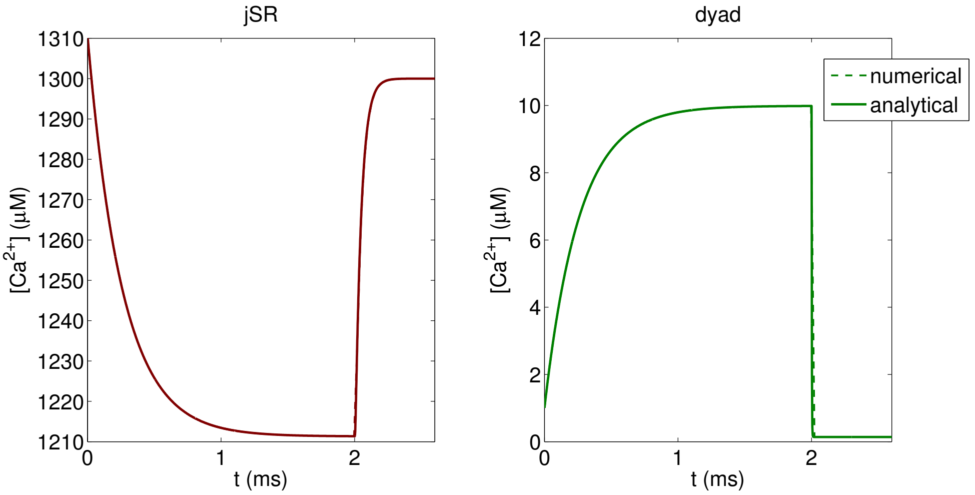

Figure 5.5: Plots of how the concentration in the midpoint of the domains change with time for the 1D problem with a constructed $\bar{g}(t)$. The numerical solutions are generated using the explicit solver with $\Delta x = 15$ nm.

Δx (nm)	Error	Error/Δx^2 (nm^{-2})
15	0.0029183	1.30e-05
10	0.0012787	1.28e-05
5	0.00031204	1.25e-05
3	0.00011072	1.23e-05
2	4.8683e-05	1.22e-05
1	1.2046e-05	1.20e-05

Table 5.1: Comparing the numerical and analytical solutions of the 1D problem with a constructed $\bar{g}(t)$ at $t = 0.001$ ms for the explicit solver. Error is calculated using (5.1) with the norm (5.2). The parameter values are found in Table 2.1 and Table 3.1.

In the convergence tables, the error is defined as

$$\text{Error} = \frac{\|\mathbf{s} - \mathbf{s}_a\|}{\|\mathbf{s}_a\|} + \frac{\|\mathbf{c} - \mathbf{c}_a\|}{\|\mathbf{c}_a\|}, \tag{5.1}$$

where $\mathbf{s}$ and $\mathbf{c}$ are the numerical solutions at the considered point in time and $\mathbf{s}_a$ and $\mathbf{c}_a$ are vectors containing the corresponding analytical solutions in the grid points.

The norm is defined as

$$\|\mathbf{u}\| = \sqrt{\sum_{i=0}^{N} (u_i)^2}, \tag{5.2}$$

where u_i, $i = 0, ...N$ are each of the entries in the vector $\mathbf{u}$.

In all the simulations, the values of N_s and N_c are chosen so that

$$\Delta x_s = \Delta x_c = \Delta x.$$

For both the implicit and the explicit solver, we use $\Delta t = \frac{0.1}{k_c}\Delta x^2$. Studying the last column of the convergence tables, the error seems to be close to proportional to Δx^2 and Δt. This fits well with the fact that we introduced errors of order $\mathcal{O}(\Delta t)$ and $\mathcal{O}(\Delta x^2)$ when we replaced the time derivative with the finite differences (4.3) or (4.4) and the derivative in space with the finite difference (4.6).

Δx (nm)	Error	Error/Δx^2 (nm^{-2})
15	0.0028054	1.25e-05
10	0.0012604	1.26e-05
5	0.0003117	1.25e-05
3	0.00011092	1.23e-05
2	4.8818e-05	1.22e-05
1	1.2086e-05	1.21e-05

Table 5.2: Comparing the numerical and analytical solutions of the 1D problem with a constructed $\bar{g}(t)$ at $t = 0.001$ ms for the implicit solver. Error is calculated using (5.1) with the norm (5.2). The parameter values are found in Table 2.1 and Table 3.1.

Δx (nm)	Error	Error/Δx^2 (nm^{-2})
15	0.097904	4.35e-04
10	0.057353	5.74e-04
5	0.0071173	2.85e-04
3	0.0028203	3.13e-04
2	0.0011303	2.83e-04
1	0.00028217	2.82e-04

Table 5.3: Comparing the numerical and analytical solutions of the 1D problem with a constructed $\bar{g}(t)$ at $t = 2.001$ ms for the explicit solver. Error is calculated using (5.1) with the norm (5.2). The parameter values are found in Table 2.1 and Table 3.1.

Δx (nm)	Error	Error/Δx^2 (nm^{-2})
15	0.077289	3.44e-04
10	0.046897	4.69e-04
5	0.0044931	1.80e-04
3	0.0018702	2.08e-04
2	0.00070583	1.76e-04
1	0.00017573	1.76e-04

Table 5.4: Comparing the numerical and analytical solutions of the 1D problem with a constructed $\bar{g}(t)$ at $t = 2.001$ ms for the implicit solver. Error is calculated using (5.1) with norm (5.2). The parameter values are found in Table 2.1 and Table 3.1.

5.3 Simulations of the 1D problem with a constant $\bar{g}$

In the simulations presented above, we adjusted the flux from the jSR to the dyad to fit analytical solutions by using a special $\bar{g}(t)$. Due to this, the results might not be entirely in agreement with the results of the original mathematical model in which $\bar{g}$ was a specified constant. Therefore, we will also present results of simulations of the original mathematical model.

In Section 3.1.1, we showed that it was possible to adjust the parameters of the constructed analytical solutions for an open channel to fit a specified constant $\bar{g}$, but that the resulting analytical solutions were independent of t, would be specified as initial conditions and were consequently not so suitable for testing the accuracy of the numerical simulations.

However, we also argued that we could expect the solutions also for other initial conditions to approach these analytical steady-state solutions as $t \to \infty$. This will be confirmed by the simulations in this section. We use a constant $\bar{g}$ and the constant initial conditions (2.16) and observe that the solutions approach the analytical steady-state solutions derived in Section 3.1.1. We assume that the solutions have reached steady state when we close the channel, so the solutions described on page 34 are used as analytical solutions for after the channel is closed.

The parameter values used in the simulations are specified in Table 2.1.

5.3.1 Results case II

In Figure 5.6 and Figure 5.7, the numerical solutions at some different points in time are plotted along with the expected steady-state solutions. We observe that as time increases, the solutions approach the steady-state solutions and that the steady-state solutions are reached quite quickly.

Figure 5.8 shows how the concentration in the midpoint of each domain change with time. The analytical solution of the problem with the constructed flux are plotted in addition to the numerical solution of the problem with a constant $\bar{g}$. We see that the concentration in the dyad increases and the concentration in the jSR decreases much faster for the constant $\bar{g}$ than for the constructed $\bar{g}(t)$. In addition, the concentration in the dyad reaches an initial peak before settling at the steady-state solution. This behaviour is not displayed for the constructed $\bar{g}(t)$.

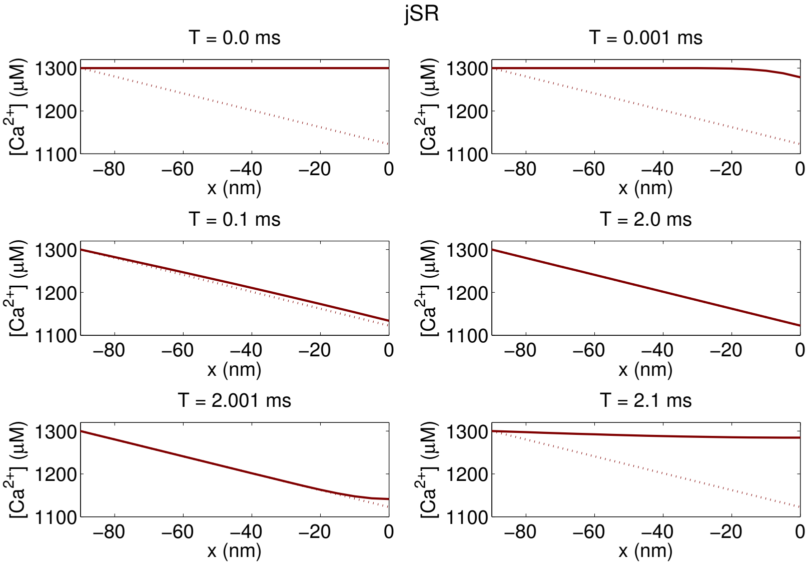

Figure 5.6: Solutions of the 1D problem with a constant $\bar{g}$ and constant initial conditions in the jSR. The solid line is the numerical solution for $\Delta x = 5$ nm, and the dotted line is the analytical steady-state solution.

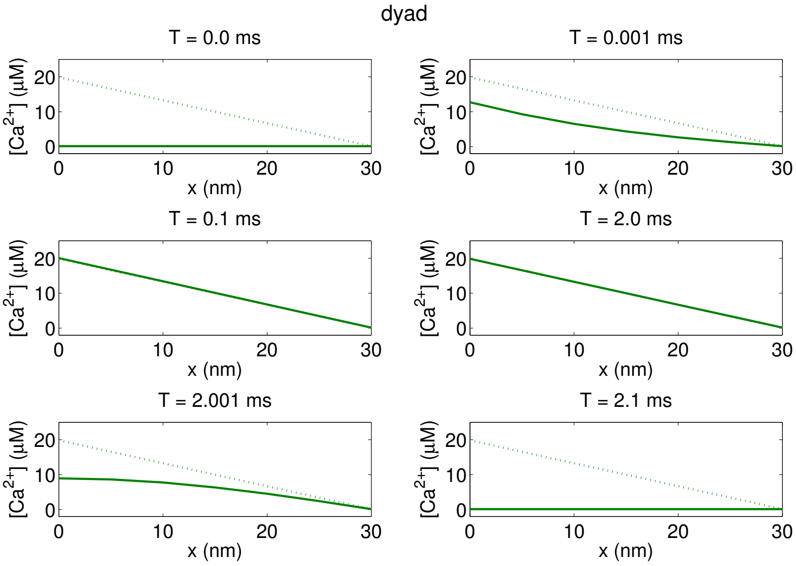

Figure 5.7: Solutions of the 1D problem with a constant $\bar{g}$ and constant initial conditions in the dyad. The solid line is the numerical solution for $\Delta x = 5$ nm, and the dotted line is the analytical steady-state solution.

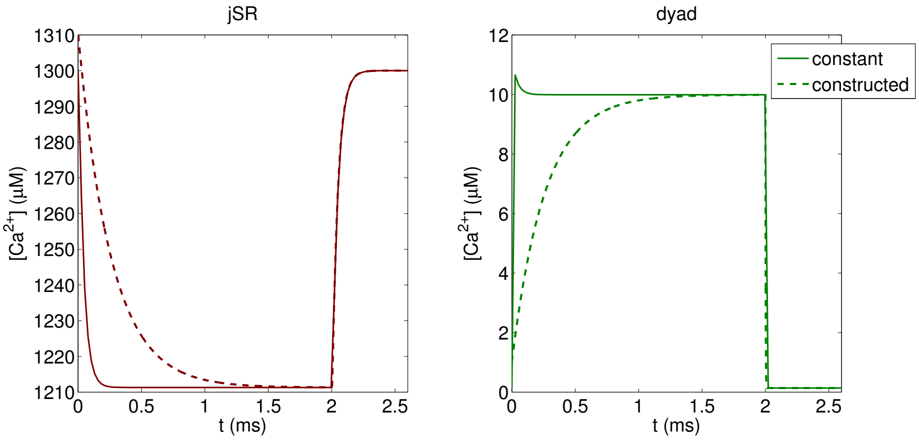

Figure 5.8: Plots of how the concentration in the midpoint of the domains change with time for the 1D problem with a constant $\bar{g}$ and constant initial conditions. The numerical solutions for the explicit solver with $\Delta x = 5$ nm are plotted with a solid line. In addition, the analytical solutions for the 1D problem with a constructed $\bar{g}(t)$ are plotted with a dashed line.

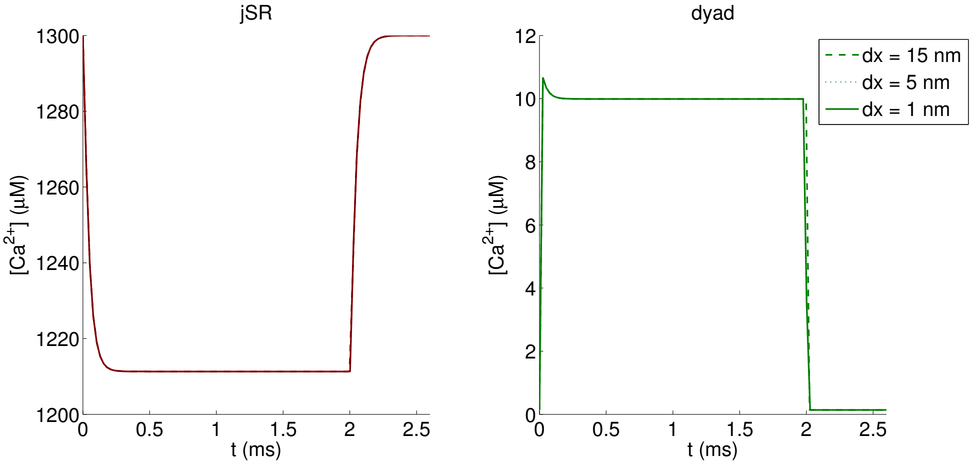

Figure 5.9: Plots of how the numerical approximation of the concentration in the midpoint of each domain change with time for the 1D problem with a constant $\bar{g}$ and constant initial conditions for some different values of Δx.

To see if there is any clear difference in accuracy for different resolutions in this case, the numerical solution in the midpoints are plotted for different grid resolutions in Figure 5.9. The solutions for different resolutions overlap and no clear difference is visible except for a small difference in the dyad domain right after the channel is closed.

5.3.2 Convergence tables

In the case of a constant $\bar{g}$ and constant initial conditions, we do not have analytical solutions for an open channel, except that we expect the solutions to approach the steady-state solutions (3.17) and (3.18) as $t \to \infty$. Studying Figure 5.6 and Figure 5.7, the solutions seem to reach this steady state quite quickly.

In Table 5.5, we compare the numerical solutions at time $t = 1.0$ ms for different resolutions to the analytical steady-state solutions. The error is calculated like before using (5.1) with the norm (5.2).

We again choose N_s and N_c so that

$$\Delta x_s = \Delta x_c = \Delta x$$

and use $\Delta t = \frac{0.1}{k_c} \Delta x^2$.

Considering Table 5.5, we observe that the difference between the analytical steady-state solutions and the numerical solutions at $t = 1.0$ ms is very small for all values of Δx and that the accuracy is not improved when we reduce the value of Δx. In fact, the error is larger for small values of Δx than for large values of Δx. This suggests that the numerical schemes are able to reproduce the linear steady-state solutions to machine precision independently of Δx.

After steady state is reached, we have analytical solutions for after the channel is closed. In Table 5.6 and Table 5.7, numerical solutions at time $t = 2.001$ ms are compared to these analytical solutions when we close the channel at $t = t^* = 2.0$ ms. The errors are very close to the ones reported for the closed channel when the flux was constructed. This is reasonable since the steady-state solutions are the same for a constant $\bar{g}$ and the constructed $\bar{g}(t)$ when we have chosen the value of a_4 like explained on page 27.

Δx (nm)	Error
15	1.1711e-12
10	1.1267e-12
5	1.2051e-12
3	1.6668e-12
2	3.6178e-12
1	1.4726e-11

Δx (nm)	Error
15	1.2386e-12
10	1.1385e-12
5	1.1042e-12
3	1.3135e-12
2	3.0439e-12
1	1.2187e-11

Table 5.5: Convergence tables where the numerical solutions of the 1D problem with a constant $\bar{g}$ at $t = 1.0$ ms are compared to the analytical steady state solutions. The numerical solutions are generated from the explicit solver (left) and implicit solver (right). Error is calculated using (5.1) with the norm (5.2). The parameter values are found in Table 2.1.

Δx (nm)	Error	Error/Δx^2 (nm^{-2})
15	0.097905	4.35e-04
10	0.057354	5.74e-04
5	0.0071174	2.85e-04
3	0.0028203	3.13e-04
2	0.0011304	2.83e-04
1	0.00028217	2.82e-04

Table 5.6: Comparing the analytical and numerical solutions of the 1D problem with a constant $\bar{g}$ at $t = 2.001$ ms for the explicit solver. Error is calculated using (5.1) with the norm (5.2). The parameter values are found in Table 2.1.

Δx (nm)	Error	Error/Δx^2 (nm^{-2})
15	0.077291	3.44e-04
10	0.046897	4.69e-04
5	0.0044932	1.80e-04
3	0.0018702	2.08e-04
2	0.00070584	1.76e-04
1	0.00017574	1.76e-04

Table 5.7: Comparing the analytical and numerical solutions of the 1D problem with a constant $\bar{g}$ at $t = 2.001$ ms for the implicit solver. Error is calculated using (5.1) with the norm (5.2). The parameter values are found in Table 2.1.

5.4 Simulations of the 1D problem including buffers

We have also run simulations of the 1D problem including reactions with buffers to study how including these reactions would affect the results of the simulations.

A buffer called calsequestrin (CSQN) is introduced in the jSR, and a buffer called calmodulin (CMDN) is introduced in the dyad. The formulation of the problem is found in Section 2.2.2, and the parameter values are found in Table 2.1 and Table 2.3.

5.4.1 Results case III

Figure 5.10 shows the numerical approximations of the calcium concentrations and the concentration of calcium bound to buffers at $t = 0.1$ ms. In addition, the calcium concentration at the same point in time is plotted for the case when there are no buffers present. We see that the calcium concentration in the jSR is much lower for the solution without buffers, suggesting that more calcium has been released from the jSR to the dyad. As there is less calcium in the jSR, less calcium is released through the RyR channel and there is slightly less calcium in the dyad.

Figure 5.11 shows how the concentration in the midpoint of the domains change with time. Again, we also include plots of the calcium concentration without buffers present. We see that the release of calcium from the jSR is slower when buffers are included and that the initial peak in calcium concentration in the dyad is slightly higher. The concentration of calcium bound to CSQN seems to follow the concentration of free calcium while the concentration of calcium bound to CMDN increases even after the concentration of free calcium has reached steady state.

To see if there is any visible difference of accuracy for different resolutions in this case, the calcium concentration in the midpoint of the domains are plotted for some different values of Δx in Figure 5.12. We again observe that the lines overlap and no difference between the solutions for different resolutions is visible.

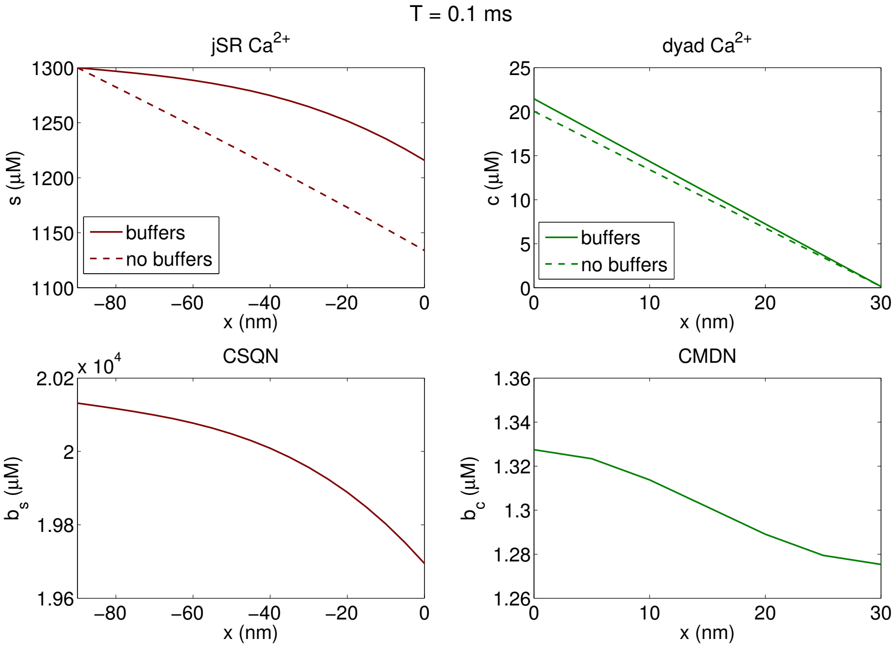

Figure 5.10: Numerical solutions of the concentration of free calcium and calcium bound to buffer in the jSR and dyad domains at $t = 0.1$ ms. The parameters are found in Table 2.1 and Table 2.3, and $\Delta x = 5$ nm. The dashed lines are the corresponding solutions when no buffers are present.

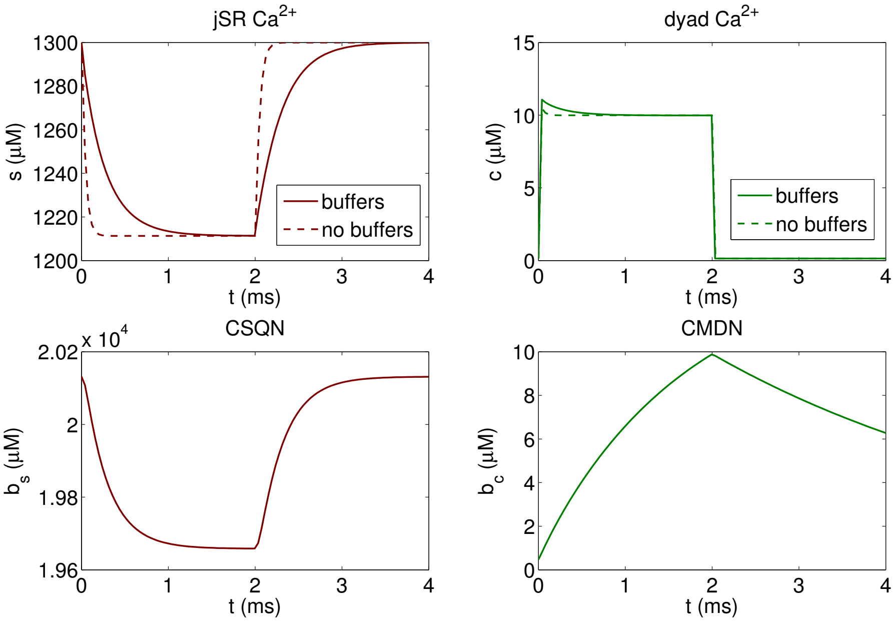

Figure 5.11: Plots of how the calcium concentration and the concentration of calcium bound to buffers in the midpoint of the domains change with time. The parameters are found in Table 2.1 and Table 2.3, and $\Delta x = 5$ nm. The dashed lines are the corresponding solutions when no buffers are present.

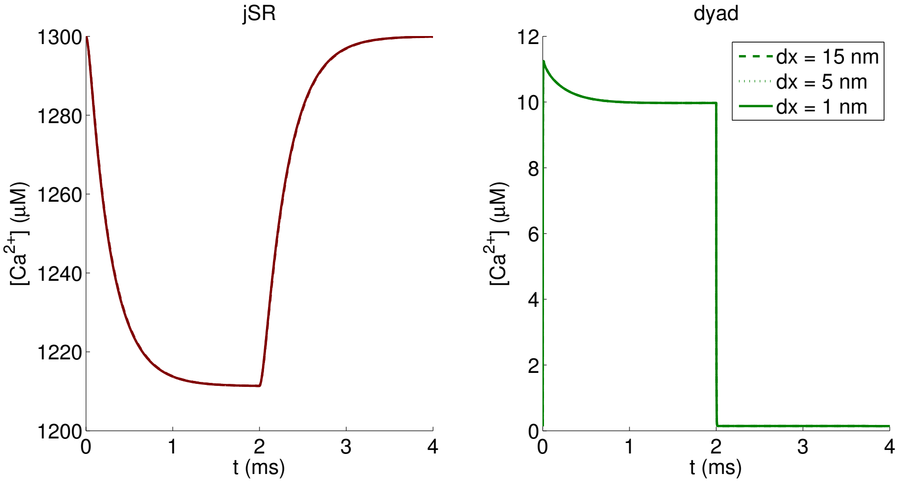

Figure 5.12: Plots of how the numerical approximations of the calcium concentration in the midpoints of each domain change with time for the 1D problem including buffers for different values of Δx.

5.5 Simulations of the 2D problem

In this section, we present results from numerical simulations of the 2D problem described in Section 2.1.4. To find analytical solutions of the system, $\bar{g}(t, y)$ is constructed to be a function of t and y and the initial conditions are functions of x and y. The expressions for the function $\bar{g}(t, y)$ and the initial conditions are found on page 37.

In all the simulations, the discretization parameters are chosen so that

$$\Delta x_s = \Delta x_c = \Delta y = \Delta x,$$

and the parameter values are found in Table 2.1, Table 3.1 and Table 3.3.

5.5.1 Results case IV

Figure 5.13 and Figure 5.14 shows the analytical solutions of the 2D problem in the jSR and the dyad at some different points in time. The solutions behave like they did in the 1D case with a constructed $\bar{g}(t)$, except for the special Dirichlet boundary conditions creating some variation in the y-direction, especially in the regions close to the Dirichlet boundaries.

Figure 5.15 shows how the concentration in the midpoint of each of the domains change with time for some different resolutions. In this case, we observe some clear differences between the analytical and numerical solutions. In particular, the numerical solutions in the dyad for the 15 nm and 7.5 nm resolutions are visibly different than the analytical solution.

5.5.2 Convergence tables

Like for the 1D case, the errors of the numerical simulations in 2D are collected in convergence tables for some points in time. In Table 5.8, we consider the solutions at $t = 0.001$ ms, and in Table 5.9, we consider the solutions at $t = 2.001$ ms.

The error is again calculated from (5.1) with the norm (5.2), and we use $\Delta t = \frac{0.1}{k_c}\Delta x^2$. We observe that the error also for this case seems to be close to proportional to Δt and Δx^2.

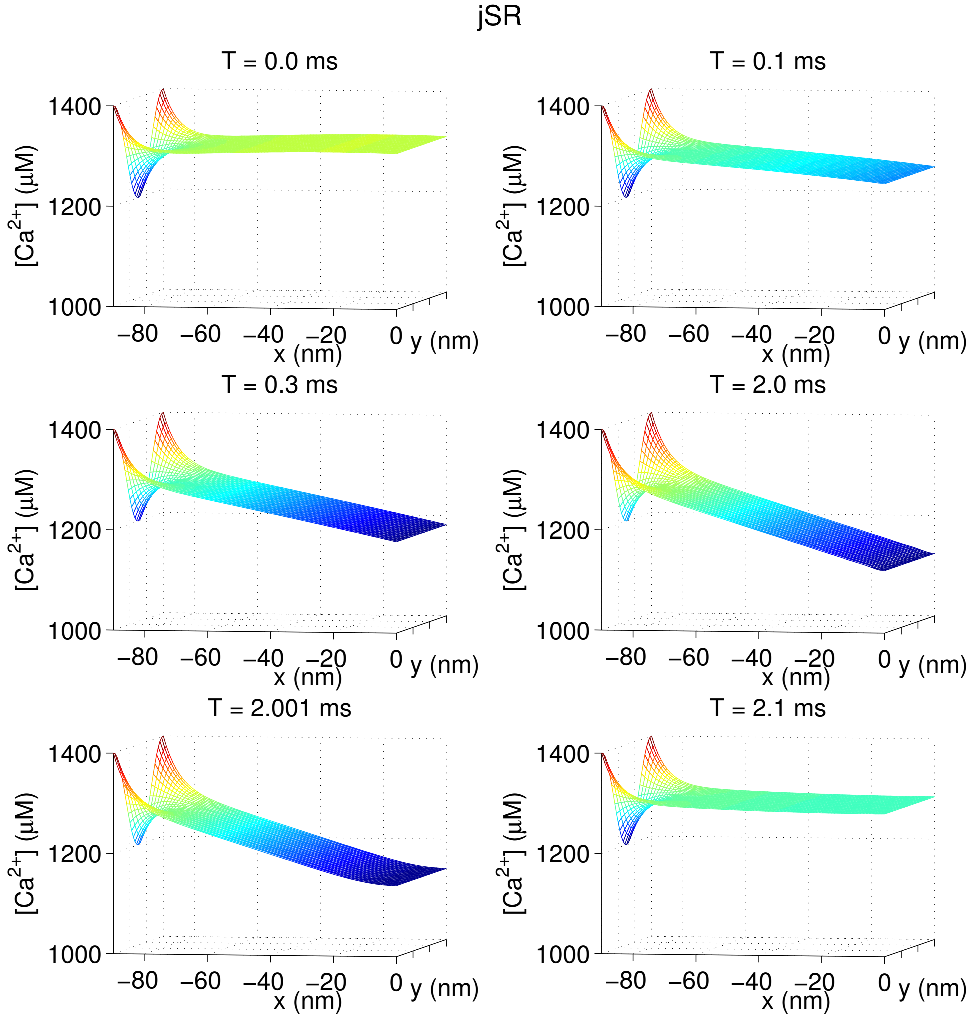

Figure 5.13: Plots of the analytical solution of the 2D problem in the jSR at some different points in time.

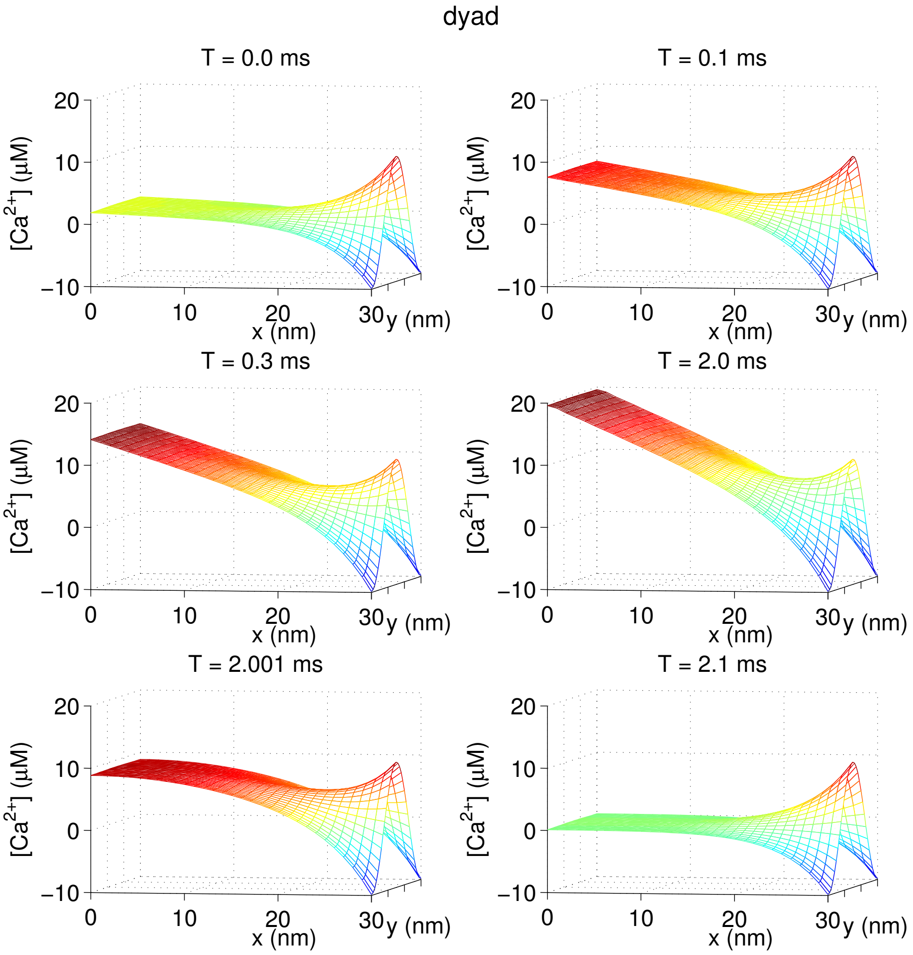

Figure 5.14: Plots of the analytical solution of the 2D problem in the dyad at some different points in time.

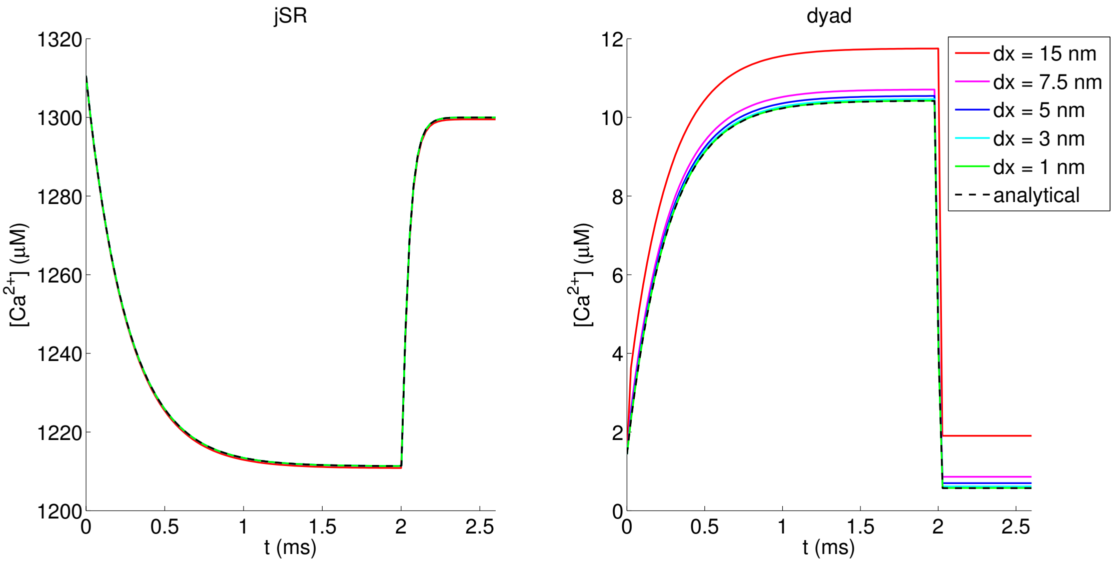

Figure 5.15: Plots of how the solutions in the midpoint of each domain change with time for the 2D problem for some different values of Δx.

Δx (nm)	Error	Error$/\Delta x^2$ (nm^{-2})
15	0.14452	6.42e-04
10	0.089784	8.98e-04
5	0.036755	1.47e-03
3	0.015916	1.77e-03
2	0.0077063	1.93e-03
1	0.0020898	2.09e-03

Table 5.8: Convergence table for the 2D solution at $t = 0.001$ ms. Error is calculated using (5.1) with the norm (5.2). The parameter values are found in Table 2.1, Table 3.1 and Table 3.3.

Δx (nm)	Error	Error$/\Delta x^2$ (nm^{-2})
15	0.10977	4.88e-04
10	0.064281	6.43e-04
5	0.01816	7.26e-04
3	0.0072183	8.02e-04
2	0.0032708	8.18e-04
1	0.00083679	8.37e-04

Table 5.9: Convergence table for the 2D solution at $t = 2.001$ ms. Error is calculated using (5.1) with the norm (5.2) The parameter values are found in Table 2.1, Table 3.1 and Table 3.3.

5.6 Simulations of the 3D problem

Finally, we present results from the numerical simulations of the 3D problem described in Section 2.1.5. The RyR channel is in this case restricted to a quadratic area on the boundary between the jSR and the dyad (see Figure 2.2), and $\bar{g}$ is constant.

As explained in Section 4.2.1 about the finite volume scheme used in the simulations, the discretization parameter is Δx in each spatial direction and in both domains.

In the 3D case, simulations are only carried out for an open channel, and all the parameter values are found in Table 2.1, except for the length parameters, which are specified in Table 2.2.

5.6.1 Results case V

Figure 5.16 shows the numerical solutions in the x, y-direction for $z \approx L_z/2$ in the jSR domain at some different points in time. Figure 5.17 shows the corresponding solutions in the dyad domain. We observe that the calcium concentration in the jSR decreases and the calcium concentration in the dyad increases most extensively in the areas close to the RyR channel. As time goes by, the change in concentration spreads deeper into the domains.

In Figure 5.18 and Figure 5.19, the 3D solutions are plotted in the y, z-direction close to the boundary between the jSR and the dyad, where the largest change in concentration occurs. The change is most prominent in the middle of the area covered by the RyR channel.

Figure 5.20 shows how the numerical approximation of the concentration in a specific point in each of the domains change with time for some different values of Δx. The chosen points are (-5.75 nm, 6.25 nm, 6.25 nm) for the jSR and (5.75 nm, 6.25 nm, 6.25 nm) for the dyad. The numerical approximations of the concentration in these points are simply chosen as the concentration in the finite volume cells containing the points. We again observe differences in the solutions for the different resolutions, now even for the smallest values of Δx.

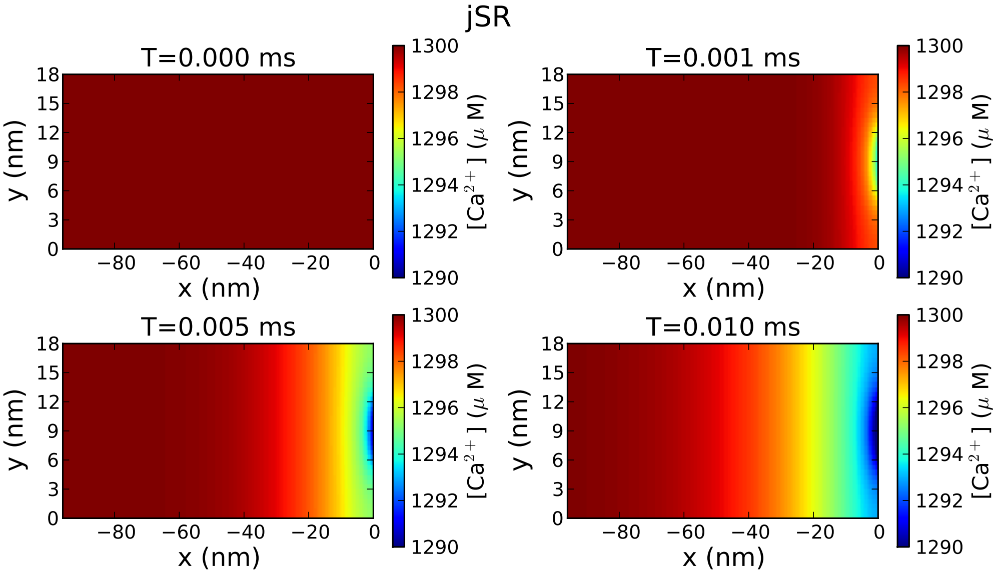

Figure 5.16: Numerical solution of the 3D problem in the x, y-direction for $z \approx L_z/2$ in the jSR domain. The calcium concentration is indicated by the different colours of the colourbar, and we use $\Delta x = 0.5$ nm.

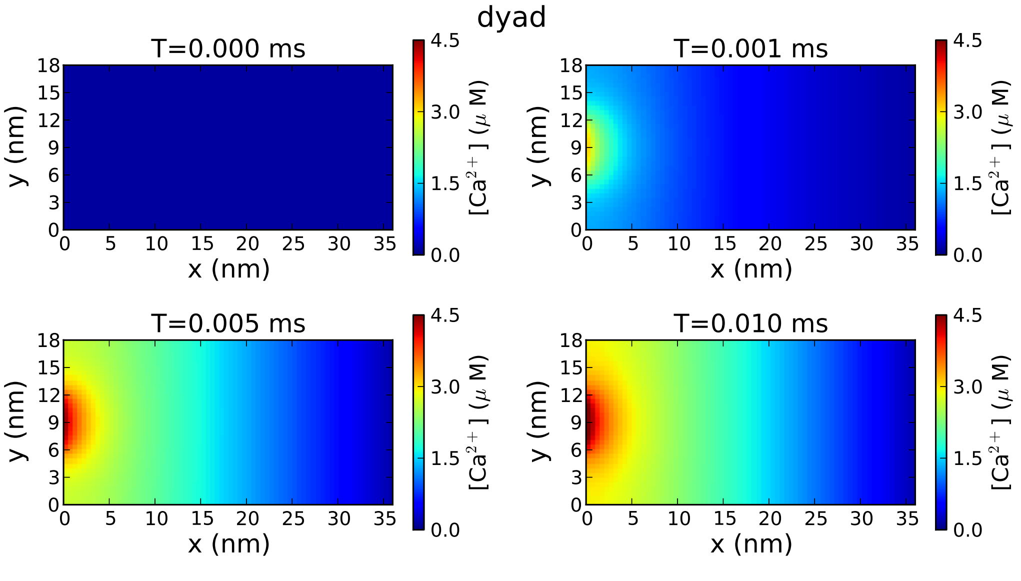

Figure 5.17: Numerical solution of the 3D problem in the x, y-direction for $z \approx L_z/2$ in the dyad domain. The calcium concentration is indicated by the different colours of the colourbar, and we use $\Delta x = 0.5$ nm.

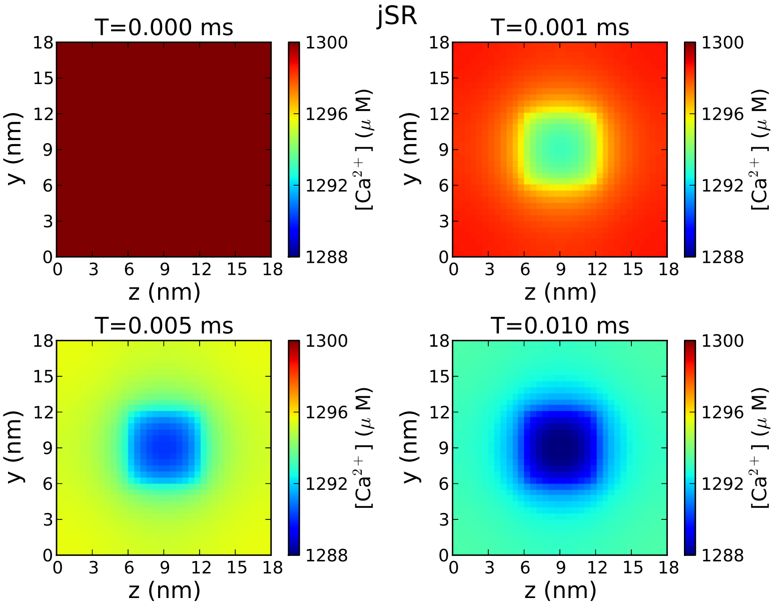

Figure 5.18: Numerical solution of the 3D problem in the y, z-direction for $x \approx 0$ nm in the jSR domain. The calcium concentration is indicated by the different colours of the colourbar, and we use $\Delta x = 0.5$ nm.

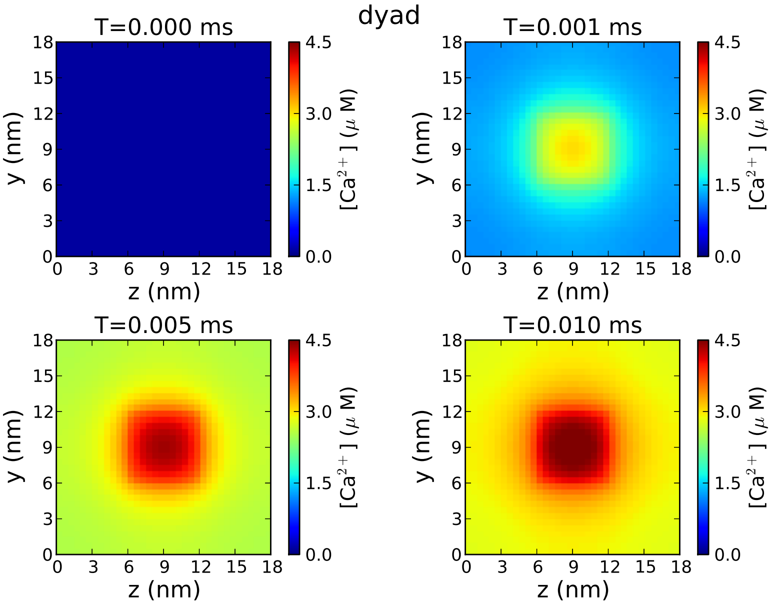

Figure 5.19: Numerical solution of the 3D problem in the y, z-direction for $x \approx 0$ nm in the dyad domain. The calcium concentration is indicated by the different colours of the colourbar, and we use $\Delta x = 0.5$ nm.

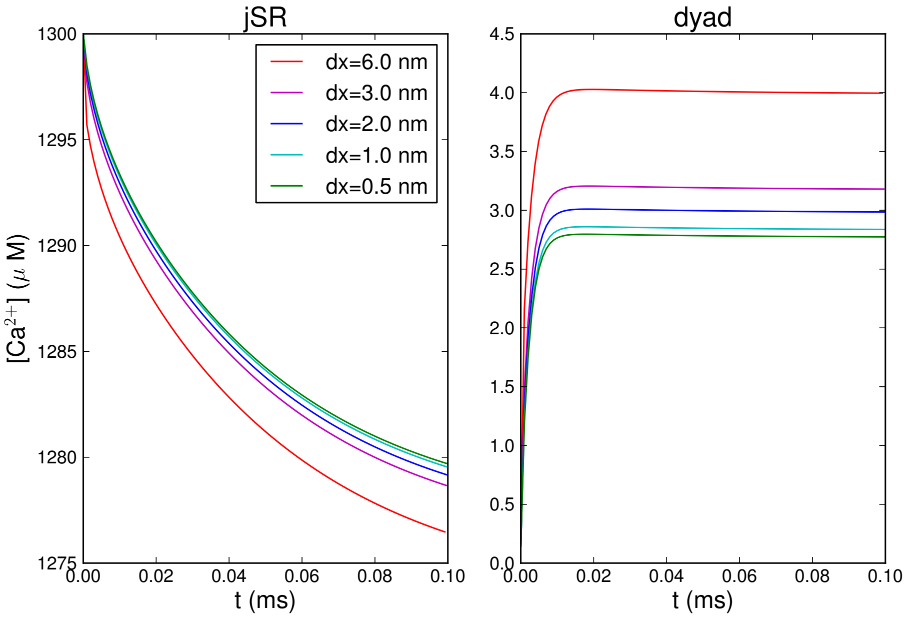

Figure 5.20: Plots of how the numerical approximation of the concentration in a point in each of the domains change with time for some different values of Δx.

Δx (nm)	Error
6	0.088952
3	0.036803
2	0.025128
1	0.0040438
0.67	0.00013779

Table 5.10: Convergence table for the solution of the 3D problem at $t = 0.001$ ms. Error is calculated using (5.1) with the norm (5.2).

5.6.2 Convergence table

We do not have any analytical solutions to compare the numerical approximations to for the 3D problem. Instead, we compare the numerical solutions to numerical solutions produced by simulations with a small Δx.

The solution in each cell in the coarse grid is compared to the solution in the cell in the fine grid covering the midpoint of the cell in the coarse grid. To make sure the cells in the fine grid cover the midpoints of the cells in the coarse grid, the solutions for $\Delta x = 6$ nm, $\Delta x = 2$ nm and $\Delta x = 0.667$ nm are compared to the solutions using the fine resolution $\Delta x = 0.222$ nm, and the solutions for $\Delta x = 3$ nm and $\Delta x = 1$ nm are compared to the solutions using the fine resolution $\Delta x = 0.333$ nm.

The error is again calculated from (5.1) with the norm (5.2), where the analytical solutions are replaced by the fine-grid numerical solutions.

Chapter 6

Consequences

In this chapter, we discuss some consequences of the results from Chapter 5.

We start by giving a short summary of the observed accuracy of the numerical simulations before we consider how the errors in calcium concentration can affect the rates in a Markov model of the RyR channel and how this effect can provide insight into what accuracy is necessary for the simulations.

Afterwards, we consider the consequences of using a resolution suitable for the dyad in whole-cell simulations and discuss some weaknesses of our approach to finding the necessary grid resolution for numerical simulations of calcium dynamics.

6.1 Summary of the accuracy of the numerical simulations

The convergence tables and plots from Chapter 5 provide some insight into the accuracy of the numerical simulations of the considered 1D, 2D and 3D problems. We will now give a short summary of these results.

6.1.1 Accuracy of the 1D simulations

In the simulations of the 1D problem with a constructed $\bar{g}(t)$, the accuracy was generally very good. Considering, for instance, the plots in Figure 5.1 and Figure 5.2, we observed that the numerical solutions for $\Delta x = 15$ nm was almost indistinguishable from the analytical solutions, except for a small difference right after the channel was closed. The accuracy of the solution at $t = 2.001$ ms was investigated in Table 5.3 and Table 5.4, where we observed that in order to get an error smaller than one per cent, we needed to use a Δx of 5 nm or less.

For the 1D problem with a constant $\bar{g}$ and the 1D problem including buffers, we did not have analytical solutions to which we could compare the numerical approximations. However, Figure 5.9 and Figure 5.12 did not display any considerable difference between the solutions for coarse and fine resolutions, suggesting that the numerical approximations were good even for a Δx as large as 15 nm, just like for the constructed $\bar{g}(t)$. The only visible difference between analytical and numerical solutions occurred right after the channel was closed and the error seemed to be close to the one for the constructed $\bar{g}(t)$.

6.1.2 Accuracy of the 2D simulations

In the 2D problem, we introduced some special Dirichlet boundary conditions to create variation in the y-direction. This had a substantial effect on the accuracy of the numerical solutions in the dyad.

Figure 5.15 showed a noticeable difference between the analytical and numerical solutions for the larger values of Δx. In Table 5.8 and Table 5.9, we observed that for $t = 0.001$ ms and $t = 2.001$ ms, we needed to use a Δx of 2 nm or 3 nm, respectively, to get errors smaller than one per cent.

6.1.3 Accuracy of the 3D simulations

In the 3D problem, we replaced the special Dirichlet boundary conditions with a model where the RyR channel was restricted to a limited area of the boundary between the jSR and the dyad.

In this case, we observed differences between the numerical approximations for different resolutions even for small values of Δx (see Figure 5.20). In Table 5.10, we observed that the relative errors were larger in this case than for the 1D and 2D problems and that we needed to use a Δx of 1 nm or less to get an error smaller than one per cent.

6.2 Errors in Markov model of the RyR

From the results summarized above, we have some measures of how accurate the numerical approximations are for different grid resolutions. A remaining question is what accuracy is necessary in these kinds of simulations.

One aspect to consider in this regard is the model of the opening and closing of the RyR channel. In the simple models studied in this thesis, we determined when the channel was open or closed deterministically by assuming that the channel was initially open and assigning some value to the closing time t^*. In more realistic and complex models of calcium dynamics, the opening and closing of the RyR channel is often modelled stochastically.

In such models, the probability of the channel opening or closing will often depend on the calcium concentration outside the channel; recall from Section 1.2 that the RyRs were activated by an increased calcium concentration in the dyad [19]. The numerical approximations should therefore be accurate enough to match the calcium sensitivity of the model of the RyR channel. If a tiny change in the calcium concentration has a huge impact on the model of the opening and closing of the channel, a small error in the calcium concentration may result in a very unrealistic behaviour of the channel, and we therefore need the approximations of the calcium concentration to be very accurate. On the other hand, if a large change in calcium concentration is needed to give any substantial effect on the model of the RyR channel, the consequences of a small error in the calcium concentration are not as serious, and we do not need the accuracy to be as good.

6.2.1 Markov model for the RyR channel

A Markov model is often used to model the opening and closing of the RyR channel. An example of a Markov model is given in Figure 6.1. In this model, we assume that the channel can be either in the closed state (C) or the open state (O). If the channel is in the closed state, the probability of the channel changing to open state during a time interval Δt is given by $\Delta t k_{co}$, where k_{co} is called the opening rate of the channel. Likewise, if the channel is open, the probability of the channel changing to closed state is given by $\Delta t k_{oc}$, where k_{oc} is the closing rate.

The parameters k_{co} and k_{oc} may depend the calcium concentration in the dyad. Figure 6.2 shows how some opening and closing rates from the paper [2] depend on this concentration.

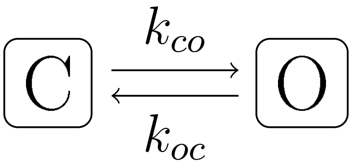

Figure 6.1: Markov model for the RyR channel. The channel can either be in the open state (O) or the closed state (C). The parameters k_{co} and k_{oc} are the opening and closing rates of the channel and may depend on the calcium concentration in the dyad, e.g. like the rates in Figure 6.2.

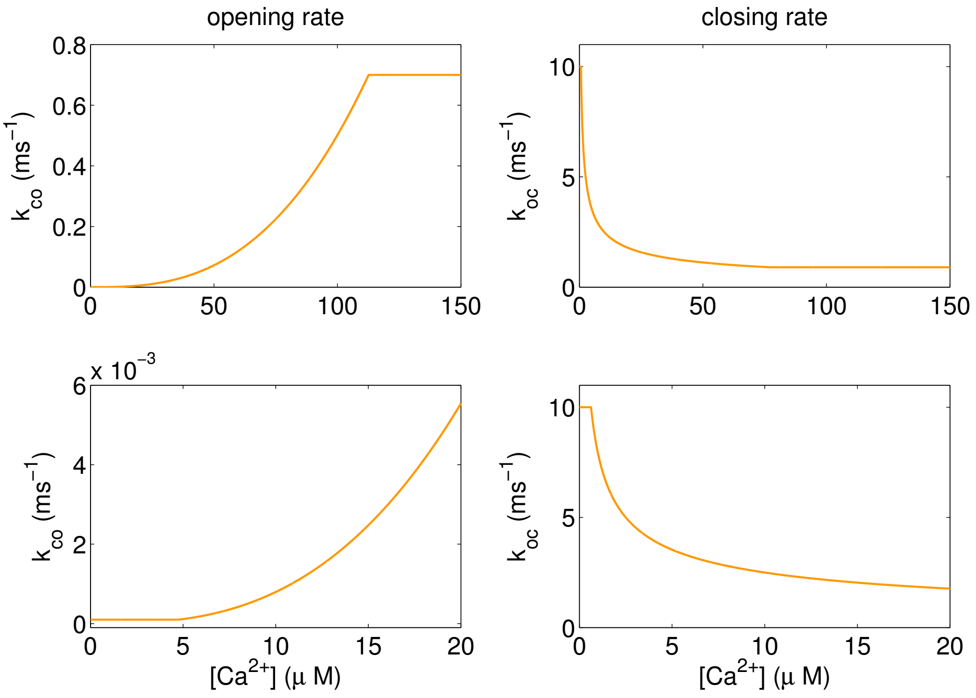

Figure 6.2: Plots of how the opening and closing rates in a Markov model from [2] depend on the calcium concentration in the dyad. In the upper panel, we consider calcium concentrations from 0 to 150 μM, and in the lower panel, we zoom in on concentrations from 0 to 20 μM.

To get an impression of how the errors in calcium concentration can affect the model of the opening and closing of the RyR channel, it could be interesting to examine how large the resulting errors would be for the rate parameters in the Markov model. Below we calculate errors in the rates from [2] corresponding to the errors in calcium concentration from our numerical simulations.

6.2.2 Computing errors in the rate parameters

Using the results of accuracy from the numerical simulations presented in Chapter 5, we can compute the errors in the opening and closing rates for different resolutions by comparing the rates corresponding to the analytical calcium concentration to the rates corresponding to the numerical approximations of the calcium concentration.

In Table 6.1—Table 6.3, we report the errors in the opening and closing rates from [2] for the 1D, 2D and 3D problems. The rate errors are defined as

$$\text{Error} = \frac{\|k(\mathbf{c}_a) - k(\mathbf{c})\|}{\|k(\mathbf{c}_a)\|} \cdot 100\%, \tag{6.1}$$

where the vector $\mathbf{c}$ is a collection of the numerical calcium concentration in each of the grid points of the dyad domain for every 0.001 ms of the simulation time and $\mathbf{c}_a$ is a vector containing the corresponding analytical concentrations. Furthermore, $k(\mathbf{c})$ is a vector containing either the opening or the closing rates corresponding to the calcium concentrations in $\mathbf{c}$, and we use the maximum norm defined as

$$\|\mathbf{u}\| = \max_i |u_i|,$$

where u_i are each of the elements in the vector $\mathbf{u}$.

In other words, we find the relative maximum rate error over time and space.

Rate errors for the 1D problem

Table 6.1 gives the errors in the opening and closing rates of the RyR channel corresponding to the errors in the numerical simulation of the 1D problem with a constructed $\bar{g}(t)$. The simulation is run from $t = 0$ ms to $t = 2.6$ ms, and we close the channel at $t = 2.0$ ms (see Figure 5.5).

Considering Table 6.1, we observe that a 5 nm resolution is enough to make the errors in both rates smaller than one per cent in this case.

Δx (nm)	Error (%)	
	opening rate	closing rate
15	7.7	5.7
10	1.9	2.8
5	0.22	0.48
3	0.16	0.2
2	0.034	0.077
1	0.0084	0.02

Table 6.1: Maximum errors over time and space of the opening and closing rates from [2] for the simulation of the 1D problem. The error is defined by (6.1).

Δx (nm)	Error (%)	
	opening rate	closing rate
15	11	43
10	2.2	35
5	0.68	18
3	0.26	8.1
2	0.11	3.9
1	0.029	1.1

Table 6.2: Maximum errors over time and space of the opening and closing rates from [2] for the simulation of the 2D problem. The error is defined by (6.1).

Rate errors for the 2D problem

In Table 6.2, we report the errors in the opening and closing rates for the 2D problem. The simulation is again run from $t = 0$ ms to $t = 2.6$ ms, and the channel is closed at $t = 2.0$ ms. For the 2D problem, we get some negative concentrations close to the Dirichlet boundary (see Figure 5.14), and in the calculations of the rate parameters, we use the absolute value of the concentrations.

In this case, we observe that none of the considered resolutions give an error smaller than one per cent for the closing rate, but the 1 nm resolution is close, with an error of 1.1 per cent.

Δx (nm)	Error (%) closing rate
6	8.5
3	5.8
2	3.2
1	1.9

Table 6.3: Maximum errors over time and space of the closing rate from [2] for the simulation of the 3D problem. The error is defined by (6.1).

Rate errors for the 3D problem

In Table 6.3, we consider the rate errors for the 3D problem. The simulation is run from $t = 0$ ms to $t = 0.1$ ms with an open channel.

Since we do not have analytical solutions in this case, we replace the analytical solutions in the calculation of the errors with the numerical solutions for a smaller value of Δx. The rates corresponding to the numerical concentration in each cell in the coarse grid is compared to the rates corresponding to the concentration in the fine grid cell covering the midpoint of the coarse grid cell. For $\Delta x = 6$ nm and $\Delta x = 2$ nm, the fine grid solutions come from a simulation using $\Delta x = 0.667$ nm, and for $\Delta x = 3$ nm and $\Delta x = 1$ nm, the fine grid solution comes from a simulation using $\Delta x = 0.333$ nm.

For the 3D problem, the calcium concentration in the dyad is lower than 5 μM (see Figure 5.17). For concentrations this low, the opening rate is assumed to be constant (see Figure 6.2), and changing the concentration due to the numerical errors does not change the opening rate. We therefore only consider the errors in the closing rate in this case. We again observe that we do not get an error smaller than one per cent for any of the considered resolutions, but the 1 nm resolution gives an error smaller than two per cent.

6.2.3 Necessary resolution for simulations of a dyad

The errors in the rate parameters represent a potential consequence of errors in the numerical approximations of the calcium concentration in the dyad since errors in the rate parameters can lead to even larger simulation errors beacuse of errors in the opening and closing of the RyR channel. We are therefore interested in using a grid resolution that gives sufficiently small errors in the rate parameters.

Considering the tables of rate errors above, we are again left with the question of how large errors are acceptable. Assuming that we want the error in the rate parameters to be smaller than one per cent, a resolution of 5 nm is enough for the 1D problem, but for the 2D and 3D problems, the errors are considerably larger. None of the considered resolutions make both errors that small, suggesting that a Δx smaller than 1 nm is necessary.

If we instead just want the errors to be smaller than two per cent, the 1 nm resolution is sufficient for all the considered cases. Similarly, if we accept an error of 3.9 per cent, a 2 nm resolution is enough and if an error of 8.1 per cent is acceptable, we can use a 3 nm resolution.

6.3 Consequences for whole-cell simulations

We now return to the problem introduced in Section 1.3 and consider the consequenses for whole-cell simulations of calcium dynamics.

Recall from Chapter 1 that we are interested in modelling calcium dynamics in cardiac cells because calcium plays an important role in the mechanisms controlling the contraction and relaxation of the heart and that an increased cytosolic calcium concentration is the signal that initiates contraction [19]. The increased concentration is a result of calcium entering the dyad through the voltage-gated LCCs and diffusing through the dyad to activate the RyR channels, which release a larger amount of calcium from the SR [20]. Since these important processes all take place in the dyad, whole-cell simulations of calcium dynamics need to capture the details of what happens in these small regions.

Based on the results presented above, a Δx of 1 nm might be a suitable choice of resolution to get accurate simulations of calcium dynamics in a dyad. To get an impression of the consequences of using this resolution in whole-cell simulations, we will now use estimates from [3] to roughly estimate the size of the computational problem for a 100 ms whole-cell simulation.

Grid size

Assuming that we use a 1 nm resolution in a finite volume method like the one used in the 3D simulations in this thesis for a simulation of a cardiac muscle cell of size 10 μm × 10 μm × 100 μm, the grid would consist of $10,000 \times 10,000 \times 100,000 = 10^{13}$ boxes with sides of length 1 nm.

Number of time steps

If we use an explicit numerical method, the number of required time steps in the simulation depends on how large we can make Δt without violating the governing stability condition. We assume that the model of calcium dynamics includes reactions with buffers, but that the diffusion part of the equations gives the strictest stability condition. In that case, a suitable choice of Δt for a 3D simulation could be [3]

$$\Delta t = \frac{1}{7}\frac{\Delta x^2}{k_{\max}} \approx 6.5 \times 10^{-7} \text{ ms},$$

where $k_{\max}$ is the value of the largest diffusion coefficient appearing in the given system of equations, in our case k_c. Using this Δt, a 100 ms simulation would require approximately 1.5×10^8 time steps.

If we had used an implicit numerical method instead of the explicit method, we would avoid the stability condition and could use a much larger value of Δt. This would imply much fewer time steps in the simulation. On the other hand, using an implicit method implies more floating-point operations for each time step because we need to solve a nonlinear system of equations for every time step.

Total problem size

Assuming that we use an explicit method, the paper [3] estimates around 150 floating-point operations for each box at each time step. Consequently, the total number of floating-point operations for a 100 ms simulation with a 1 nm resolution would be approximately

$$150 \times 10^{13} \times 1.5 \times 10^8 \approx 2.3 \times 10^{23}.$$

This is a huge computational problem. Even using 10,000 nodes on the supercomputer Tianhe-2, where each node has an estimated realistic performance of 150 Gflop/s [3], a 100 ms simulation would require a simulation time of almost five years.

If we could increase the value of Δx to 2 nm, the number of boxes in the grid would be reduced by a factor of 2^3 and the number of time steps would be reduced by a factor of 2^2, making the total number of floating-point operations $2^5 = 32$ times smaller. Using 10,000 nodes on Tianhe-2, a 100 ms simulation would then give a simulation time of about two months.

Increasing the value of Δx to 3 nm would make the size of the problem even smaller as the total number of floating-point operations would be reduced by a factor of $3^5 = 243$ compared to the 1 nm resolution. A 100 ms simulation using 10,000 nodes would then require a simulation time of about a week.

On the other hand, Tianhe-2 is a very powerful supercomputer, currently at the top of the TOP500 list of the most powerful supercomputers in the world [23]. For a more ordinary supercomputer, it might be more realistic to use about 100 nodes. Assuming that each node has a performance of 150 Gflop/s, the simulation time would then be increased by a factor of 100 compared to using 10,000 nodes, and even for the 3 nm resolution the simulation time would be longer than two years.

6.4 Weaknesses

The results of accuracy obtained in this thesis are inevitably affected by our choice of models and parameters. In this section, we discuss weaknesses of some of these choices.

As already mentioned, a disadvantage with the analytical solutions derived in Chapter 3 is that we need to use a constructed $\bar{g}(t)$ in the flux through the RyR channel instead of a constant $\bar{g}$ specified by the model. This weakness has already been discussed in Chapter 3 and Chapter 5. In short, the results of the simulations were clearly affected by using the constructed $\bar{g}(t)$ instead of the constant $\bar{g}$ (see e.g. Figure 5.8), but we found no clear indications that the accuracy was considerably worse for the constant $\bar{g}$ than for the constructed $\bar{g}(t)$ (see e.g. Figure 5.9).

Another factor to consider is the use of constant Dirichlet boundary conditions on the boundaries between the jSR and the nSR and between the dyad and the cytosol for the 1D and 3D problems. This simplification has a physiological justification since the nSR and cytosol represent large compartments relatively far from the RyR channel. Nonetheless, the Dirichlet boundary conditions do ensure that the numerical solutions are correct at these boundaries for every time step, which could make the numerical simulations more accurate than if we had extended the domain and let calcium diffuse through the nSR and cytosol as we would in whole-cell simulations.

Furthermore, the special Dirichlet boundary conditions for the 2D problem had a substantial effect on the accuracy of the simulations, yet the form of these boundary conditions was more motivated by the possibility to fit analytical solutions than to fit a model of calcium dynamics. As a result, we did not really have any proper justifications for the choice of parameter values in these boundary conditions. This is a weakness of the 2D problem beacuse the choice of these parameter values probably had a large impact on the accuracy of the simulations of the model.

In an attempt to get more realistic results, we constructed a 3D problem in which the RyR channel was restricted to a limited area on the boundary between the jSR and the dyad. Even though we did not find analytical solutions in this case and could not determine the error exactly, these simulations provided error estimates that might be more in agreement with realistic simulations of calcium dynamics because we did not have to construct the problem to fit analytical solutions. Nevertheless, the 3D problem studied in this thesis is a very simplified model of calcium dynamics in the dyad. Some of the simplifications that can have affected the accuracy of the simulations are the simple rectangular geometry of the domains and the fact that we only included one RyR channel while there in reality are several RyR channels in the dyad [8].

Another weakness of our approach to studying the accuracy of numerical simulations of calcium dynamics is the absence of a stochastic model for the opening and closing of the RyR channel. As observed above, the errors in the calcium concentration can lead to errors in the opening and closing of the RyR channel, which again lead to even larger simulation errors. Above, we calculated errors in the rate parameters of a Markov model, but we have not done any simulations studying this effect.

In conclusion, we have made several simplifications and choices in our models of calcium dynamics that can have affected the results about the accuracy of the numerical simulations. An advantage of our choices is that we have in some cases been able to derive analytical solutions, which has allowed us to determine exactly how close the numerical approximations are to the exact solutions. A disadvantage is that the obtained results about accuracy might not be representative for more complex and realistic simulations of calcium dynamics.

Chapter 7

Conclusions

In this thesis, we have investigated what grid resolution is necessary for numerical simulations of calcium dynamics in cardiac cells by studying some simple models of calcium dynamics in the dyad. The models consisted of a jSR domain in addition to a dyad domain and modelled how the calcium concentration varied in time and space in these domains when there was release of calcium from the jSR to the dyad through an RyR channel.

Numerical simulations of the models were carried out for different grid resolutions, and the accuracy of the simulations was studied by comparing solutions produced by the numerical simulations to analytical solutions in 1D and 2D and numerical solutions on a fine grid in 3D. Afterwards, this accuracy was compared to the calcium sensitivity of an example of a Markov model for the opening and closing of the RyR channel.

Since it is hard to say exactly how large errors are acceptable, it is hard to conclude exactly what grid resolution would be necessary for the simulations, but the experiments provide some insight into what errors might be expected for different resolutions. For instance, we found that a 1 nm resolution gave errors of up to 1.9 per cent in the rate parameters of the Markov model, a 2 nm resolution gave errors of up to 3.9 per cent and a 3 nm resolution gave errors of up to 8.1 per cent. For whole-cell simulations all of these resolutions result in huge computational problems, which, especially for the 1 nm resolution, might be too large even for very powerful supercomputers.

The models studied in this thesis are very simple models of calcium dynamics, and the accuracy results obtained for these models might not be representative for simulations of more complex models. To gain more insight into what grid resolution is necessary for numerical simulations of calcium dynamics in a dyad, it could therefore be useful to run more simulations, testing what resolution is required in simulations of more complex and realistic models.

One suggestion for further investigation is to model the opening and closing of the RyR channel stochastically using a Markov model. That would allow us to study more closely how errors in the calcium concentration can lead to errors in the opening and closing of the RyR channel and what resolution is required to give sufficiently accurate simulations in this setting.

Moreover, the model could be extended to include several RyR channels in the dyad, allowing us to study what resolution is necessary to give a realistic interaction between the RyR channels in the simulations, i.e. that the release from one channel activates release from nearby channels in a realistic manner.

Another suggestion for further study is to replace the simple geometry used in the simulations in this thesis with a more realistic geometry of the dyad and investigate how this affects the necessary resolution.

Bibliography

[1] N. A. Bearda, D. R. Laverb, and A. F. Dulhuntya. “Calsequestrin and the calcium release channel of skeletal and cardiac muscle”. In: *Progress in Biophysics and Molecular Biology* 85.1 (2004), pp. 33–69.

[2] M. B. Cannell, C. H. T. Kong, M. S. Imtiaz, and D. R. Laver. “Control of Sarcoplasmic Reticulum Ca^{2+} Release by Stochastic RyR Gating within a 3D Model of the Cardiac Dyad and Importance of Induction Decay for CICR Termination”. In: *Biophysical Journal* 104.10 (2013), pp. 2149–2159.

[3] J. Chai, J. Hake, N. Wu, M. Wen, X. Cai, G. T. Lines, J. Yang, H. Su, C. Zhang, and X. Liao. “Towards simulation of subcellular calcium dynamics at nanometre resolution”. In: *International Journal of High Performance Computing Applications* (2013).

[4] A. C. Guyton and J. E. Hall. *Textbook of Medical Physiology (Ninth edition).* W.B. Saunders, 1996.

[5] J. Hake, A. G. Edwards, Z. Yu, P. M. Kekenes-Huskey, A. P. Michailova, J. A. McCammon, M. J. Holst, M. Hoshijima, and A. D. McCulloch. “Modelling cardiac calcium sparks in a three-dimensional reconstruction of a calcium release unit”. In: *The Journal of Physiology* 18 (2012), pp. 4403–4422.

[6] E. R. Higgins, P. Goel, J. L. Puglisi, D. M. Bers, M. Cannell, and J. Sneyd. “Modelling calcium microdomains using homogenisation”. In: *Journal of Theoretical Biology* 247 (2007), pp. 623–644.

[7] L. Hogben. *Handbook of linear algebra.* Chapman & Hall/CRC, 2007.

[8] L. T. Izu, Y. Xie, D. Sato, T. Bányász, and Y. Chen-Izu. “Ca^{2+} waves in the heart”. In: *Journal of Molecular and Cellular Cardiology* 58 (2013), pp. 118–124.

[9] A. M. Katz. *Physiology of the Heart.* Lippincott Williams & Wilkins, 2011.

[10] J. Keener and J. Sneyd. *Mathematical Physiology.* Springer, 2009.

[11] S. H. Lui. *Numerical Analysis of Partial Differential Equations.* John Wiley & Sons, 2011.

[12] J. Y. Murthy and S. R. Mathur. *Class notes for Numerical Methods in Heat, Mass, and Momentum Transfer.* School of Mechanical Engineering. Purdue University. 1998.

[13] T. Myint-U and L. Debnath. *Linear Partial Differential Equations for Scientists and Engineers.* Birkhäuser, 2007.

[14] M. Nivala, E. de Lange, R. Rovetti, and Z. Qu. “Computational modeling and numerical methods for spatiotemporal calcium cycling in ventricular myocytes”. In: *Frontiers in Physiology* 3.114 (2012), pp. 1–12.

[15] P. J. Olver. *Introduction to Partial Differential Equations.* Springer, 2014.

[16] A. J. Pappano and W. G. Wier. *Cardiovascular Physiology (Tenth edition).* Elsevier Mosby, 2013.

[17] A. Quarteroni. *Numerical Mathematics.* Springer, 2007.

[18] R. F. Schmidt and G. Thews. *Human Physiology.* Springer, 1983.

[19] T. R. Shannon. “Integrated Calcium Management in Cardiac Myocytes”. In: *Tutorials in Mathematical Biosciences II: Mathematical Modeling of Calcium Dynamics and Signal Transduction.* Ed. by J. Sneyd. Springer, 2005.

[20] D. U. Silverthorn. *Human Physiology: An Integrated Approach (Sixth edition).* Pearson, 2013.

[21] G. D. Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods (third edition).* Oxford University Press, 1985.

[22] C. Soeller and M. B. Cannell. “Numerical Simulation of Local Calcium Movements During L-Type Calcium Channel Gating in the Cardiac Diad”. In: *Biophysical Journal* 73 (1997), pp. 97–111.

[23] *TOP500 November 2014.* URL: http://www.top500.org/lists/2014/11/ (visited on 03/24/2015).

[24] A. Tveito and R. Winther. *Introduction to Partial Differential Equations: A Computational Approach.* Springer, 2005.

[25] A. Tveito, H. P. Langtangen, B. F. Nielsen, and X. Cai. *Elements of Scientific Computing.* Springer, 2010.

[26] J. M. Varah. “A lower bound for the smallest singular value of a matrix”. In: *Linear Algebra and its Applications* 11.1 (1975), pp. 3–5.

[27] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics. The finite volume method.* Prentice Hall, 1995.

[28] R. L. Winslow, R. Hinch, and J. Greenstein. “Mechanisms and Models of Cardiac Excitation-Contraction Coupling”. In: *Tutorials in Mathematical Biosciences II: Mathematical Modeling of Calcium Dynamics and Signal Transduction.* Ed. by J. Sneyd. Springer, 2005.