University of Padova

SCHOOL OF ENGINEERING
DEPARTMENT OF INFORMATION ENGINEERING

Master degree in Bioengineering

Can Peroxiredoxin-II Protect Vascular Endothelial Growth Factor Receptors from Oxidation by Hydrogen Peroxide?

Author:
Anahita Bayani

Supervisor at University of Padova:
Prof. Alfredo Ruggeri

Supervisors at University of Coimbra:
Dr. Armindo Salvador
Dr. Rui Travasso

Academic Year 2014/2015

Contents

List of abbreviations

Ang-1	Angiopoietin-1 growth factor
ECM	Extra Cellular Matrix
FTCS	Forward Time Centered Space
H_2O_2	Hydrogen Peroxide
PDE	Partial Differential Equations
Prx2	Peroxiredoxin-II
Prx_I	Inactive Peroxiredoxin-II
ROS	Reactive Oxygen Species
Trx1	Thioredoxin
VEGF	Vascular Endothelial Growth Factor
VEGF-R	Vascular Endothelial Growth Factor Receptor

Chapter 1

Introduction

1.1 Angiogenesis

The physiological process of the body to prompt and maintain blood flow through regulated and specific vessel growth mechanisms has longly suscitated interest in the broad research community. To fully understand the developmentof several pathologies, a deep knowledge of angiogenesis is required. In fact several pathologies are associated to either angiogenesis impairment (mainly leading to ischemies, e.g. tumors, diabetic retinopathy). Therefore there is a large interest in the classification and quantification of all the involved parameters in angiogenesis activity of normal and healthy functioning conditions.

Angiogenesis is the process responsible for generating new blood vessels from the already existing capillary structures. In this sense, proliferation and cell migration are crucial steps for forming and stabilising new endothelium, eventually undergoing specific differentiatio. The main angiogenic regulating factors can be summarised into three main contributions [1]. Firstly, fibroblasts growth factors FGF2 and bFGF lead mesenchymal cells development into angioblasts, particularly affecting the earliest stages of vascular generation. A second determining family of proteins heavily involved in angiogenesis regulation is the vascular endothelial growth factor (VEGF), with VEGF-A mostly inducing endothelial cells proliferation and migration through signalling regulation of the tyrosine kinases receptors VEGF-R1 and VEGF-R2. Finally, angiopoietin-1 growth factor (Ang-1) also influences new vessels formation, by covering and stabilising newly formed endothelium with pericytes, upon binding with receptor Tie-2.

Unlike pathological angiogenic processes, normal angiogenesis is time-limited and particularly occurrs in specific physiological events, such as organ regeneration and injuries healing, embryogenesis, female ovulation, menstruation and placenta formation. On the other hand, variations in the equilibrium between angiogenic promotors and repressors may lead to

tumoral, diabetic, arthritic and psoriatic related pathologies.

Two different angiogenic mechanisms are recognised nowadays [2], sprouting and intussusceptive angiogeneses, both present in many different tissues and organs, with the former mainly characterised by new branching around an exisitng vessel and the latter consisting of an expanding transvascularisation within the tissue. Sprouting angiogenesis is prompted by target levels of hypoxia within affected tissues whose parenchymal cells respond to by secreting VEGF-A, thus inducing new endothelial formation by a VEGF-R2 (receptor for VEGF-A) dependant cell migration and aggregration, followed by proliferation and capillary sprout elongation. The newly formed capillaries are further stabilised by perycites and extra-cellular matrix (ECM) mechanical signalling. Intussusceptive or splitting angiogenesis on the other hand presents a lumen formation in the vessel walls, eventually dividing it. Lacking cell proliferation and migration, it it is generally faster and more efficient than sprouting angiogenesis, occurring most prominently during embryional development and at sites of already existing capillaries. Controlling mechanisms of intussusceptive angiogenesis are still not completely understood, with suggestions of sensitivity to VEGF-A and mechanical stresses as inducing factors.

1.2 Vascular endothelial growth factor receptors

While three different VEGF tyrosine kinase receptors are known, namely VEGF-R1, VEGF-R2 and VEGF-R3, the second is the one mostly involved in angiogenesis processes, particularly affecting vessels permeability and sprouting [3]. Sharing a similar structure, they induce autophosphorylation of an intracellular kinase domain upon binding of an extracellular ligand domain, thus causing the receptors dimerization [4]. In this sense, VEGF-R2 has been found to present four autophosphorylation sites, connecting to VEGF-A, VEGF-C, VEGF-D and VEGF-E. Different downstream signalling pathways are promoted by VEGF-R2, both at lymphatic and vascular level. Tyrosine residues initiate signalling towards endothelial cell proliferation, also causing successive cell migration via inhibition of an adaptor protein. Overall, though specifics in the functioning mechanisms and pathways are not fully understood, VEGF-R2 determines endothelial cells development, heavily featuring at their survival, proliferation, migration stages.

1.3 VEGF-R2 and H_2O_2

Reactive oxygen species (ROS) have longly been studied and analysed in regard to their detrimental effect to normal physiologic processes, being primarily involved in oxidative stress damages and generally promoting cell

death. Contextualising ROS activity within angiogenesis, it is actually interesting to underline how at low levels they induce growth factors production and activation. The complexity of ROS intervention in angiogenic related pathways is however highly challenging, with a recognised role in preserving and affecting angiogenesis but also possibly inhibiting it.

Hydrogen peroxide (H_2O_2) in particular, is a stable and durable species, thanks to its electrons being paired. It results from superoxide dismutation and enzymatic production and, at low levels, is directly involved in intracellular signalling and regulatory activity, also by promoting tyrosine phosphorylation in grow factor receptors [5]. This is an important key aspect in angiogenic mechanisms, especially on its action on VEGF-R2 by phosphorylation signalling and mediation [6]. While H_2O_2 is a first initiator of angiogenesis, it also prominently features in the intracellular mechanisms which regulate it. Internal production of superoxides typically presided by NADPH oxidases, are specifically enhanced by VEGF binding to VEGF-R2, thus further stimulating NADPH oxidase. The resulting ROS inhibit tyrosine phosphatases with a consequent increase in VEGF-R2 autophosphorylation [5] which prompt VEGF signalling action of endothelial sprouting and further support the growth factor action. On the other hand, VEGF also induces the H_2O_2 production through superoxide dismutases, thus exposing ROS dual role in angiogenic processes.

1.4 Reactive interactions between Peroxiredoxin II and H_2O_2

Physiological presence of H_2O_2 in human metabolism has long been under investigation and still many aspects remain unclear. While the main source of reactive oxygen species comes from mithocondria, at blood level erythrocytes present superoxide from the natural hemoglobin autoxidation [7]. In this regard, it is of particular interest to fully understand and determine mechanisms and concentrations involved in H_2O_2 interactions with erythrocytes.

Within this frame, the role of peroxiredoxin II protein (Prx2) is of essential importance, reducing H_2O_2 through cycling reactions. Prx2 has been in fact detected as one of the main proteins involved in cellular H_2O_2 reduction, while also being the third most abundant protein in human erythrocytes. Particularly, H_2O_2 reduction by Prx2 evolves through a cycling structured chain of reactions that provide sulfenic acid (C51-SOH) via oxidation of Prx2 cysteine subunits (C51-SH), when exposed to H_2O_2. Disulfide (C51-S-S-C172) formed from reacting sulfenic acid and a second cysteine from a contiguous subunit (C172-SH) is further reduced by thioredoxin (Trx1), thus completing a first cycle by regenerating Prx2. On the other hand, it is interesting to highlight how Prx2 inactivation occurs upon sulfenic acid

oxidation into sulfinic (C51-SO_2H) or sulfonic (C51-SO_3H) acids by further presence of H_2O_2 [8].

On the other hand, the exact functioning of Prx2 in this regard may still be unclear, in the light of the contrasting results yielded by direct reduction rates measurements and experimental evidencies.

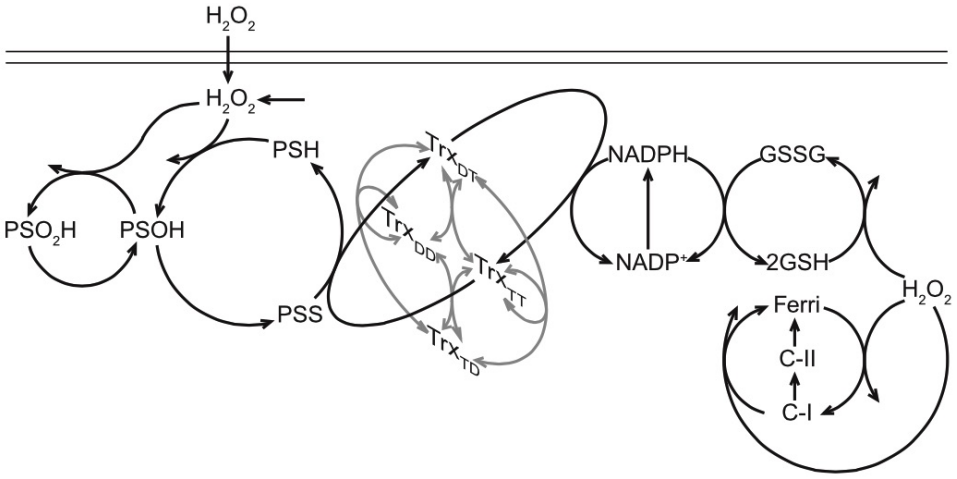


Figure 1.1: H_2O_2 reduction scheme in human erythrocytes, considering both internal and external hydrogen peroxide sources and monomer state for all components. Original figure from [8].

1.5 Prx2 as a protection mechanism for VEGF-R2

Along with the relevance of oxidative species and particularly H_2O_2 in controlling and signalling cellular mechanisms, significant attention is also upon antioxidant enzymes as regulating factors. In this sense, the role of Prx2 as the most abundant antioxidant has been further characterised [6], highlighting its function, among many others, in preserving the signalling pathway between VEGF and VEGF-R2. Notably, Prx2 protects VEGF-R2 activation and autophosphorilation as essential elements in binding to VEGF, thus promoting vascular endothelial cells proliferation, migration and survival. In this regard, preservation of VEGF-R2 tyrosine kinase has been shown to be strongly dependant on Prx2 presence that, accordingly, is crucial in maintaining controlled levels of cellular H_2O_2, in order to not compromise VEGF-R2 activated state. Dephosphorylation induced by phosphatases on VEGF-R2 (of which the exact mechanisms have not yet been fully understood) affects the receptor functioning, eventually attenuating its signalling [9], while, on the other hand, H_2O_2 has also an inhibiting action on phosphatases, thus enhancing the proteins phosphorylation. As such, Prx2 is one of the key enzymes (along with gluthanione peroxidase, catalase and

peroxiredoxin I) in regulating H_2O_2 physiological level within endothelial cells, also by acting on membrane mechanisms, especially because of its local accumulation near VEGF-R. This results are also crucial for the receptor activity, particularly conditioning its downstream signalling.

In the absence of Prx2, H_2O_2 inactivates VEGF-R2 by oxidizing its C1206 cysteine residue [6]. In this sense, the protecting mechanisms acted by Prx2 has been determined to be strictly connected to the enzyme distribution within the cellular environment, with both Prx2 and VEGF-R2 being actually mainly localised in vascular endothelial cells caveolae and in tight connection. This further suggests that VEGF-R2 reduction is preserved through Prx2 colocalisation in the caveolae. In fact, VEGF pro-angiogenic activity via its type-2 receptor interaction evolves at caveolar domains, practically constituting the functional platform for angiogenesis. In this regard, Prx2 emerged as one of the proteins with highest sensibility to VEGF at caveolae level [10], thus further highlighting Prx2 crucial role in regulating ROS levels in order to preserve angiogenesis processes.

Chapter 2

Problem modelling

2.1 Objectives

Standing within the biological frame explained in the introduction, the main objective of this thesis is to model a system evidencing the interactions occurring at cellular level within the membrane, by considering a simplified version of the reaction scheme in figure 1.1, as visible in figure 2.1. In order to assess Prx2 contribution in regulating H_2O_2 levels within VEGF receptors, only a fraction of the cell is considered, comprising a tract of membrane and the immediate inner portion.

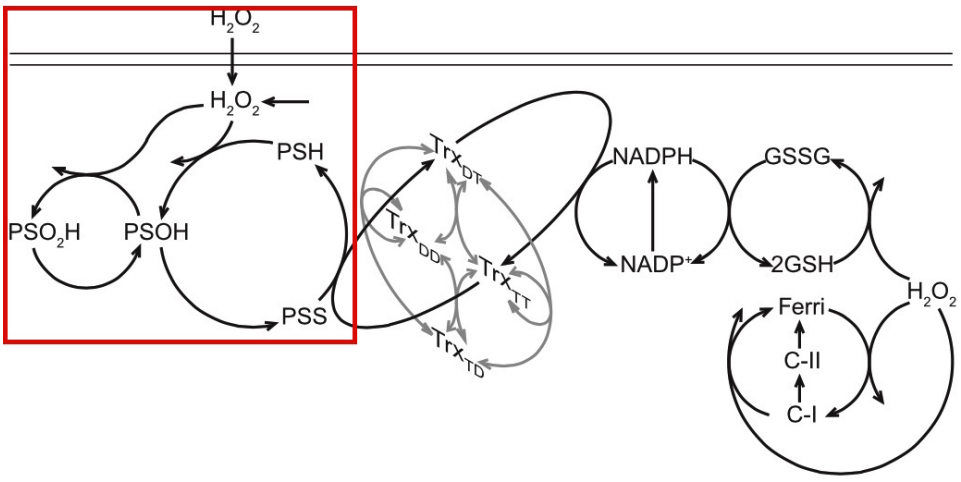


Figure 2.1: H_2O_2 reduction scheme in human erythrocytes. In red box the scheme considered for modellisation. Original figure from [8].

Since such interactions are effectively modelled by partial differential equations (PDE), firstly a brief theoretical basis will be given, followed by the resolution of some initial problems, in order to familiarise with the mathematical tools and their implementation in Matlab. As such, two introducing

problems are proposed, while the ultimate model will consider Prx2 in its monomeric components, with the addition of intermediate steps in the modelling process, to evaluate the system suitability, gradually introducing more complicate aspects.

The viability of such a model comes straightforwardly from the studies of Benfeitas et al [8] in which new H_2O_2 reduction rates by Prx2 were proposed, resolving the discrepancy existing to that point between computationally predicted data and experimental evidence.

2.2 PDE theoretical basis

Many physical problems are adequately described by partial differential equations (PDE), that is presenting a dependence on two or more independent variables (contrary to ordinary differential equations that rely on a single independent variable), modelling different phenomena, with solutions available through numerical methods. In this regard, let us recall the main definitions and properties of PDEs which will serve as mathematical basis of the dynamic interactions between Prx2, hydrogen peroxide and VEGF presented in the introduction.

Definition 2.1 (PDE)**.** A partial differential equation is a function in a domain $\Omega \subset \mathbb{R}^n$ relating indepedent variables $\mathbf{x} \in \Omega$ to an unknown function u in Ω and its finite partial derivatives. Thus, the m-th order PDE, with $m \in \mathbb{Z}^+$, in $\Omega \subset \mathbb{R}^n$ is given by

$$F(\mathbf{x}, u, \nabla u(\mathbf{x}), \nabla^2 u(\mathbf{x}), \dots, \nabla^m u(\mathbf{x})) = 0 \qquad \mathbf{x} \in \Omega.$$

Straightforwardly, a PDE is linear if there is linearity between the dependent variables (unknown function) and its derivatives depending on $\mathbf{x}$.

A unique solution to PDEs is provided upon additional conditions that allow to identifiy one in the infinite possible values associated to the generic equation. In this sense, typically given costraints include initial and boundary conditions. As for the latter, the solution or its derivatives value at the domain boundary is specified. Three main sets of conditions may be given.

- Dirichlet condition, defining $u(\mathbf{x})$ at the domain frontier $\delta\Omega$:

$$u(\mathbf{x}) = f(\mathbf{x}) \qquad \mathbf{x} \in \delta\Omega;$$

- Neumann condition, providing normal derivative at boundary (representing flux through the boundary):

$$\frac{\partial u}{\partial x_i} u(\mathbf{x}) = f(\mathbf{x}) \qquad i = 1, \dots, n, \quad \mathbf{x} \in \delta\Omega;$$

- Robin or third kind condition, relating u and its normal derivatives values at boundary:

$$\alpha(\mathbf{x})\frac{\partial u}{\partial x_i}u(\mathbf{x}) + u(\mathbf{x}) = f(\mathbf{x}) \qquad i = 1, \ldots, n, \quad \mathbf{x} \in \delta\Omega.$$

Alternatively, mixed, oblique and nonlocal boundary conditions can also be provided.

Within this frame, it is particularly interesting considering a generic second order PDE, in the form of

$$A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} + D\Big(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\Big) = 0, \tag{2.1}$$

of independent variables (x, y) and unknown function u, with the discriminant $B^2 - 4AC$ determining the three types of hyperbolic ($B^2 - 4AC > 0$), parabolic ($B^2 - 4AC = 0$) and elliptic ($B^2 - 4AC < 0$) PDEs.

In order to reach a numerical solution, PDEs can be conveniently discretised in a number of ways, such as with a three-point formula based on finite differences approximation:

$$\frac{\partial^2 u}{\partial^2 x} = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\Delta x)^2} + O(x^2) \tag{2.2}$$

and

$$\frac{\partial^2 u}{\partial^2 y} = \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{(\Delta y)^2} + O(y^2), \tag{2.3}$$

being $u(x, y)$ defined in a rectangle $[0, a] \times [0, b]$ and considering its discrete grid of points

$$(x_i, y_j) = (i\Delta x, j\Delta y) \qquad 0 \leq i \leq N - 1, \quad 0 \leq j \leq M - 1,$$

by $\Delta x = a/(N - 1)$ and $\Delta y = b/(M - 1)$.

While several types of PDEs can be recognised, both in their mathematical setting of ellyptic, hyperbolic or parabolic and the variables they represent, a focus will be on parabolic heat equations, which will be here briefly presented and further developed. As per this setting, the considered function is defined in time t and space coordinates (x, y), in the general form of heat conduction or diffusion equation:

$$\frac{\partial u}{\partial t} = k\nabla^2 u + D\Big(u, x, y, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\Big), \qquad t > 0, \quad (x, y) \in \Omega. \tag{2.4}$$

Analogously to the explicit method for numerical solutions as in (2.2) and (2.3), a forward time centered space (FTCS) scheme can also be adopted,

particularly when dealing with equations in the form of (2.4). As such, the approximation is yielded by

$$u(x,y,t+\Delta t) = u(x,y,t) + \Delta t \cdot \Bigg[D\Big(u,x,y,\frac{\partial u}{\partial x},\frac{\partial u}{\partial y}\Big) + \\ + \frac{u(x+\Delta x,y,t) - 2u(x,y,t) + u(x-\Delta x,y,t)}{\Delta x^2} + \\ + \frac{u(x,y+\Delta y,t) - 2u(x,y,t) + u(x,y-\Delta y,t)}{\Delta y^2} \Bigg], \quad (2.5)$$

with stability in the two dimensions space guaranteed under

$$\frac{k \cdot \Delta t}{\Delta \ell^2} < \frac{1}{4} \qquad \ell = x, y$$

if $\Delta x^2 = \Delta y^2$.

In this sense, a number of introductory problems were considered and solved with Matlab, in order to approach the diffusion modelling of hydrogen peroxide within the biological context presented in chapter 1.

2.3 Problem 1

The first problem to be considered was an evolving model based on PDEs in the form of

$$\frac{\partial m}{\partial t} = m - m^3 + \nabla^2 m, \quad (2.6)$$

representing a ferromagnet magnetisation. The FTCS resolving scheme was adopted, as in eq. (2.5), integrating in a squared domain of edge $L = 100$, given periodic boundary condition and initial points $m(x,y,0) = 0.05 \cdot$ (`rand()`-0.5). The resolution implemented in Matlab (code in appendix A.1) provides a diffusion map evolution in time.

2.4 Problem 2

In order to better assess how initial conditions and the system dimensions affect the results, a second set of problems were introduced, by considering the classic heat equation as in (2.4) and testing differential values.

Firstly, equation

$$\frac{\partial m}{\partial t} = \nabla^2 m \quad (2.7)$$

was solved with initial conditions $m(x,y,0) = e^{-\frac{(x-L/2)^2+(y-L/2)^2}{100}}$.

Analogously, a system of equations was considered:

$$\begin{cases} \frac{\partial m}{\partial t} = \nabla^2 m + 5n \\ \frac{\partial n}{\partial t} = \nabla^2 n - 0.1n \end{cases} \quad (2.8)$$

with initial conditions given by

$$\begin{cases} m(x,y,0) = e^{-\frac{(x-L/2)^2+(y-L/2)^2}{100}} \\ n(x,y,0) = e^{-\frac{(x-L/3)^2+(y-L/2)^2}{50}} \end{cases}$$

Finally, the last variant to be dealt with was the heat equation

$$\frac{\partial m}{\partial t} = \nabla^2 m \tag{2.9}$$

in three spatial dimensions (x, y, z), with initial conditions $m(x, y, z, 0) = e^{-\frac{(x-L/2)^2+(y-L/2)^2+(z-L/2)^2}{100}}$.

The respective Matlab codes are listed in appendix A.2.

2.5 Problem 3

In order to present a more precise description of the system evolution, the typical diffusion equation, as in the form of (2.7) is repurposed and modified with particular attention to the parameters values and their orders of magnitude, as well as the boundary conditions. In this sense, let us consider a setting in which H_2O_2 is produced extracellularly, permeating the membrane through colocalisation of an aquaporin protein with VEGF-R2 and Prx2 within the caveolae and reacting with them. In such a context, H_2O_2 concentration is evaluated in the $0.01 - 100\ \mu M$ scale; as for the boundaries, periodic costraints were mantained in the ordinates dimension while frontier abscissas values took into account null concentration at bottom and an entering rate of $\frac{\partial c}{\partial x} = -\frac{1}{D} c_0 \frac{dV}{A}$ at top (that is at membrane, having modelled an $L_x \times L_y$ box, assuming a fraction of membrane and its immediate inner portion), with D being the diffusion set at $D = 1.83 \cdot 10^{-9}\ m^2 s^{-1}$, the cellular area $A = 1.3 \cdot 10^{-10}\ m^2$ and the flux $dV = 1.02 \cdot 10^{-14}\ m^3 s^{-1}$.

From the above considerations, particularly recognising the parameters as related to H_2O_2, thus $\frac{\partial c_{H_2O_2}}{\partial t} = D_{H_2O_2} \nabla^2 c_{H_2O_2}$, the Prx2 contribution is added.

Within this framework, Prx2 evolution in time is also considered, by describing its interaction with H_2O_2 by a PDE, as well as taking into account also the amount of non active Prx2 ($Prx2_I$), thus resulting in the following system:

$$\begin{cases} \dfrac{\partial c_{H_2O_2}}{\partial t} = D_{H_2O_2} \nabla^2 c_{H_2O_2} - k \cdot c_{H_2O_2} \cdot c_{Prx2} \\ \dfrac{\partial c_{Prx2}}{\partial t} = D_{Prx2} \nabla^2 c_{Prx2} - k \cdot c_{H_2O_2} \cdot c_{Prx2} \\ \dfrac{\partial c_{Prx2_I}}{\partial t} = D_{Prx2} \nabla^2 c_{Prx2_I} + k \cdot c_{H_2O_2} \cdot c_{Prx2} \end{cases} \tag{2.10}$$

resolved as in appendix A.3.

2.6 Problem 4

As already highlighted in figure 1.1, the reduction of H_2O_2 by Prx2 is better described at a monomeric level, particularly considering the peroxidatic cysteines that participate in the reactions chain. As such, while following the modelling pattern of the previous section, the aim is now to represent in the equations system C51-SH cysteine oxidation into the sulfenic acid C51-SOH (PSH→PSOH), which in turn condenses with the resolving cysteine C172-SH forming the disulfide dimer C51-S-S-C172 (PSOH→PSS), eventually reduced by Trx1 (PSS→PSH). A second cycle involves instead C51-SOH reaction with H_2O_2 yielding sulfinic acid C51-SO_2H (PSOH→PSO_2H), thus blocking the condensation [11].

The above described reactions result then in the following system:

$$\begin{cases} \frac{\partial c_{H_2O_2}}{\partial t} = D_{H_2O_2}\nabla^2 c_{H_2O_2} - k_1 \cdot c_{H_2O_2} \cdot c_{PSH} - k_4 \cdot c_{H_2O_2} \cdot c_{PSOH} \\ \frac{\partial c_{PSH}}{\partial t} = D\nabla^2 c_{PSH} - k_1 \cdot c_{H_2O_2} \cdot c_{PSH} + k_3 \cdot c_{PSS} \\ \frac{\partial c_{PSOH}}{\partial t} = D\nabla^2 c_{PSOH} + k_1 \cdot c_{H_2O_2} \cdot c_{PSH} - k_4 \cdot c_{H_2O_2} \cdot c_{PSOH} - k_2 \cdot c_{PSOH} + k_5 \cdot c_{PSO_2H} \\ \frac{\partial c_{PSS}}{\partial t} = D\nabla^2 c_{PSS} + k_2 \cdot c_{PSOH} - k_3 \cdot c_{PSS} \\ \frac{\partial c_{PSO_2H}}{\partial t} = D\nabla^2 c_{PSO_2H} + k_4 \cdot c_{PSOH} \cdot c_{H_2O_2} - k_5 \cdot c_{PSO_2H}. \end{cases} \tag{2.11}$$

As for the parameters value, the expressions were taken from Benfeitas et al. [8] analysis and adapted to the monomeric simplified model that is considered in table 2.6.

Reaction	Rates	
PSH+H_2O_2→PSOH+H_2O	$k_1 = 10^2$	$\mu M^{-1}s^{-1}$
PSOH→PSS+H_2O	$k_2 = 1.7$	s^{-1}
PSS→PSH	$k_3 = 2.1 \cdot 10^{-2}$	s^{-1}
PSOH+H_2O_2→PSO_2H+H_2O	$k_4 = 1.2 \cdot 10^{-2}$	$\mu M^{-1}s^{-1}$
PSO_2H→PSOH	$k_5 = 10^{-4}$	s^{-1}

Diffusion costants are set at $D_{H_2O_2} = 1.8 \cdot 10^{-9}\ m^2s^{-1}$ and $D = 10^{-11}\ m^2s^{-1}$.

The matlab code used for model 2.11 are reported in appendix A.4 accordingly.

2.7 Problem 5

To represent in more detail the biological context of Prx2 action, it is important to consider that the concentration of this protein is considerably higher

nearby VEGF-R. Within the matlab framework, in which the fraction of the cell under analysis is set as an $(L_x \times L_y)$ box, two different equations have now to be taken into account: internal Prx2, in function of both (x, y), and Prx2 at membrane, depending only on one dimension y. As such, two new rates are also introduced, an association constant k_a and dissociation constant k_d. Considering an ϵ thickness for the membrane, the new equations that update model 2.11 are:

$$\begin{cases} \dfrac{\partial c_{Prx2}(x,y)}{\partial t} = D\nabla^2 c_{Prx2} + reactives - k_a \cdot c_{Prx2}(x,y) \cdot \delta_{j\leq\epsilon} + \dfrac{k_d c^m_{Prx2}}{\epsilon}\delta_{j\leq\epsilon} \\ \dfrac{\partial c^m_{Prx2}(y)}{\partial t} = \displaystyle\sum_{j=1}^{\epsilon} k_a(x) c_{Prx2}(x_j, y) - k_d c^m_{Prx2} + reactives + D(x)\dfrac{\partial^2 c^m_{Prx2}}{\partial y^2} \end{cases} \tag{2.12}$$

with δ being the discriminating term to identify membrane proximity and c^m_{Prx2} indicating Prx2 concentration at membrane. Equations 2.12 are thus integrated in the matlab code as in appendix A.5. Corresponding equations for the remaining Peroxiredoxin species are also implemented with equal values for the association and disocciation constants. The "reactive" terms in equations 2.12 correspond to the terms in equations 2.11 that describe the reactions where the different oxidation states of the peroxiredoxin participate.

Chapter 3

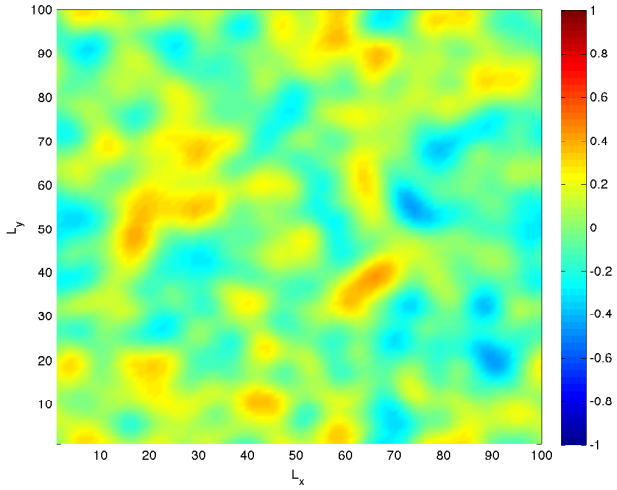
Results

According to the matlab codes presented in the previous chapter 2, the corresponding plots and results are here considered and discussed.

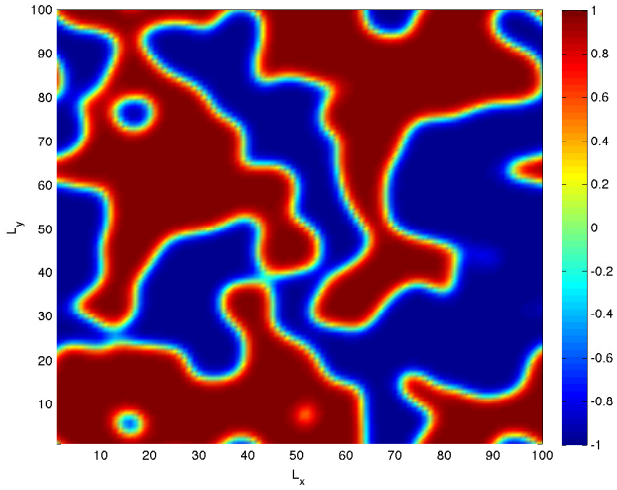
3.1 Magnetisation model

Section 2.3 firstly introduced a reaction-diffusion system in $2D$, in the form of a magnetisation problem. As such, simulation of the reaction is showed in its inital phase in which aleatory fluctuations are visible (figure 3.1a), an intermediate one with small magnetised regions (figures 3.1b and 3.1c), in which small domains coarsen to form larger domains, and the state of the system at its final interaction, where magnetisation is wider (figure 3.1d), with domains that have become larger.

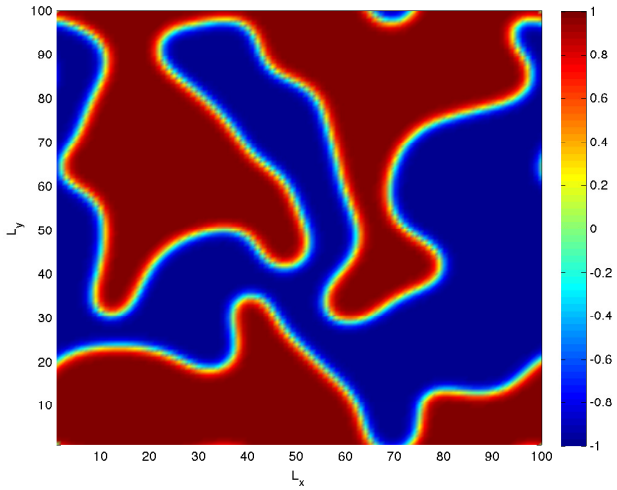
Equation 2.6 represents in fact an increase in magnetic domains, as in the density plots of figure 3.1. This equation describes the magnetisation rearrangement, due to a rapid lowering of the system temperature [12].



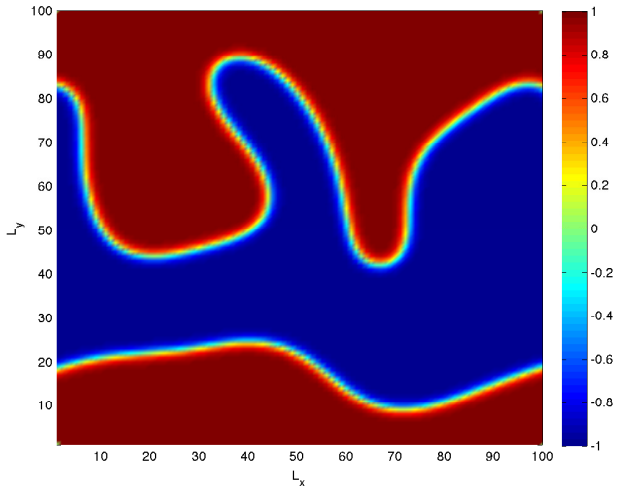
(a) Plot at iteration no.50 and time $t = 5s$



(b) Plot at iteration no.100 and time $t = 10s$



(c) Plot at iteration no.250 and time $t = 25s$



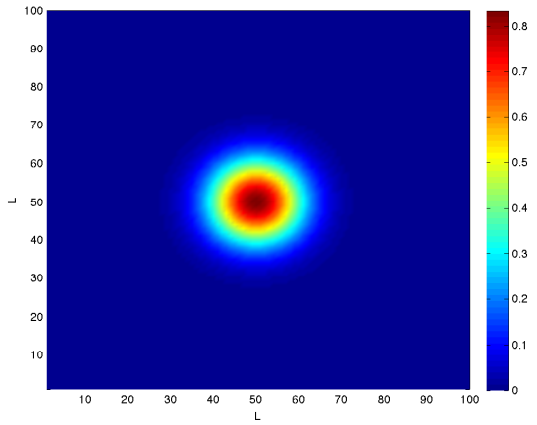
(d) Plot at iteration no.1000 and time $t = 100s$

Figure 3.1: Density plots evolution of a reaction-diffusion process of magnetisation.

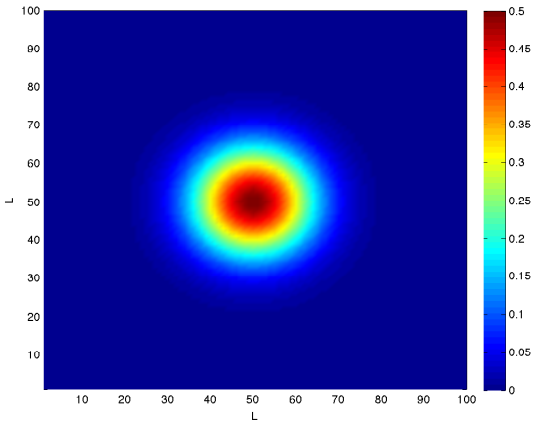
3.2 Heat diffusion model

In section 2.4 heat equations were considered, by varying dimensions and initial boundary conditions of the systems. Particularly, resolution of eq. 2.7 simply follows a diffusion process within an $(L \times L)$ box, as showed in figures 3.2.

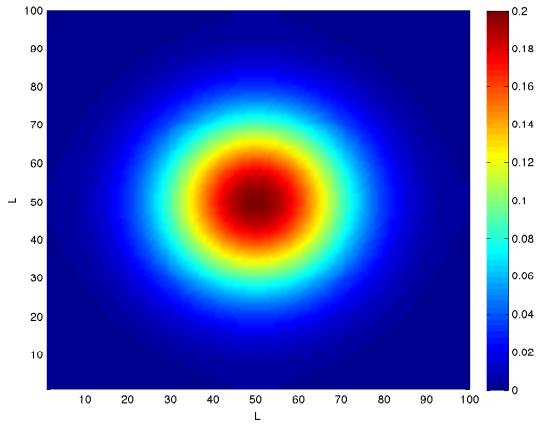
An initial concentration can be observed that progressively diffuses throughout the whole system, from an initial central localisation.



(a) Plot at iteration no.50 and time $t = 5s$



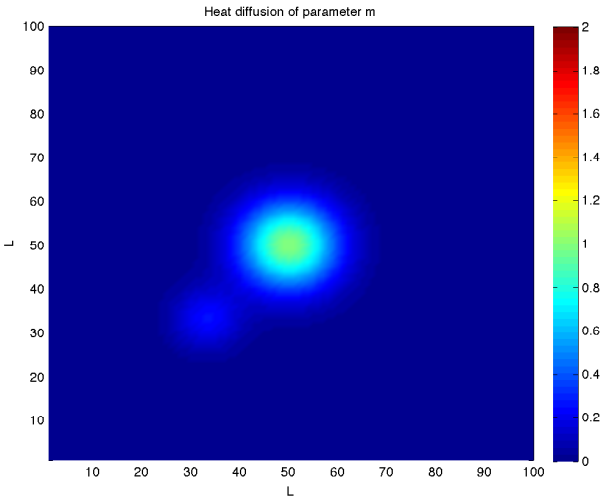
(b) Plot at iteration no.250 and time $t = 25s$



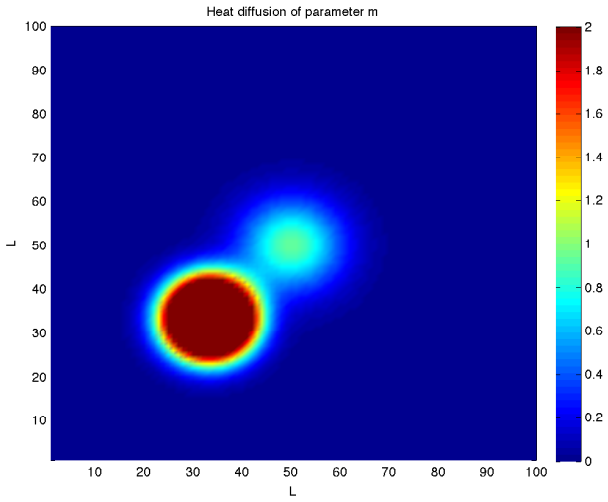
(c) Plot at iteration no.1000 and time $t = 100s$

Figure 3.2: Density plots evolution of a heat-diffusion process.

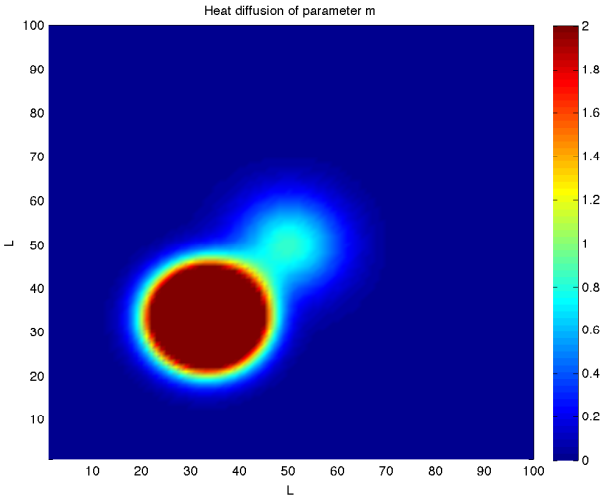
Heat diffusion equations were also coupled in a system, as in eq. 2.8, where diffusion of two entities m and n was studied. As the equation suggests, it is visible in figure 3.3 that m increases where n is concentrated. On the other hand, as the colorbar indications show, n is consumed at a higher rate (figure 3.4).



(a) Plot at iteration no.1 and time $t = 0.05s$

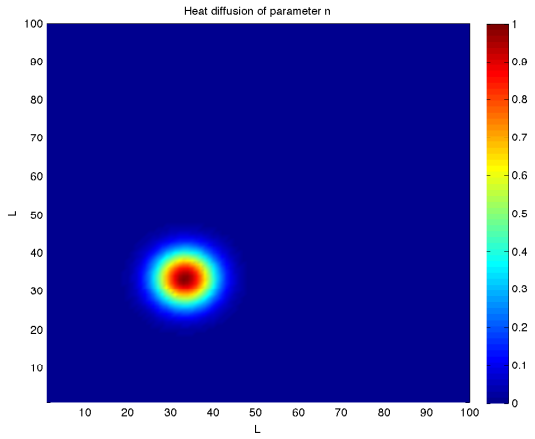


(b) Plot at iteration no.40 and time $t = 2s$

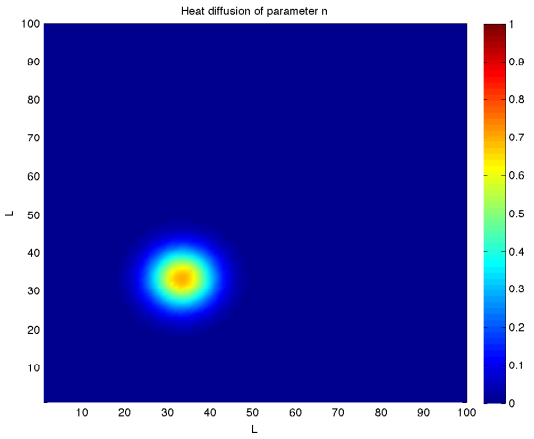


(c) Plot at iteration no.100 and time $t = 5s$

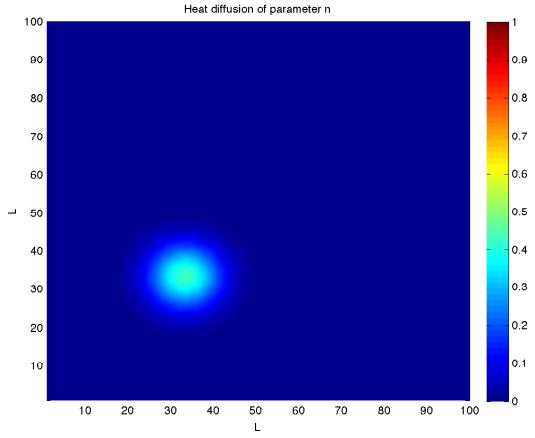
Figure 3.3: Density plots showing the evolution of entity m heat-diffusion process.



(a) Plot at iteration no.1 and time $t = 0.05s$



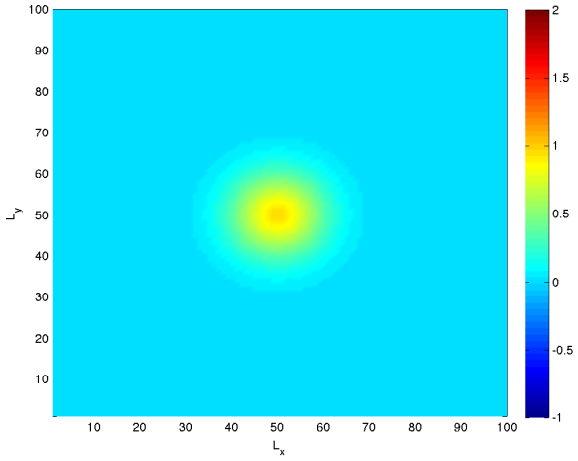
(b) Plot at iteration no.40 and time $t = 2s$



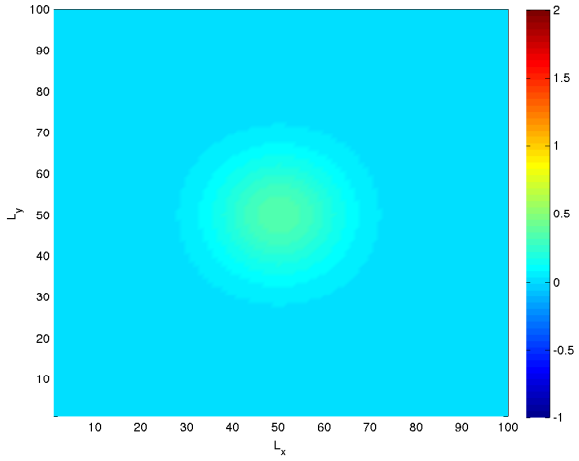
(c) Plot at iteration no.100 and time $t = 5s$

Figure 3.4: Density plots showing the evolution of entity n heat-diffusion process.

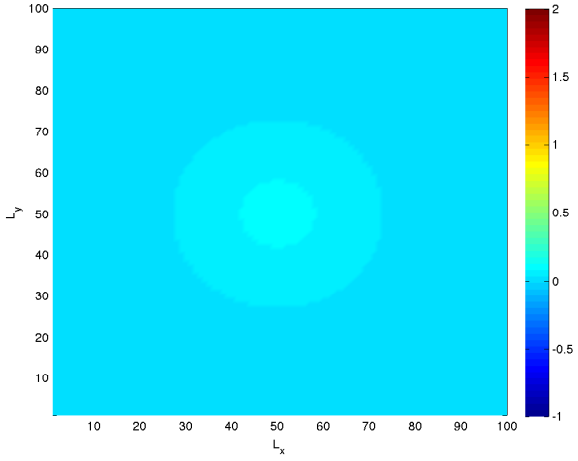
The final equation proposed in section 2.4 (eq. 2.9) regarded a $3D$ system. In order to show its evolution in time, $2D$ plots of L_x and L_y dimensions were represented and, as denoted in figures 3.5, there is a diffusion process that yields a decrease of concentration in the centre, as confrimed by the colorbars indications, with the initial concentration in the order of 1 (expressed in light orange, as in figure 3.5a) and progressively lowering towards zero (figure 3.5c).



(a) Plot at iteration no.10 and time $t = 1s$



(b) Plot at iteration no.250 and time $t = 25s$



(c) Plot at iteration no.1000 and time $t = 100s$

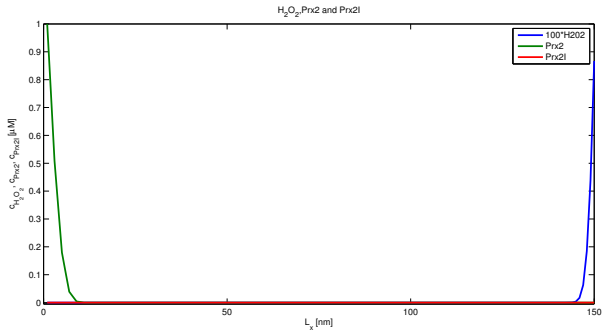
Figure 3.5: Density plots evolution of a $3D$ system, represented at L_x and L_y dimensions and a fixed line in L_z.

3.3 Primary cellular model

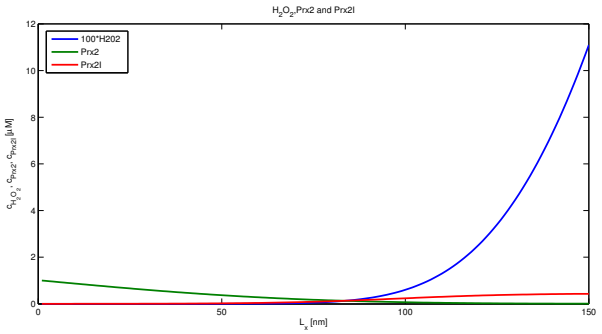
With problem 3, in section 2.5, a first attempt in dealing with H_2O_2 concentration within a cell was proposed. As such, while the system described by equation 2.10 was in its simplest form, a first set of results can be observed in figures 3.6, 3.7, 3.8 and 3.9 where, respectively, the concentrations of the different species as a function of L_x at a median line of the L_y dimension and the density plots are represented.

Accordingly, the results in figure 3.7 show how H_2O_2 enters the membrane at $L_x = 150\ nm$ as a function of time. Plots of hydrogen peroxide concentration along the L_x dimension of the cell are also presented in figure 3.6, in which a decreasing trend is evident, particularly due to Prx2 intervention in reducing hydrogen peroxide at the membrane proximity. Prx2 diffuses in fact through the cell along L_x, thus reaching H_2O_2 in the membrane proximity. This progressing concentration is represented in figures 3.8.

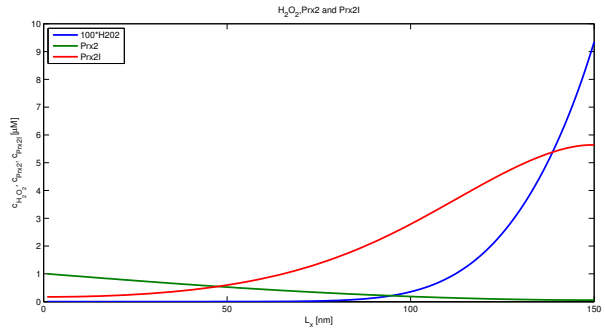
When Prx2 and H_2O_2 react, the concentration of Prx_I increases. In this sense, an amount of Prx_I begins to be visible from figure 3.6b, with an increasing concentration through time, in correspondence to Prx2 consumption (figures 3.6c and 3.6d). The relative density plots at figure 3.9 further highlight this, by showing an initial null concentration (figure 3.9a) which can then be found towards the membrane (figures 3.9b, 3.9c and 3.9d), thus validating the purpose of H_2O_2 reduction close to the membrane.



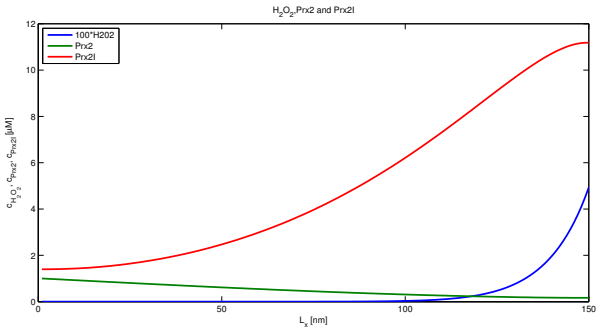
(a) Plot at iteration no.10 and time $t = 10^{-7}s$



(b) Plot at iteration no.3000 and time $t = 3 \cdot 10^{-5}s$

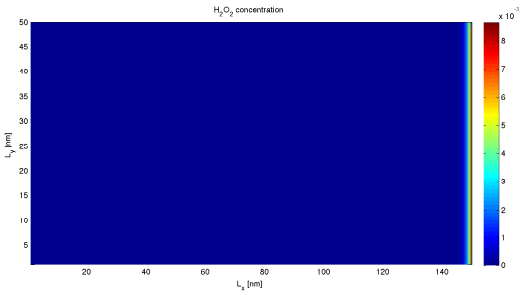


(c) Plot at iteration no.6000 and time $t = 6 \cdot 10^{-5}s$

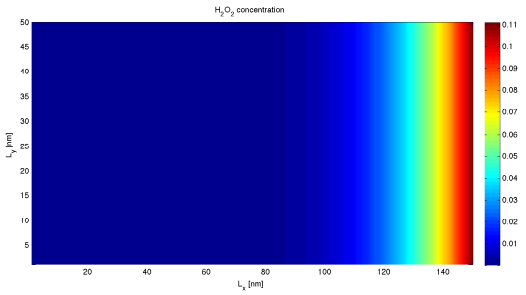


(d) Plot at iteration no.10000 and time $t = 10ms$

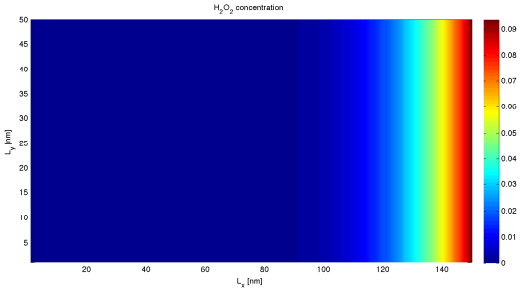
Figure 3.6: Plots of H_2O_2, Prx2 and Prx_I concentrations throughout L_x at $L_y/2$.



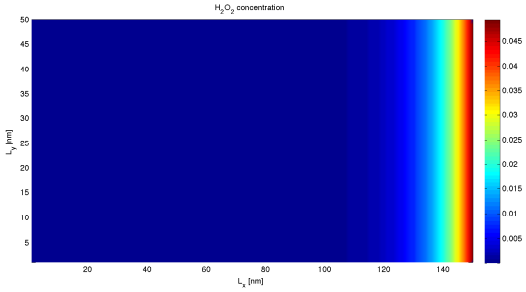
(a) Plot at iteration no.10 and time $t = 10^{-7}s$



(b) Plot at iteration no.3000 and time $t = 3 \cdot 10^{-5}s$

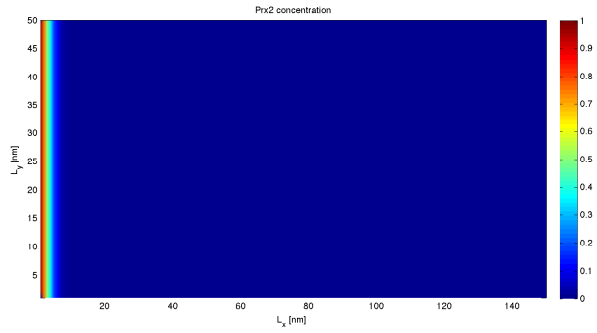


(c) Plot at iteration no.6000 and time $t = 6 \cdot 10^{-5}s$

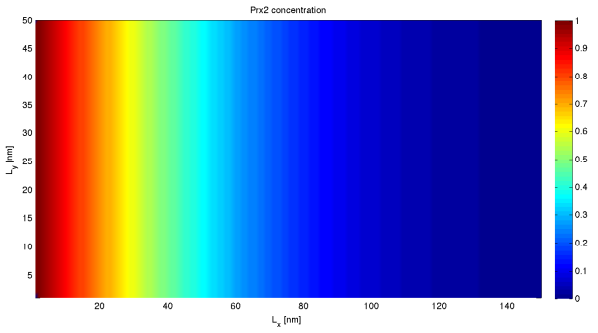


(d) Plot at iteration no.10000 and time $t = 10ms$

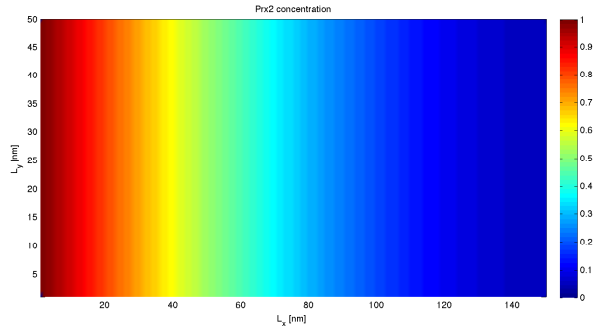
Figure 3.7: Density plots of H_2O_2 concentration in an $(L_x \times L_y)$ portion of a cell.



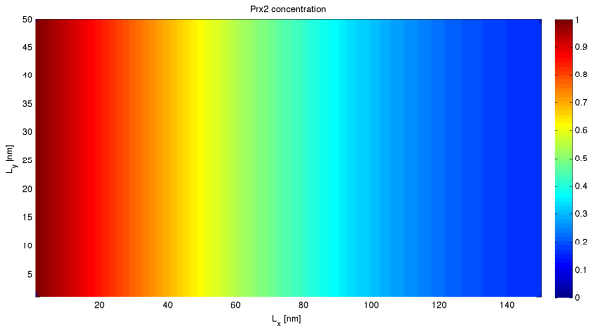
(a) Plot at iteration no.10 and time $t = 10^{-7}s$



(b) Plot at iteration no.3000 and time $t = 3 \cdot 10^{-5}s$

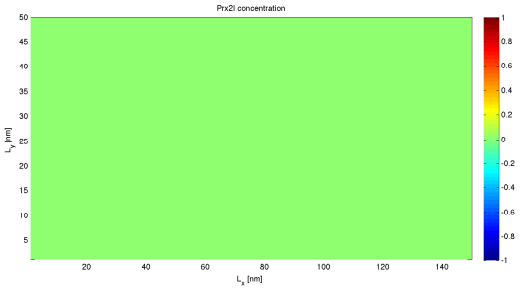


(c) Plot at iteration no.6000 and time $t = 6 \cdot 10^{-5}s$

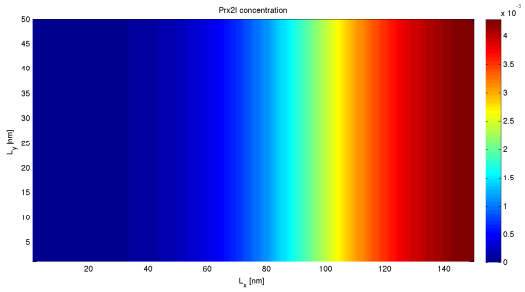


(d) Plot at iteration no.10000 and time $t = 10ms$

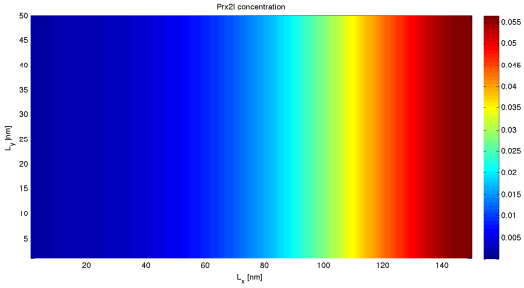
Figure 3.8: Density plots of Prx2 concentration in an $(L_x \times L_y)$ portion of a cell.



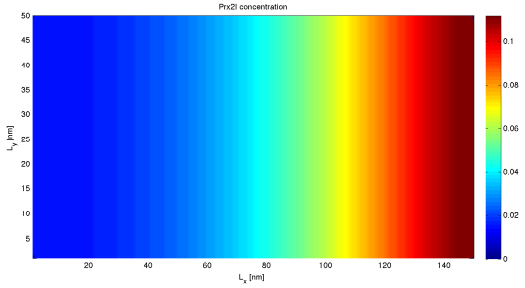
(a) Plot at iteration no.10 and time $t = 10^{-7}s$



(b) Plot at iteration no.3000 and time $t = 3 \cdot 10^{-5}s$



(c) Plot at iteration no.6000 and time $t = 6 \cdot 10^{-5}s$



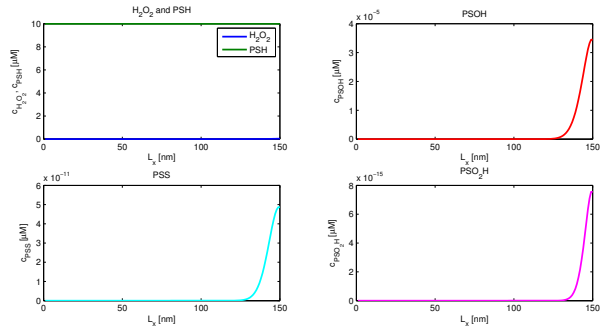
(d) Plot at iteration no.10000 and time $t = 10ms$

Figure 3.9: Density plots of $\mathrm{Prx_I}$ concentration in an $(L_x \times L_y)$ portion of a cell.

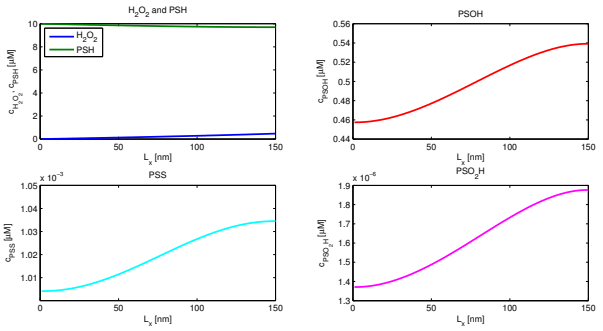
Finally, in order to assess the reliability of models in 2.6 and 2.7, a discussion of the expected and obtained results is here presented, particularly addressing biological significance and repercussions.

3.4 Monomeric peroxiredoxin II homogeneous membrane model

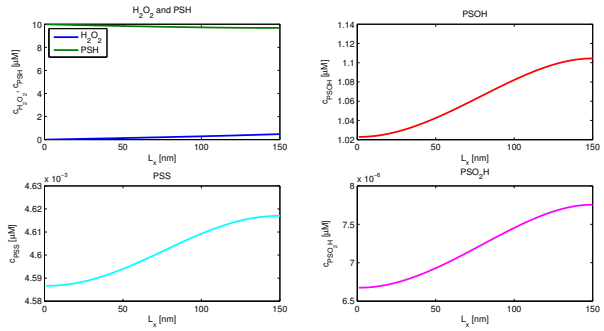
In reference to problem 4, where Prx2 was already considered in its monomeric components, cellular membrane, represented by the L_x^{th} line of the $(L_x \times L_y)$ box, has no preferential receptor site for Prx2 to bind to. In this sense, it is not expected to have an enhanced reactive activity on a limited portion of the membrane. In fact, throughout the L_x^{th} line and progressing into the cell, hydrogen peroxide provokes PSH oxidation to PSOH, which can then be further oxidized by H_2O_2 to PSO_2H or yield PSS. The latter is in turn reduced back to PSOH. As such figure 3.10 highlights this trend throughout the 10 ms of the simulation time.



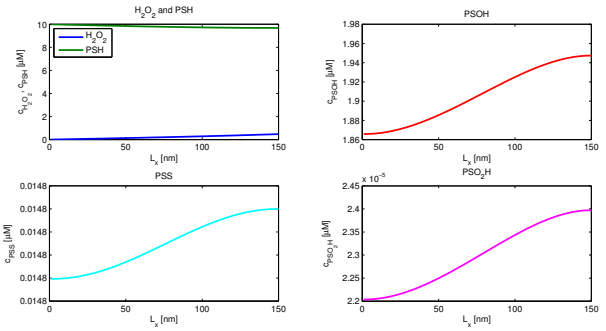
(a) Plot at iteration no.100 and time $t = 2.7 \cdot 10^{-3} ms$



(b) Plot at iteration no.100000 and time $t = 2.7 ms$



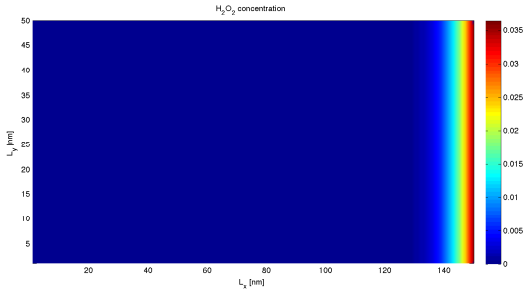
(c) Plot at iteration no.200000 and time $t = 5.4 ms$



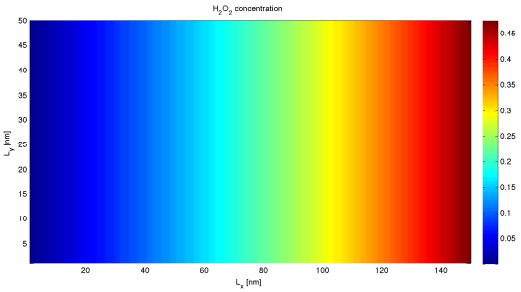
(d) Plot at iteration no.350000 and time $t = 9.45 ms$

Figure 3.10: Plots of H_2O_2, PSH, PSOH, PSS, PSO_2H concentrations in the cell fraction $(0 < L_x < 150$ delining its intern and $L_x = 150$ the membrane).

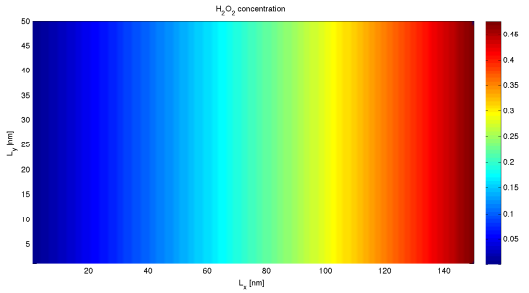
Furthermore, the density plots of H_2O_2 and Prx2 also confirm the homogeneous concentration at membrane with a quite uniform gradient through the cell, as visible in figure 3.11.



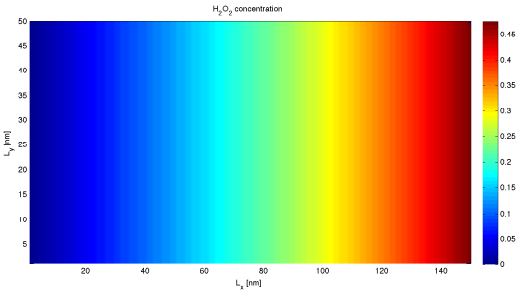
(a) Plot at iteration no.100 and time $t = 2.7 \cdot 10^{-3} ms$



(b) Plot at iteration no.100000 and time $t = 2.7 ms$



(c) Plot at iteration no.200000 and time $t = 5.4 ms$

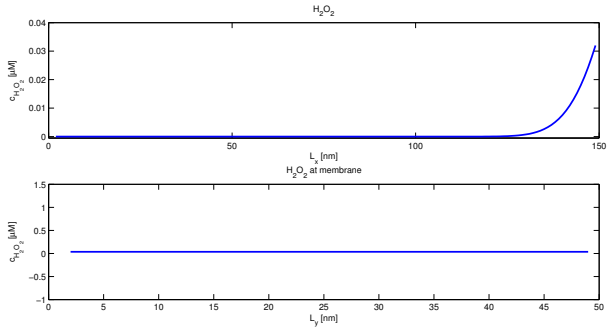


(d) Plot at iteration no.350000 and time $t = 9.45 ms$

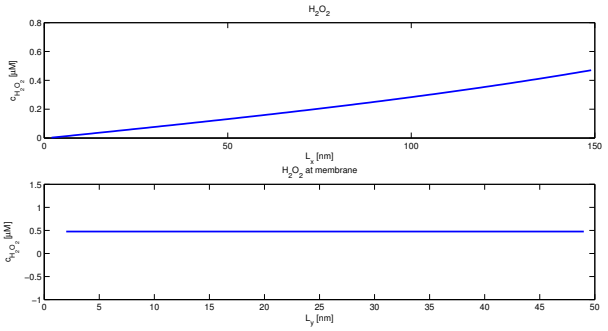
Figure 3.11: Density plots of H_2O_2 concentration in the cell fraction (from blue corresponding to zero to red for the highest concentration).

Finally, graphs of H_2O_2 and Prx2 monomers singular concentrations through the L_x and L_y dimensions summarise the density plots by showing the progressing concentration at one line ($L_y/2$ as internal and L_x for membrane), as visible in figures 3.12 to 3.16.

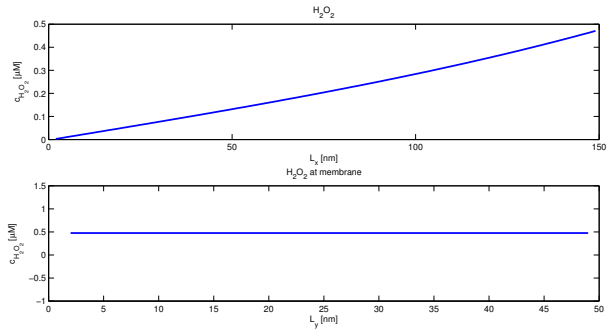
Overall, the system is described by H_2O_2 entrance at L_x and Prx2, considered in its monomeric components, diffusion in the cell. In particular, a descrease in PSH at membrane can be observed, due to the higher presence of H_2O_2 in this region, with the resulting reactions yielding PSOH, which has therefore a higher concentration near the membrane. On the other hand, in presence of H_2O_2, PSOH can be further oxidated to PSO_2H, thus explaining the latter also presenting a peak in concentration towards the membrane. This concentration eventually increases in function of time due to a lower availability of H_2O_2 which in turn is being reduced by PSH.



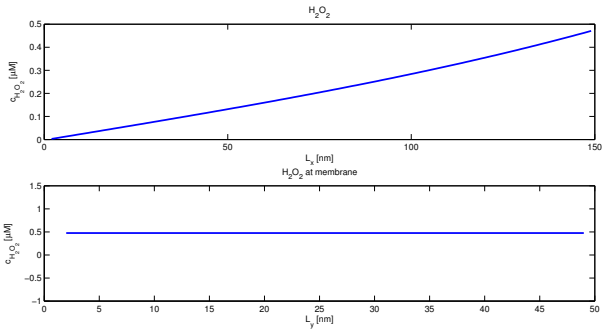
(a) Plot at iteration no.100 and time $t = 2.7 \cdot 10^{-3} ms$



(b) Plot at iteration no.100000 and time $t = 2.7ms$

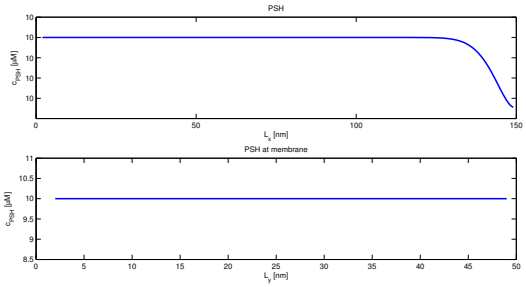


(c) Plot at iteration no.200000 and time $t = 5.4ms$

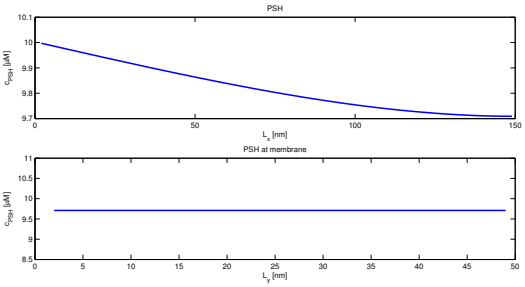


(d) Plot at iteration no.350000 and time $t = 9.45ms$

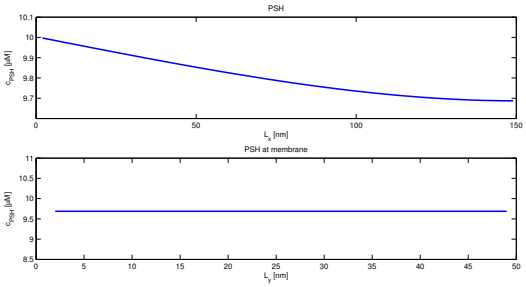
Figure 3.12: Plots of H_2O_2 concentration at cell fraction middle width ($L_y/2$) throughout the L_x dimension (above) and at membrane ($L_x - 1$) throughout the cell fraction width L_y (bottom).



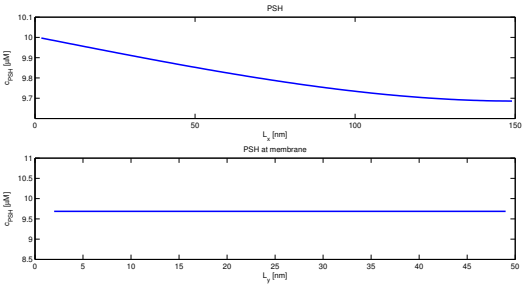
(a) Plot at iteration no.100 and time $t = 2.7 \cdot 10^{-3} ms$



(b) Plot at iteration no.100000 and time $t = 2.7 ms$

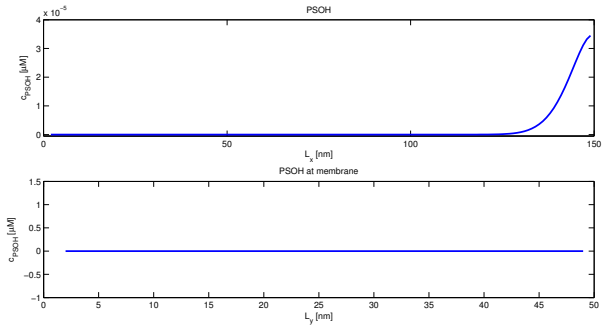


(c) Plot at iteration no.200000 and time $t = 5.4 ms$

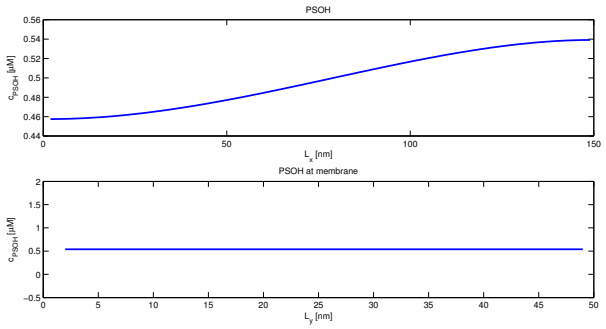


(d) Plot at iteration no.350000 and time $t = 9.45 ms$

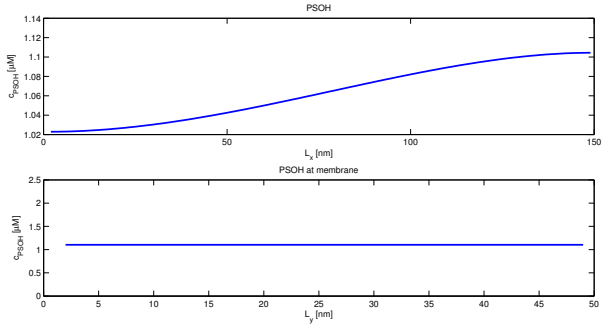
Figure 3.13: Plots of PSH concentration at cell fraction middle width ($L_y/2$) throughout the L_x dimension (above) and at membrane ($L_x - 1$) throughout the cell fraction width L_y (bottom).



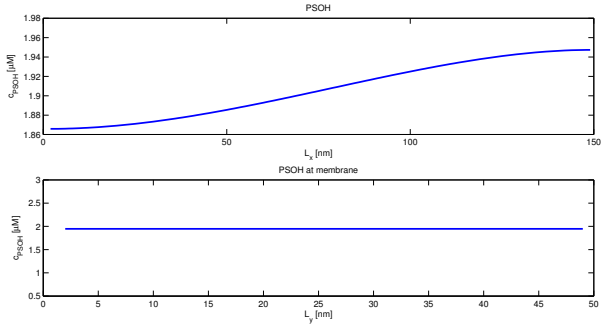
(a) Plot at iteration no.100 and time $t = 2.7 \cdot 10^{-3} ms$



(b) Plot at iteration no.100000 and time $t = 2.7 ms$

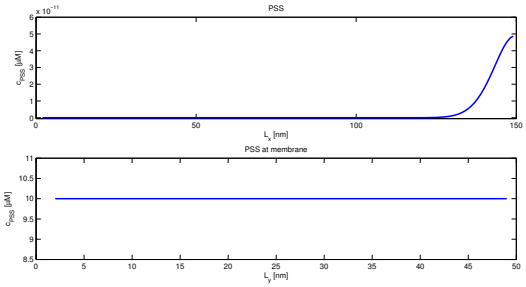


(c) Plot at iteration no.200000 and time $t = 5.4 ms$

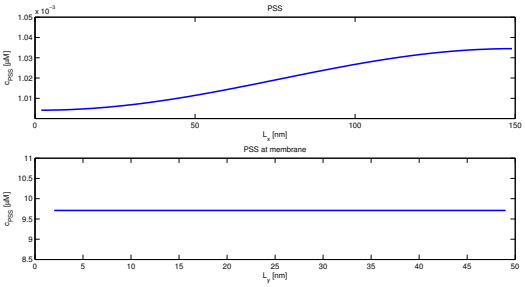


(d) Plot at iteration no.350000 and time $t = 9.45 ms$

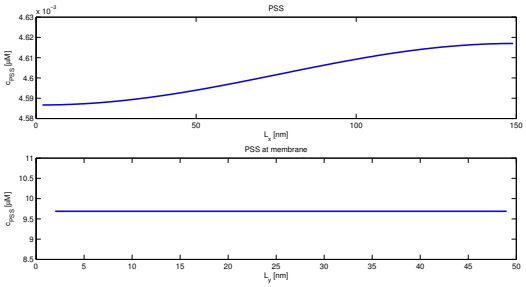
Figure 3.14: Plots of PSOH concentration at cell fraction middle width ($L_y/2$) throughout the L_x dimension (above) and at membrane ($L_x - 1$) throughout the cell fraction width L_y (bottom).



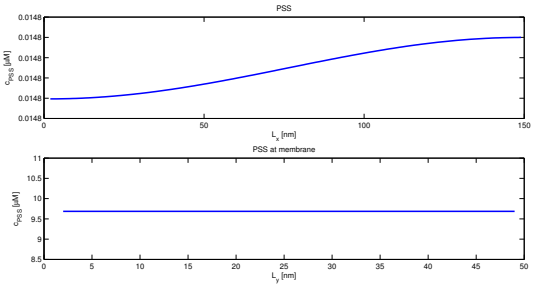
(a) Plot at iteration no.100 and time $t = 2.7 \cdot 10^{-3} ms$



(b) Plot at iteration no.100000 and time $t = 2.7 ms$

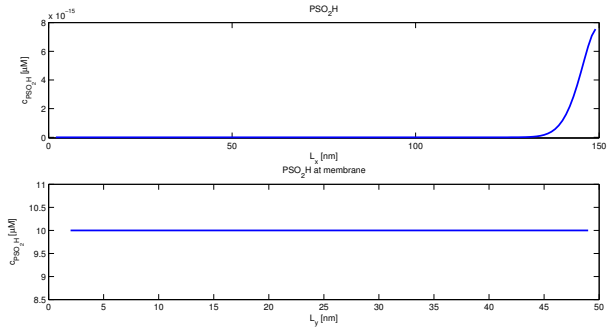


(c) Plot at iteration no.200000 and time $t = 5.4 ms$

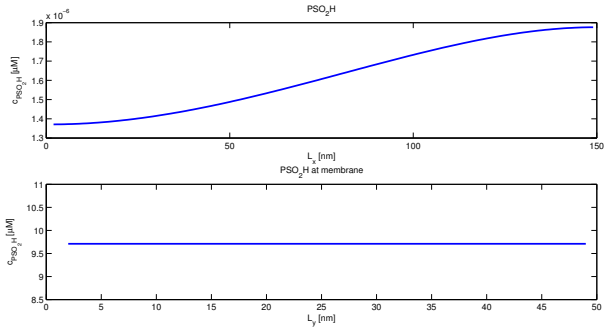


(d) Plot at iteration no.350000 and time $t = 9.45 ms$

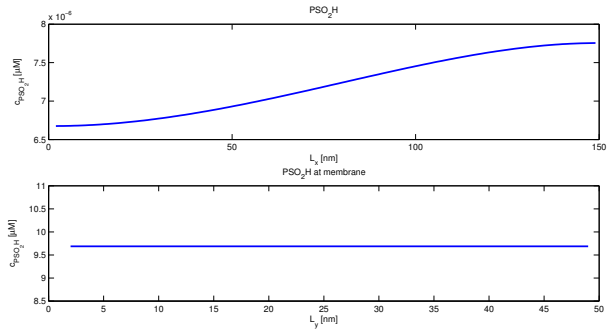
Figure 3.15: Plots of PSS concentration at cell fraction middle width $(L_y/2)$ throughout the L_x dimension (above) and at membrane $(L_x - 1)$ throughout the cell fraction width L_y (bottom).



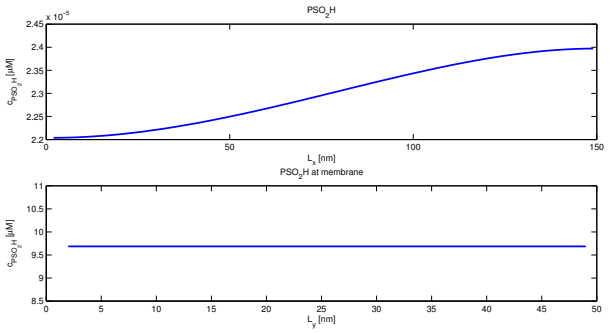
(a) Plot at iteration no.100 and time $t = 2.7 \cdot 10^{-3} ms$



(b) Plot at iteration no.100000 and time $t = 2.7 ms$



(c) Plot at iteration no.200000 and time $t = 5.4 ms$



(d) Plot at iteration no.350000 and time $t = 9.45 ms$

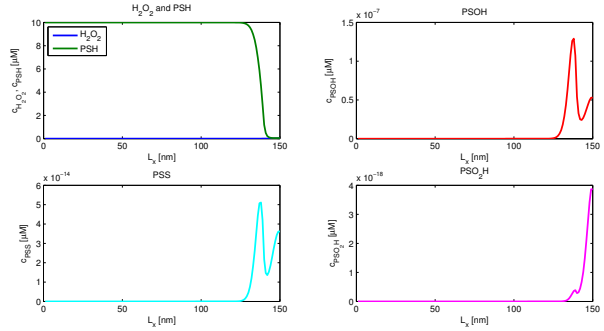
Figure 3.16: Plots of PSO_2H concentration at cell fraction middle width ($L_y/2$) throughout the L_x dimension (above) and at membrane ($L_x - 1$) throughout the cell fraction width L_y (bottom).

In this sense it is particularly interesting to observe that, as shown in figure 3.12, H_2O_2 enters the cell at $L_x - 1$ gradually progressing through the L_x dimension, although at lower concentrations because of the Prx2 intervention. PSH consumption is in fact evident at figure 3.13, in which its level, from the initially imposed costant value $c_{H_2O_2} = 10\ \mu M$, suddenly decreases approaching the membrane from the very early itarations (figure 3.13a). This trend continues further lowering PSH concentration (which reacts with H_2O_2) to quite the middle of the cell (figures 3.13b, 3.13c and 3.13d). On the other hand, PSOH, PSS and PSO_2H show an opposite progression, being produced by PSH action and thus gradually entering the cell (figures 3.14, 3.15 and 3.16 respectively).

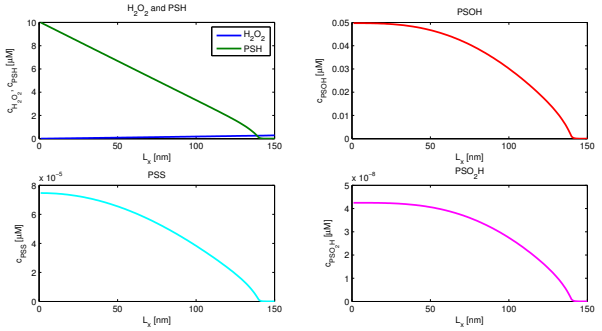
3.5 Monomeric peroxiredoxin II heterogeneous membrane model

The introduction of problem 5 in section 2.7 characterised the model by adding a receptor site within the membrane, thus emulating VEGF-R interactions with H_2O_2 and Prx2 monomers. As such, evident differences were expected in the graphics of concentrations, particularly at membrane level but also affecting the values of all the assessed components throughout the cell. That was only partially obtained, with minor changes in concentrations, as visible but not overwhelmingly in the density plots.

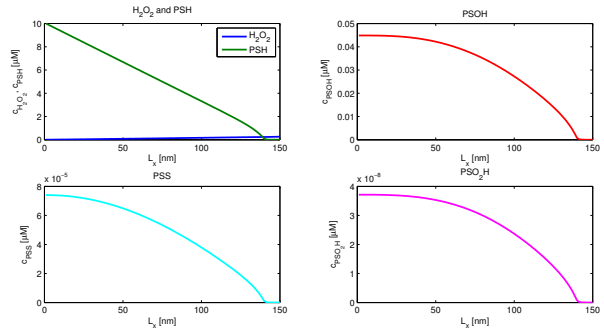
In this regard, it is interesting to firstly observe how figures in 3.10 from model 4 eventually present considerably higher concentration values of Prx2 monomers compared to those of figures in 3.17 from model 5. This is consistent with the presence of VEGF-R on the membrane in the latter problem, with an enhanced activity of Prx2 at membrane, particularly evident in the decreasing progression of PSH, PSOH, PSS and PSO_2H throughout L_x (figures 3.17a, 3.17b, 3.17c and 3.17d). H_2O_2 reduction is also slightly improved, as figure 3.17a suggests. The concentration of the parameters is in fact decreased in the neighborhood of the receptor, though this lowering is not overwhelming.



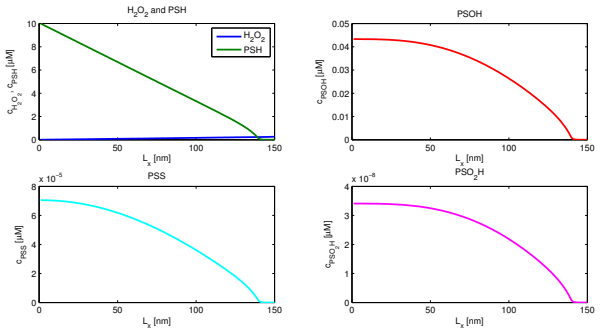
(a) Plot at iteration no.100 and time $t = 1.3 \cdot 10^{-3} ms$



(b) Plot at iteration no.250000 and time $t = 3.25 ms$

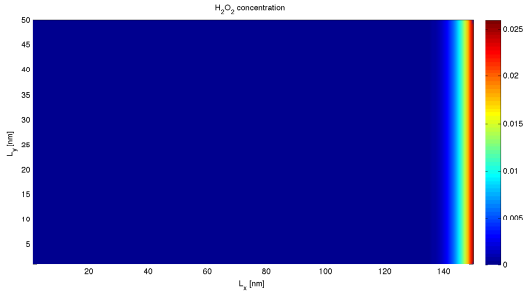


(c) Plot at iteration no.500000 and time $t = 6.5 ms$

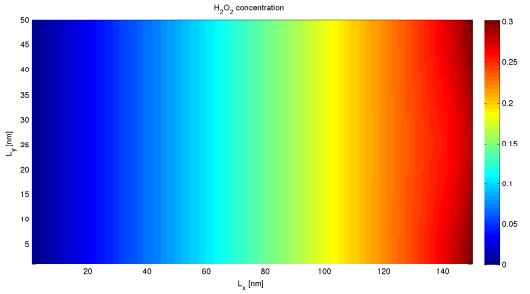


(d) Plot at iteration no.750000 and time $t = 9.75 ms$

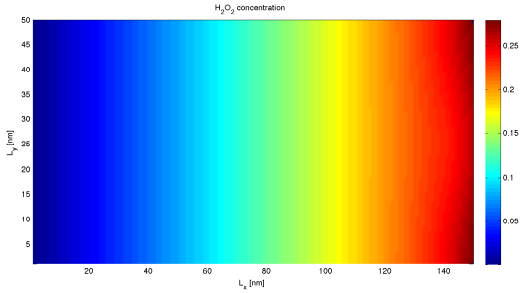
Figure 3.17: Plots of H_2O_2, PSH, PSOH, PSS, PSO_2H concentrations in the cell fraction ($0 < L_x < 150$ delining its intern and $L_x = 150$ the membrane).



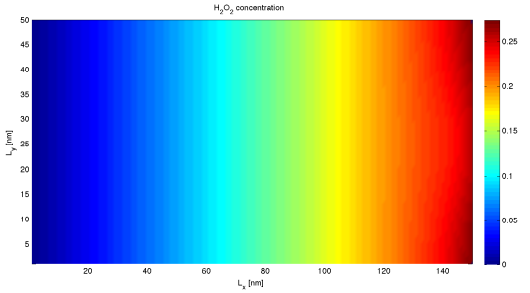
(a) Plot at iteration no.100 and time $t = 1.3 \cdot 10^{-3} ms$



(b) Plot at iteration no.250000 and time $t = 3.25 ms$

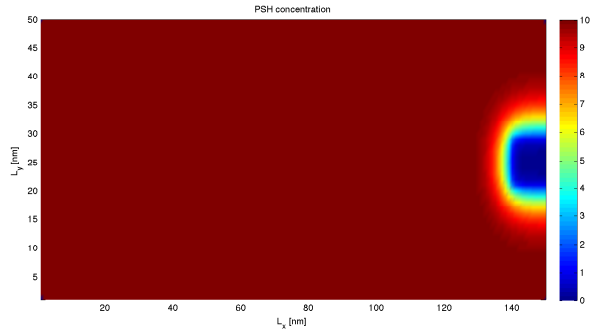


(c) Plot at iteration no.500000 and time $t = 6.5 ms$

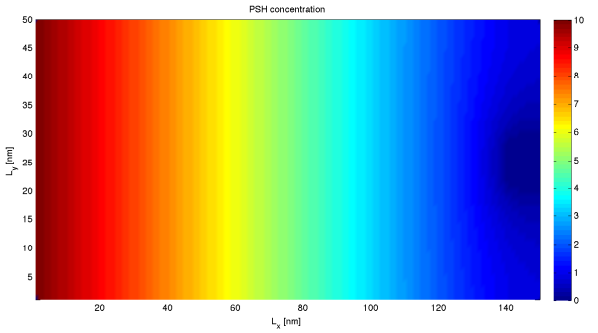


(d) Plot at iteration no.750000 and time $t = 9.75 ms$

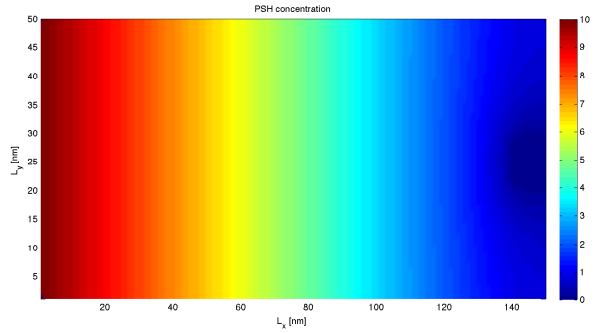
Figure 3.18: Density plots of H_2O_2 concentration in the cell fraction (from blue corresponding to zero to red for the highest concentration).



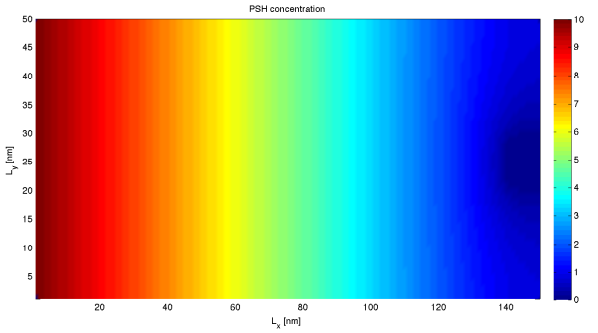
(a) Plot at iteration no.100 and time $t = 1.3 \cdot 10^{-3} ms$



(b) Plot at iteration no.250000 and time $t = 3.25 ms$

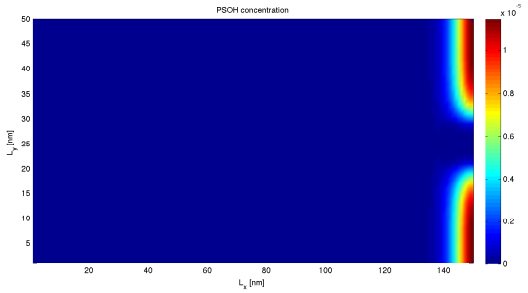


(c) Plot at iteration no.500000 and time $t = 6.5 ms$

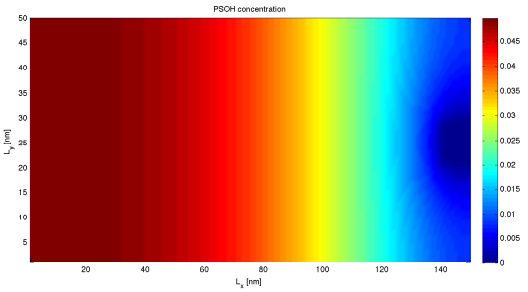


(d) Plot at iteration no.750000 and time $t = 9.75 ms$

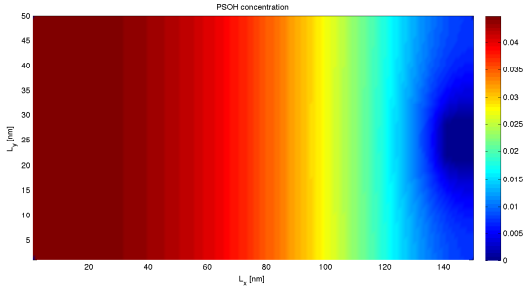
Figure 3.19: Density plots of PSH concentration in the cell fraction (from blue corresponding to zero to red for the highest concentration).



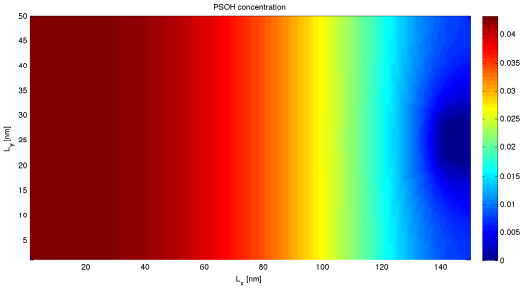
(a) Plot at iteration no.100 and time $t = 1.3 \cdot 10^{-3} ms$



(b) Plot at iteration no.250000 and time $t = 3.25 ms$

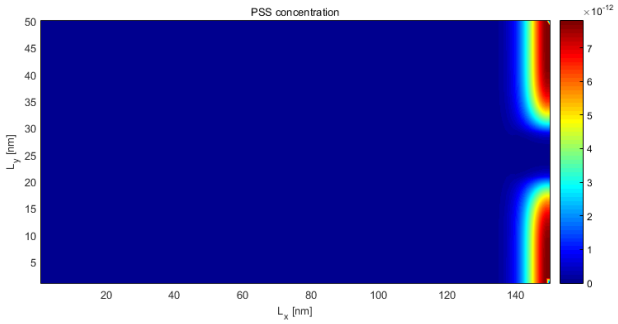


(c) Plot at iteration no.500000 and time $t = 6.5 ms$

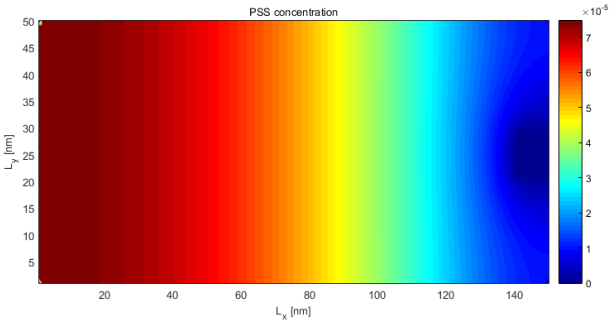


(d) Plot at iteration no.750000 and time $t = 9.75 ms$

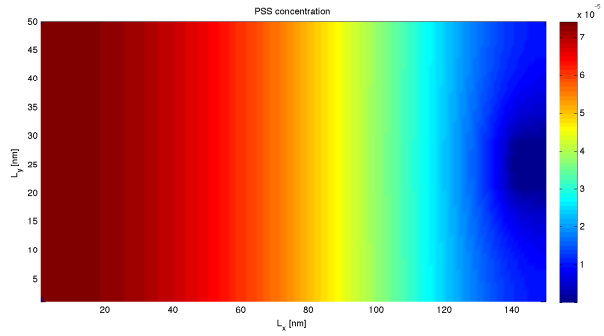
Figure 3.20: Density plots of PSOH concentration in the cell fraction (from blue corresponding to zero to red for the highest concentration).



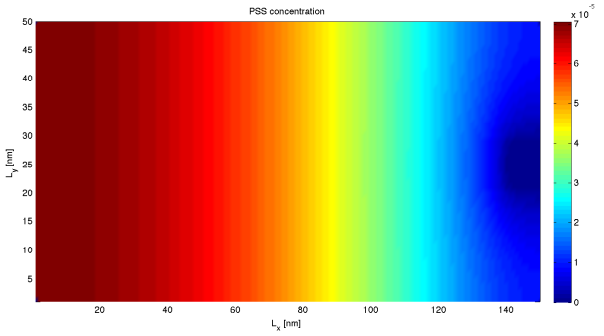
(a) Plot at iteration no.100 and time $t = 1.3 \cdot 10^{-3} ms$



(b) Plot at iteration no.250000 and time $t = 3.25 ms$

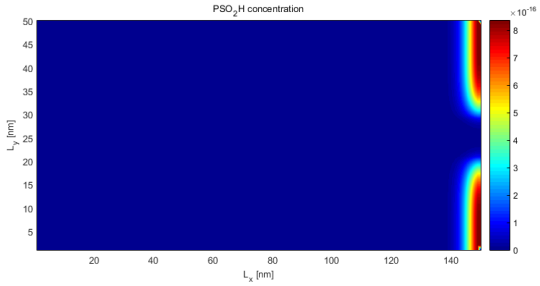


(c) Plot at iteration no.500000 and time $t = 6.5 ms$

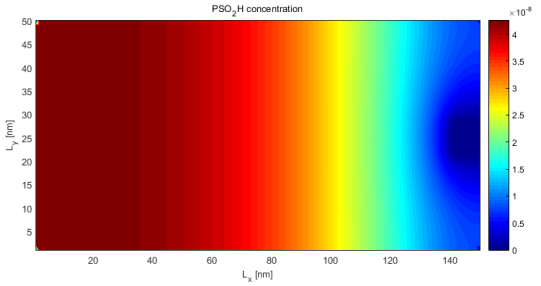


(d) Plot at iteration no.750000 and time $t = 9.75 ms$

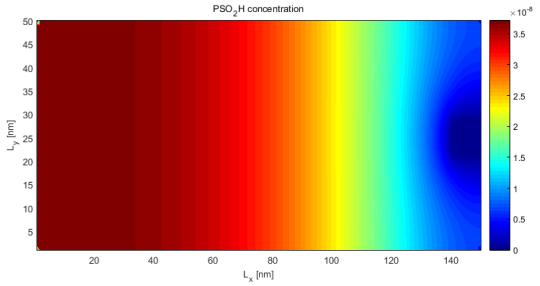
Figure 3.21: Density plots of PSS concentration in the cell fraction (from blue corresponding to zero to red for the highest concentration).



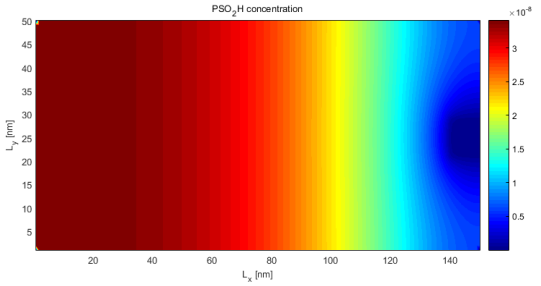
(a) Plot at iteration no.100 and time $t = 1.3 \cdot 10^{-3} ms$



(b) Plot at iteration no.250000 and time $t = 3.25 ms$

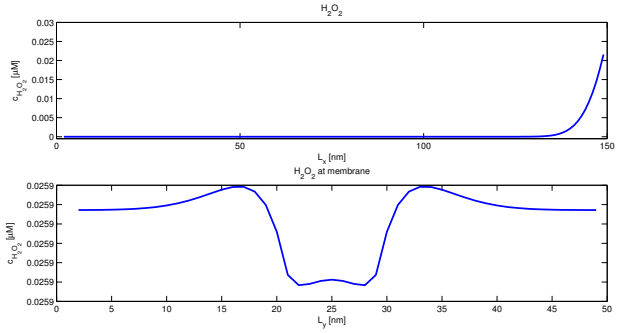


(c) Plot at iteration no.500000 and time $t = 6.5 ms$

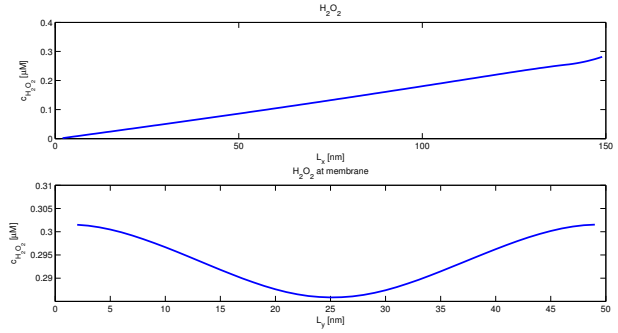


(d) Plot at iteration no.750000 and time $t = 9.75 ms$

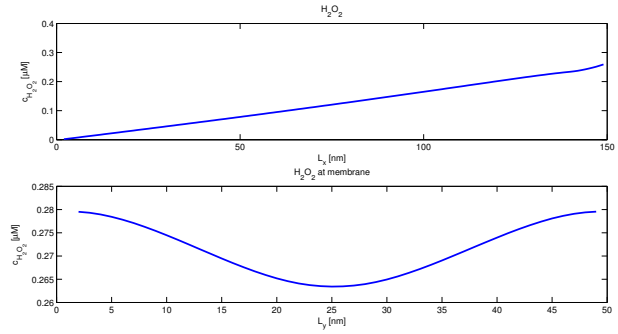
Figure 3.22: Density plots of PSO_2H concentration in the cell fraction (from blue corresponding to zero to red for the highest concentration).



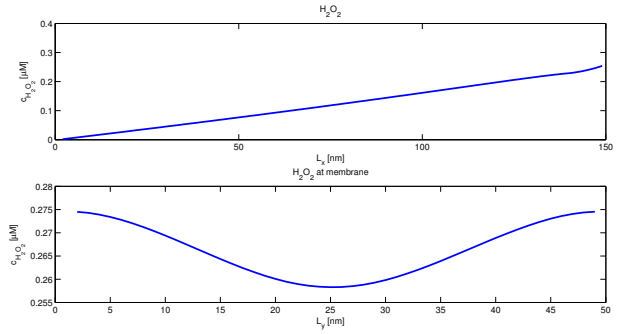
(a) Plot at iteration no.100 and time $t = 1.3 \cdot 10^{-3} ms$



(b) Plot at iteration no.250000 and time $t = 3.25 ms$

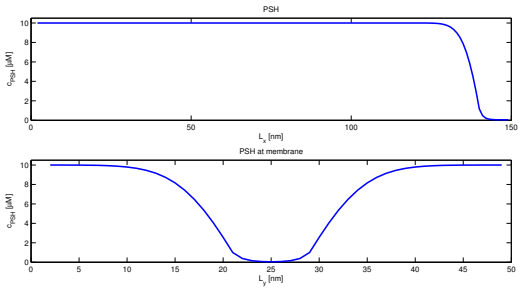


(c) Plot at iteration no.500000 and time $t = 6.5 ms$

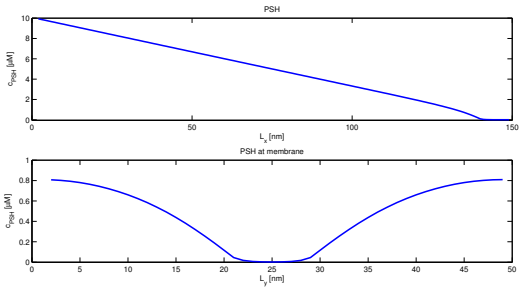


(d) Plot at iteration no.750000 and time $t = 9.75 ms$

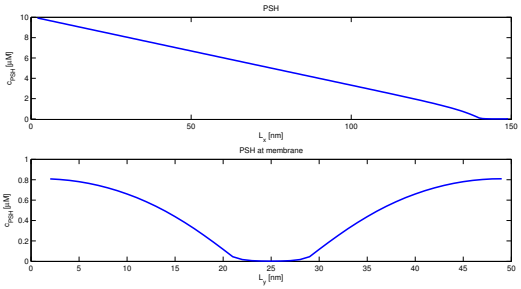
Figure 3.23: Plots of H_2O_2 concentration at cell fraction middle width ($L_y/2$) throughout the L_x dimension (above) and at membrane ($L_x - 1$) throughout the cell fraction width L_y (bottom).



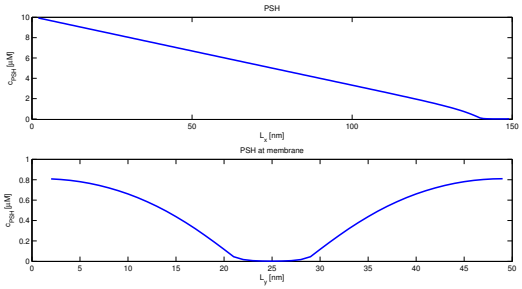
(a) Plot at iteration no.100 and time $t = 1.3 \cdot 10^{-3} ms$



(b) Plot at iteration no.250000 and time $t = 3.25 ms$

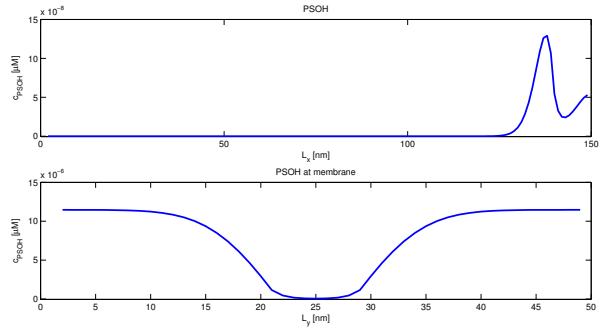


(c) Plot at iteration no.500000 and time $t = 6.5 ms$

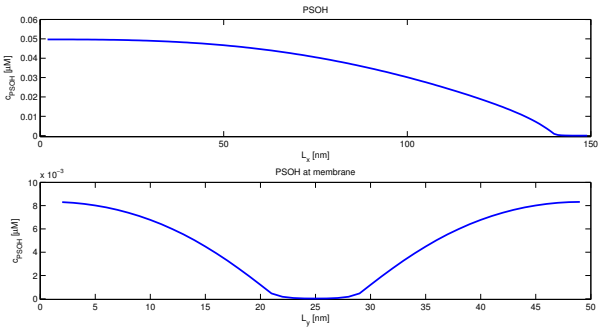


(d) Plot at iteration no.750000 and time $t = 9.75 ms$

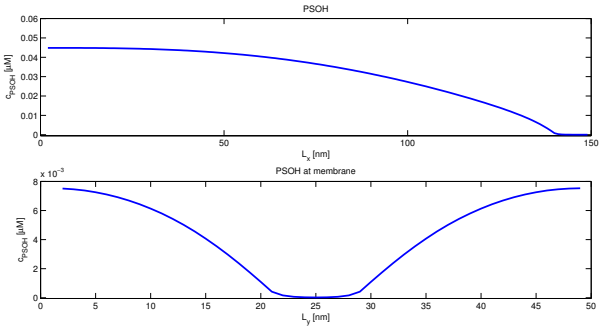
Figure 3.24: Plots of PSH concentration at cell fraction middle width $(L_y/2)$ throughout the L_x dimension (above) and at membrane $(L_x - 1)$ throughout the cell fraction width L_y (bottom).



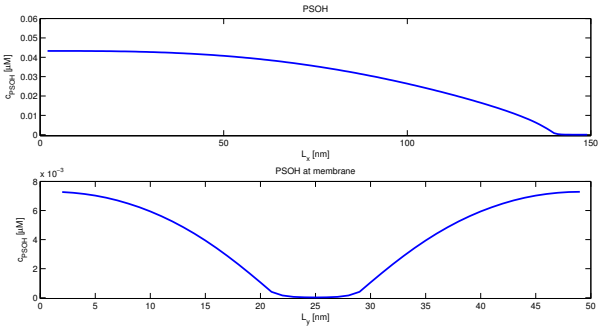
(a) Plot at iteration no.100 and time $t = 1.3 \cdot 10^{-3} ms$



(b) Plot at iteration no.250000 and time $t = 3.25 ms$

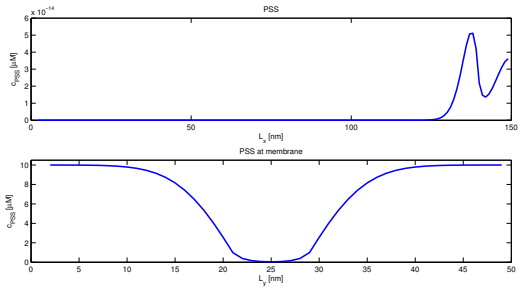


(c) Plot at iteration no.500000 and time $t = 6.5 ms$

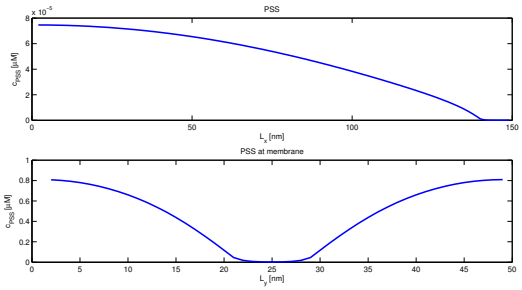


(d) Plot at iteration no.750000 and time $t = 9.75 ms$

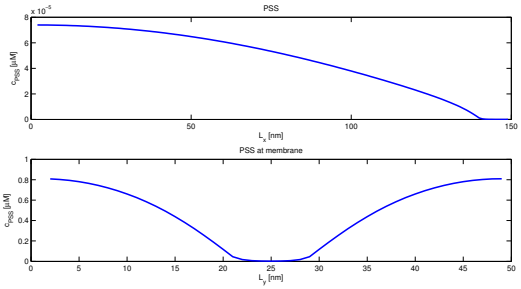
Figure 3.25: Plots of PSOH concentration at cell fraction middle width ($L_y/2$) throughout the L_x dimension (above) and at membrane ($L_x - 1$) throughout the cell fraction width L_y (bottom).



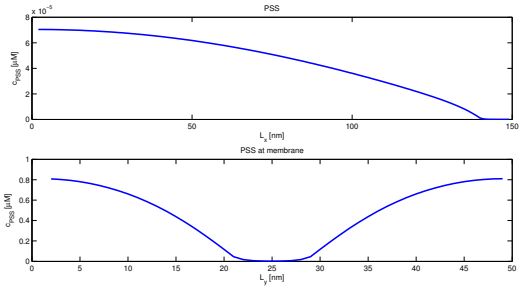
(a) Plot at iteration no.100 and time $t = 1.3 \cdot 10^{-3} ms$



(b) Plot at iteration no.250000 and time $t = 3.25 ms$

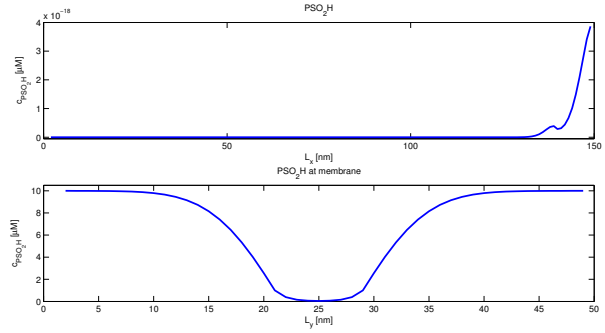


(c) Plot at iteration no.500000 and time $t = 6.5 ms$

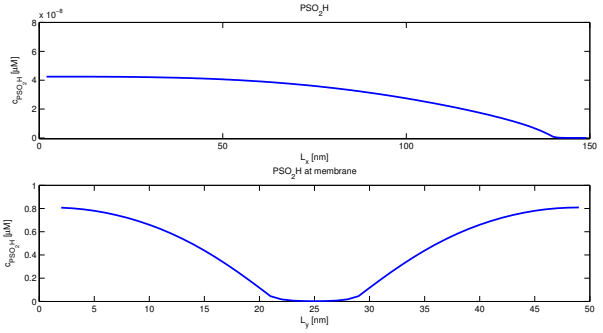


(d) Plot at iteration no.750000 and time $t = 9.75 ms$

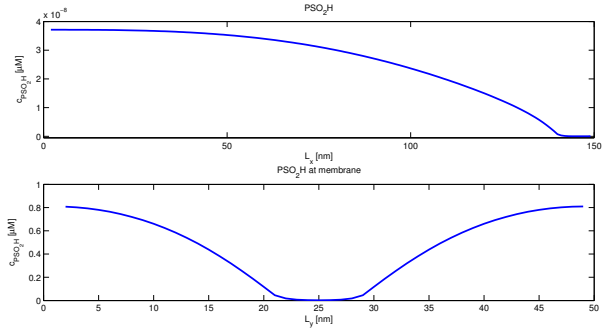
Figure 3.26: Plots of PSS concentration at cell fraction middle width $(L_y/2)$ throughout the L_x dimension (above) and at membrane $(L_x - 1)$ throughout the cell fraction width L_y (bottom).



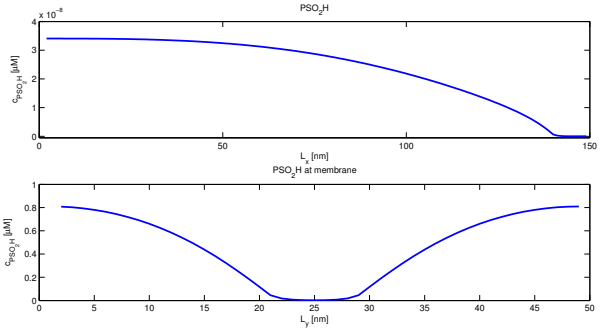
(a) Plot at iteration no.100 and time $t = 1.3 \cdot 10^{-3} ms$



(b) Plot at iteration no.250000 and time $t = 3.25 ms$



(c) Plot at iteration no.500000 and time $t = 6.5 ms$



(d) Plot at iteration no.750000 and time $t = 9.75 ms$

Figure 3.27: Plots of PSO_2H concentration at cell fraction middle width ($L_y/2$) throughout the L_x dimension (above) and at membrane ($L_x - 1$) throughout the cell width L_y (bottom).

Overall, the membrane localisation of the receptor provides PSOH withdrawal by PSH, from the centre to $L_x = 150\ nm$, causing a particularly low concentration in proximity of the membrane (PSOH is indeed completely localised at membrane at this stage). PSH is analogously concentrated within the receptor, but also presents a lower lateral diffusion. Their interactions and consequent reactions with H_2O_2 provoke its decrease at the receptor, leading to a smaller amount of hydrogen peroxide to actually enter in the cell, being mostly consumed at the membrane.

Observing the results obtained by modelling the system 2.12, a problem emerges in effectively simulating H_2O_2 reduction mechanisms. In particular, within the receptor localisation, PSH and PSOH concentrations are high only at the end of the L_x dimension, while in the receptor immediate proximity, H_2O_2 actually enters from L_x sides, because of the lack of Prx2 (as showed by figures 3.23). As such, within this model, the reduction of H_2O_2 levels achieved at receptor are in the scale of 10%. That is due to PSH and PSOH translocation at membrane, while in order to obtain a more effective reduction a reconsideration of the associaton and dissociation rates (k_{ass} and k_{diss} respectively) is needed.

As such, it can be concluded that this system succesfully simulates H_2O_2 concentration resulting from Prx2 interaction, when this protein is localised at the membrane. On the other hand, in order to reach more evident results, particularly in terms of concentration values, further studies comprising a precise characterisation of the system parameters are needed.

Chapter 4

Conclusions

In this work dynamic of Prx2 within the cellular system was studied, by modelling the interactions that this protein has with H_2O_2 at a membrane level, using PDEs.

In order to accurately model the biological context considered (as showed in fig. 2.1), a first set of problems were discussed and resolved, dealing with PDEs and analysing the main results through density plots.

While the basic model presented in section 2.5 already comprised the main components acting in the system, that are the concentrations of H_2O_2 and Prx2, the protein was considered in two simplified forms of active and inactive (Prx_I). Further detail to the model was introduced in section 2.6, in which Prx2 was considered also in its monomeric form, with equations for each monomer accounting of the scheme of reactions of fig. 1.1. The matlab codes that were written for this purpose allowed to observe the temporal evolution of all the different components acting in the system. A final improvement reflecting the actual setting of the biological system was brought out by assuming different Prx2 concentrations based on the protein localisation. In fact, while an internal source for the protein is included in the fluxes modelled in system 2.11, the main contribution comes from the Prx2 colocalised with VEGF-R2 at caveolae, that is at membrane level, in accordance with [6]. As such, the resolving code was opportunely modified to consider Prx2 translocation at membrane, within the receptor, in the modelled cell fraction. In this sense, the results throughout the various simulations proved that Prx2 translocation at membrane is actually fundamental in reducing H_2O_2 concentration within the receptor, though there is further margin to achieve improvements in terms of the amount of H_2O_2 consumption. From this point, the set of parameters yielding a higher reduction of H_2O_2 levels can be studied and determined, particularly by attempting a more efficient regulation of the association and dissociation rate values introduced in equations 2.12.

Furthermore, interesting considerations would also be assumed by ob-

serving the parameters at their stationary state, that is analysing the model components for a longer time, until the system reaches its equilibrium, while in this study the focus was on the analysis of the immediate state of the perturbed system, having performed the simulations in a time range of $[0-100]$ ms.

Appendices

Appendix A

Matlab codes

A.1 Code problem 1

Resolving code for magnetisation problem described by eq. (2.6):

```
%problem 1

clear all
close all
clc

L = 100;
dt = 0.1;
dx = 1;
r = dt/(dx*dx);
tmax = 100; %the lower the time, the greater the
   detail
nmax = tmax/dt;
nprint=10;

%initial condition
t=0;

f=zeros(L);

for i=2:L-1
    for j=2:L-1
        f(i,j)= 0.05*(rand()-0.5);
    end
end

%time iteration
```

```
for it=1:nmax
    g=zeros(L); %new matrix g() to include following
        times
    for i=2:L-1
        for j=2:L-1
            g(i,j)=f(i,j)+r*(f(i,j)-(f(i,j)^3)...
                +(f(i+1,j)+f(i-1,j)+f(i,j+1)...
                +f(i,j-1)-4*f(i,j)));
        end
    end
    %boundary conditions
    for j=2:L-1
        g(1,j)=g(L-1,j);
        g(L,j)=g(2,j);
    end
    for i=2:L-1
        g(i,1)=g(i,L-1);
        g(i,L)=g(i,2);
    end
    t=t+dt
    f=g;  %using matrix g as initial matrix in next
       iteration

    %plot
    if mod(it,nprint)==0
        pcolor(f)
        colormap(jet)
        caxis([-1 1])
        shading interp, colorbar
        xlabel('L_x'),ylabel('L_y')
        pause(0.1)
    end

    if it==5*nprint | it==nmax/10 | it==nmax/4 |...
            it==nmax
         saveas(gcf,[pwd,...
            '/prova/iteration no.',num2str(it),...
            ' and t=',num2str(t),' s.fig'])

    end

end
```

A.2 Code problem 2

Matlab code for eq. (2.7):

```
%problem 2a

clear all
close all
clc

L = 100;
dt = 0.1;
dx = 1;
r = dt/(dx*dx);
tmax = 100; %the lower the time, the greater the
   detail
nmax = tmax/dt;
nprint=10;

%initial condition
t=0;

f=zeros(L);

for i=2:L-1
    for j=2:L-1
        f(i,j)=exp(-((i-L/2)^2+(j-L/2)^2)/100);
    end
end

%time iteration
for it=1:nmax
    g=zeros(L); %new matrix g() to include following
        times
    for i=2:L-1
        for j=2:L-1
            g(i,j)=f(i,j)+r*((f(i+1,j)+f(i-1,j)...
                +f(i,j+1)+f(i,j-1)-4*f(i,j)));
        end
    end
    %boundary conditions
    for j=2:L-1
        g(1,j)=g(L-1,j);
        g(L,j)=g(2,j);
```

```
    end
    for i=2:L-1
        g(i,1)=g(i,L-1);
        g(i,L)=g(i,2);
    end
    t=t+dt;
    f=g;  %using matrix g as initial matrix in next
       iteration
    if mod(it,nprint)==0
        pcolor(f)
        colormap(jet)
        xlabel('L'),ylabel('L')
        shading interp, colorbar
        pause(0.1)
    end

    if it==5*nprint | it==nmax/4 | it==nmax
         saveas(gcf,[pwd,...
             '/prova1a/iteration no.',num2str(it),'
                and t=',num2str(t),' s.fig'])

    end
end
```

Matlab code for eq. (2.8):

```
%problem 2b

clear all
close all
clc

L = 100;
dt = 0.05;
dx = 1;
r = dt/(dx*dx);
tmax = 5; %the lower the time, the greater the
   detail
nmax = tmax/dt;
nprint=20;

%initial condition
t=0;

f1=zeros(L);
```

```
19
20 f2=zeros(L);
21
22 for i=2:L-1
23     for j=2:L-1
24         f1(i,j)=exp(-((i-L/2)^2+(j-L/2)^2)/100);
25         f2(i,j)=exp(-((i-L/3)^2+(j-L/3)^2)/50);
26     end
27 end
28
29 %time iteration
30 for it=1:nmax
31     g1=zeros(L); %new matrix g() to include
          following times
32     g2=zeros(L);
33     for i=2:L-1
34         for j=2:L-1
35             g1(i,j)=f1(i,j)+r*((5*f2(i,j)...
36                 +f1(i+1,j)+f1(i-1,j)+f1(i,j+1)...
37                 +f1(i,j-1)-4*f1(i,j)));
38             g2(i,j)=f2(i,j)+r*((-0.1*f2(i,j)...
39                 +f2(i+1,j)+f2(i-1,j)+f2(i,j+1)...
40                 +f2(i,j-1)-4*f2(i,j)));
41         end
42     end
43     %boundary conditions
44     for j=2:L-1
45         g1(1,j)=g1(L-1,j);
46         g1(L,j)=g1(2,j);
47
48         g2(1,j)=g2(L-1,j);
49         g2(L,j)=g2(2,j);
50     end
51     for i=2:L-1
52         g1(i,1)=g1(i,L-1);
53         g1(i,L)=g1(i,2);
54
55         g2(i,1)=g2(i,L-1);
56         g2(i,L)=g2(i,2);
57     end
58     t=t+dt
59     f1=g1;  %using matrix g as initial matrix in
          next iteration
60     f2=g2;
```

```
61
62      %plot1: m pcolor
63      if it==1 | it==40 | it==nmax
64      % if  mod(it-1,nprint)==0
65          figure(1)
66          pcolor(f1)
67          colormap(jet)
68          xlabel('L'),ylabel('L')
69          title('Heat diffusion of parameter m')
70          shading interp, colorbar
71          caxis([0 2])
72          pause(0.1)
73      end
74
75      if it==1 | it==40 | it==nmax
76           saveas(gcf,[pwd,...
77               '/prova1b/m iteration no.',num2str(it),
                    ' and t=',num2str(t),' s.fig'])
78
79      end
80
81      %plot2: n pcolor
82      if it==1 | it==40 | it==nmax
83      %if mod(it-1,nprint)==0
84          figure(2)
85          pcolor(f2)
86          colormap(jet)
87          xlabel('L'),ylabel('L')
88          title('Heat diffusion of parameter n')
89          shading interp, colorbar
90          caxis([0 1])
91          pause(0.1)
92      end
93
94      if it==1 | it==40 | it==nmax
95           saveas(gcf,[pwd,...
96               '/prova1b/n iteration no.',num2str(it),
                    ' and t=',num2str(t),' s.fig'])
97
98      end
99
100 end
```

Matlab code for eq. (2.9):

```
%problem 2c

clear all
close all
clc

L = 100;
dt = 0.1;
dx = 1;
dy=1;
dz=1;
r = dt/(dx*dx);
tmax = 100; %the lower the time, the greater the
   detail
nmax = tmax/dt;
nprint=10;

%initial condition
t=0;

f=zeros(L,L,L);

for i=2:L-1
    for j=2:L-1
        for k=2:L-1
            f(i,j,k)=exp(-((i-L/2)^2+(j-L/2)^2+(k-L
               /2)^2)/100);
        end
    end
end

%time iteration
for it=1:nmax
    g=zeros(L,L,L); %new matrix g() to include
       following times
    for i=2:L-1
        for j=2:L-1
            for k=2:L-1
                g(i,j,k)=f(i,j,k)+r*((f(i+dx,j,k)...
                    +f(i-dx,j,k)+f(i,j+dy,k)...
                    +f(i,j-dy,k)+f(i,j,k-dz)...
                    +f(i,j,k+dz)-6*f(i,j,k)));
            end
```

```
41          end
42      end
43      %boundary conditions
44      for j=2:L-1
45          for k=2:L-1
46              g(1,j,k)=g(L-1,j,k);
47              g(L,j,k)=g(2,j,k);
48          end
49      end
50
51      for i=2:L-1
52          for k=2:L-1
53              g(i,1,k)=g(i,L-1,k);
54              g(i,L,k)=g(i,2,k);
55          end
56      end
57
58      for i=2:L-1
59          for j=2:L-1
60             g(i,j,1)=g(i,j,L-1);
61             g(i,j,L)=g(i,j,2);
62          end
63      end
64
65      t=t+dt
66      f=g;  %using matrix g as initial matrix in next
           iteration
67
68      if mod(it,nprint)==0
69          pcolor(f(:,:,L/2))
70          colormap(jet)
71          caxis([-1 2])
72          shading interp, colorbar
73          xlabel('L_x'),ylabel('L_y')
74          pause(0.1)
75      end
76
77      if it==nprint | it==nmax/4 | it==nmax
78           saveas(gcf,[pwd,...
79               '/prova1c/iteration no.',num2str(it),'
                    and t=',num2str(t),' s.fig'])
80
81      end
82
```

```
end

save('data_problem2c.mat')
```

A.3 Code problem 3

Code resolving eq. (2.10):

```
%problem 3

clear all
close all
clc

Ly = 50;              %[nm]
Lx = 150;             %[nm]
dt = 1e-8;            %[s]
dx = 10e-9;           %[m]


%H2O2 parameters
D_h2o2=1.83e-9;       %[m^2/s]
dV=1.02e-14;          %[m^3/s]
A=1.3e-10;            %[m^2]
c0_h2o2=10;           %[uM], 0.01-100uM

%Prx2 parameters
k=1e6;                %[uM^-1 s^-1]
D_prx2=5e-9;          %[m^2/s]

r1=D_h2o2*dt/(dx*dx);
r2=D_prx2*dt/(dx*dx);

tmax = 1e-4; %the lower the time, the greater the
   detail
nmax = tmax/dt;
nprint=10;

%initial conditions
t=0;

f1=zeros(Ly,Lx);
f2=zeros(Ly,Lx);
```

```
35 f3=zeros(Ly,Lx);
36
37 %time iteration
38 for it=1:nmax
39     g1=zeros(Ly,Lx); %new matrix g() to include
          following times
40     g2=zeros(Ly,Lx);
41     g3=zeros(Ly,Lx);
42     for i=2:Ly-1
43         for j=2:Lx-1
44             g1(i,j)=f1(i,j)+r1*((f1(i+1,j)+f1(i-1,j)
                  +f1(i,j+1)+f1(i,j-1)-4*f1(i,j)))...
45                -k*dt*f1(i,j)*f2(i,j);
46             g2(i,j)=f2(i,j)+r2*((f2(i+1,j)+f2(i-1,j)
                  +f2(i,j+1)+f2(i,j-1)-4*f2(i,j)))...
47                -k*dt*f1(i,j)*f2(i,j);
48             g3(i,j)=f3(i,j)+r2*((f3(i+1,j)+f3(i-1,j)
                  +f3(i,j+1)+f3(i,j-1)-4*f3(i,j)))...
49                +k*dt*f1(i,j)*f2(i,j);
50         end
51     end
52
53     %boundary conditions
54
55     for j=2:Lx-1
56         g1(1,j)=g1(Ly-1,j);
57         g1(Ly,j)=g1(2,j);
58
59         g2(1,j)=g2(Ly-1,j);
60         g2(Ly,j)=g2(2,j);
61
62         g3(1,j)=g3(Ly-1,j);
63         g3(Ly,j)=g3(2,j);
64     end
65
66
67     for i=2:Ly-1
68         g1(i,1)=0;
69         g1(i,Lx)=g1(i,Lx-1)...
70             +dx*(1/D_h2o2*c0_h2o2*dV/A);
71
72         g2(i,1)=1;
73         g2(i,Lx)=g2(i,Lx-1);
74
```

```
75          g3(i,1)=g3(i,2);
76          g3(i,Lx)=g3(i,Lx-1);
77      end
78
79
80      t=t+dt
81      f1=g1;  %using matrix g as initial matrix in
            next iteration
82      f2=g2;
83      f3=g3;
84
85      %plot1: concentrations in time
86      if mod(it,nprint)==0
87          figure(1)
88          plot(1:150,100*f1(25,:),...
89              1:150,f2(25,:),...
90              1:150,100*f3(25,:),'g'),
91          title('H_2O_2,Prx2 and Prx2I')
92          legend('100*H2O2','Prx2','Prx2I')
93          ylabel(['c_{H_2O_2}, c_{Prx2},'...
94              'c_{Prx2I} [\muM]']),xlabel('L_x [nm]')
95          pause(0.1)
96      end
97
98      if mod(it,3e3)==0 | it==nprint | it==nmax
99          saveas(gcf,[pwd,...
100             '/prova3/iteration no.',num2str(it),...
101             ' and t=',num2str(t),' s.fig'])
102     end
103
104
105     %plot2: f1 pcolor
106     if mod(it,nprint)==0
107          figure(2)
108          pcolor(f1),shading interp, colorbar
109          xlabel('L_x [nm]'),ylabel('L_y [nm]')
110          title('H_2O_2 concentration')
111          pause(0.1)
112     end
113
114     if mod(it,3e3)==0 | it==nprint | it==nmax
115          saveas(gcf,[pwd,...
116              '/prova3/p1 iteration no.',num2str(it),
                    ' and t=',num2str(t),' s.fig'])
```

```
117
118     end
119
120     %plot3: f2 pcolor
121     if mod(it,nprint)==0
122          figure(3)
123          pcolor(f2),shading interp, colorbar
124          xlabel('L_x [nm]'),ylabel('L_y [nm]')
125          title('Prx2 concentration')
126          pause(0.1)
127     end
128
129     if mod(it,3e3)==0 | it==nprint | it==nmax
130          saveas(gcf,[pwd,...
131              '/prova3/p2 iteration no.',num2str(it),
                   ' and t=',num2str(t),' s.fig'])
132
133     end
134
135     %plot4: f3 pcolor
136     if mod(it,nprint)==0
137          figure(3)
138          pcolor(f3),shading interp, colorbar
139          caxis(caxis)
140          xlabel('L_x [nm]'),ylabel('L_y [nm]')
141          title('Prx2I concentration')
142          pause(0.1)
143     end
144
145     if mod(it,3e3)==0 | it==nprint | it==nmax
146          saveas(gcf,[pwd,...
147              '/prova3/p3 iteration no.',num2str(it),
                   ' and t=',num2str(t),' s.fig'])
148
149     end
150
151 end
152
153 save('data_problem3.mat')
```

A.4 Code problem 4

Resolving code for equations in system (2.11):

```
%problem 4

clear all
close all
clc

Lx = 150;            %[nm]
Ly = 50;             %[nm]
dt = 2.7e-8;         %[s]
dx = 10e-9;          %[m]


%H2O2 parameters
D_h2o2=1.8e-9;        %[m^2/s]
dV=1.02e-14;          %[m^3/s]
A=1.3e-10;            %[m^2]
c0prx2=10;
c0_h2o2=10;           %[uM], 0.01-100uM
k1=1e2;               %[uM^-1 s^-1]

%reaction parameters
k2=1.7;               %[s^-1]
k3=2.1e-2;            %[s^-1]
k4=1.2e-2;            %[uM^-1 s^-1]
k5=1e-4;              %[s^-1]
D=10e-10;             %[m^2/s]

r1=D_h2o2*dt/(dx*dx);
r2=D*dt/(dx*dx);

tmax = 10e-3; %the lower the time, the greater the
   detail
nmax = tmax/dt;
nprint=100;

%initial conditions
t=0;

f1=zeros(Ly,Lx);
f2=c0prx2*ones(Ly,Lx);
f3=zeros(Ly,Lx);
f4=zeros(Ly,Lx);
f5=zeros(Ly,Lx);
```

```
43
44 %time iteration
45 for it=1:nmax
46     g1=zeros(Ly,Lx); %new matrix g() to include
       following times
47     g2=zeros(Ly,Lx);
48     g3=zeros(Ly,Lx);
49     g4=zeros(Ly,Lx);
50     g5=zeros(Ly,Lx);
51     for i=2:Ly-1
52         for j=2:Lx-1
53             g1(i,j)=f1(i,j)+r1*((f1(i+1,j)...
54                 +f1(i-1,j)+f1(i,j+1)+f1(i,j-1)...
55                 -4*f1(i,j)))-k1*dt*f1(i,j)...
56                 *f2(i,j)-k4*dt*f1(i,j)*f3(i,j);
57             g2(i,j)=f2(i,j)+r2*((f2(i+1,j)...
58                 +f2(i-1,j)+f2(i,j+1)+f2(i,j-1)...
59                 -4*f2(i,j)))-k1*dt*f1(i,j)...
60                 *f2(i,j)+k3*dt*f4(i,j);
61             g3(i,j)=f3(i,j)+r2*((f3(i+1,j)...
62                 +f3(i-1,j)+f3(i,j+1)+f3(i,j-1)...
63                 -4*f3(i,j)))+k1*dt*f1(i,j)...
64                 *f2(i,j)-k4*dt*f1(i,j)*f3(i,j)...
65                 -k2*dt*f3(i,j)+k5*dt*f5(i,j);
66             g4(i,j)=f4(i,j)+r2*((f4(i+1,j)...
67                 +f4(i-1,j)+f4(i,j+1)+f4(i,j-1)...
68                 -4*f4(i,j)))+k2*dt*f3(i,j)...
69                 -k3*dt*f4(i,j);
70             g5(i,j)=f5(i,j)+r2*((f5(i+1,j)...
71                 +f5(i-1,j)+f5(i,j+1)+f5(i,j-1)...
72                 -4*f5(i,j)))+k4*dt*f3(i,j)...
73                 *f1(i,j)-k5*dt*f5(i,j);
74         end
75     end
76
77     %boundary conditions
78
79     for j=2:Lx-1
80         g1(1,j)=g1(Ly-1,j);
81         g1(Ly,j)=g1(2,j);
82
83         g2(1,j)=g2(Ly-1,j);
84         g2(Ly,j)=g2(2,j);
85
```

```
86          g3(1,j)=g3(Ly-1,j);
87          g3(Ly,j)=g3(2,j);
88
89          g4(1,j)=g4(Ly-1,j);
90          g4(Ly,j)=g4(2,j);
91
92          g5(1,j)=g5(Ly-1,j);
93          g5(Ly,j)=g5(2,j);
94      end
95
96
97      for i=2:Ly-1
98          g1(i,1)=0;
99          g1(i,Lx)=g1(i,Lx-1)...
100             +dx*(1/D_h2o2*c0_h2o2*dV/A);
101
102         g2(i,1)=c0prx2;
103         g2(i,Lx)=g2(i,Lx-1);
104
105         g3(i,1)=g3(i,2);
106         g3(i,Lx)=g3(i,Lx-1);
107
108         g4(i,1)=g4(i,2);
109         g4(i,Lx)=g4(i,Lx-1);
110
111         g5(i,1)=g5(i,2);
112         g5(i,Lx)=g5(i,Lx-1);
113     end
114
115
116     t=t+dt
117     f1=g1;  %using matrix g as initial matrix in
            next iteration
118     f2=g2;
119     f3=g3;
120     f4=g4;
121     f5=g5;
122
123     %plot1: concentrations in time
124     if mod(it,nprint)==0
125         figure(1)
126         subplot(2,2,1),plot(1:150,f1(25,:),...
127             1:150,f2(25,:)),
128         title('H_2O_2 and PSH')
```

```
129         legend('H_2O_2','PSH'),
130         ylabel('c_{H_2O_2}, c_{PSH} [\muM]')
131         xlabel('L_x [nm]')
132         subplot(2,2,2),plot(1:150,f3(25,:),'r'),
               title('PSOH'),
133         ylabel('c_{PSOH} [\muM]')
134         xlabel('L_x [nm]')
135         subplot(2,2,3),plot(1:150,f4(25,:),'c'),
               title('PSS'),
136         ylabel('c_{PSS} [\muM]')
137         xlabel('L_x [nm]')
138         subplot(2,2,4),plot(1:150,f5(25,:),'m'),
               title('PSO_2H'),
139         ylabel('c_{PSO_2H} [\muM]')
140         xlabel('L_x [nm]')
141         pause(0.1)
142     end
143
144
145     if mod(it,5e4)==0 | it==nprint | it==nmax
146         saveas(gcf,[pwd,...
147             '/prova4/plot1/iteration no.',...
148             num2str(it),' and t=',...
149             num2str(t),' s.fig'])
150     end
151
152
153     %plot2: f1 pcolor
154     if mod(it,nprint)==0
155          figure(2)
156          pcolor(f1),shading interp, colorbar
157          xlabel('L_x [nm]'),ylabel('L_y [nm]')
158          title('H_2O_2 concentration')
159          pause(0.1)
160     end
161
162     if mod(it,5e4)==0 | it==nprint | it==nmax
163          saveas(gcf,[pwd,...
164              '/prova4/plot2/iteration no.',...
165              num2str(it),' and t=',...
166              num2str(t),' s.fig'])
167
168     end
169
```

```
170
171     %plot3: f1 concentration
172     if mod(it,nprint)==0
173          figure(3)
174          subplot(2,1,1),plot(2:Lx-1,f1(Ly/2,2:Lx-1))
175          title('H_2O_2')
176          xlabel('L_x [nm]')
177          ylabel('c_{H_2O_2} [\muM]')
178          subplot(2,1,2),plot(2:Ly-1,f1(2:Ly-1,Lx)'),
                title('H_2O_2 at membrane')
179          xlabel('L_y [nm]')
180          ylabel('c_{H_2O_2} [\muM]')
181          pause(0.1)
182     end
183
184     if mod(it,5e4)==0 | it==nprint | it==nmax
185          saveas(gcf,[pwd,...
186              '/prova4/plot3/iteration no.',...
187              num2str(it),' and t=',...
188              num2str(t),' s.fig'])
189
190     end
191
192
193
194
195     %plot4: f2 concentration
196     if mod(it,nprint)==0
197          figure(4)
198          subplot(2,1,1),plot(2:Lx-1,f2(Ly/2,2:Lx-1))
199          title('PSH')
200          xlabel('L_x [nm]')
201          ylabel('c_{PSH} [\muM]')
202          subplot(2,1,2),plot(2:Ly-1,f2(2:Ly-1,Lx)'),
                title('PSH at membrane')
203          xlabel('L_y [nm]')
204          ylabel('c_{PSH} [\muM]')
205          pause(0.1)
206     end
207
208     if mod(it,5e4)==0 | it==nprint | it==nmax
209          saveas(gcf,[pwd,...
210              '/prova4/plot4/iteration no.',...
211              num2str(it),' and t=',...
```

```
212             num2str(t),' s.fig'])
213
214     end
215
216
217     %plot5: f3 concentration
218     if mod(it,nprint)==0
219         figure(5)
220         subplot(2,1,1),plot(2:Lx-1,f3(Ly/2,2:Lx-1))
221         title('PSOH')
222         xlabel('L_x [nm]')
223         ylabel('c_{PSOH} [\muM]')
224         subplot(2,1,2),plot(2:Ly-1,f3(2:Ly-1,Lx)'),
               title('PSOH at membrane')
225         xlabel('L_y [nm]')
226         ylabel('c_{PSOH} [\muM]')
227         pause(0.1)
228     end
229
230     if mod(it,5e4)==0 | it==nprint | it==nmax
231         saveas(gcf,[pwd,...
232             '/prova4/plot5/iteration no.',...
233             num2str(it),' and t=',...
234             num2str(t),' s.fig'])
235
236     end
237
238
239     %plot6: f4 concentration
240     if mod(it,nprint)==0
241         figure(6)
242         subplot(2,1,1),plot(2:Lx-1,f4(Ly/2,2:Lx-1))
243         title('PSS')
244         xlabel('L_x [nm]')
245         ylabel('c_{PSS} [\muM]')
246         subplot(2,1,2),plot(2:Ly-1,f2(2:Ly-1,Lx)'),
               title('PSS at membrane')
247         xlabel('L_y [nm]')
248         ylabel('c_{PSS} [\muM]')
249         pause(0.1)
250     end
251
252     if mod(it,5e4)==0 | it==nprint | it==nmax
253         saveas(gcf,[pwd,...
```

```
254                 '/prova4/plot6/iteration no.',...
255                 num2str(it),' and t=',...
256                 num2str(t),' s.fig'])
257
258     end
259
260
261     %plot7: f5 concentration
262     if mod(it,nprint)==0
263           figure(7)
264           subplot(2,1,1),plot(2:Lx-1,f5(Ly/2,2:Lx-1))
265           title('PSO_2H')
266           xlabel('L_x [nm]')
267           ylabel('c_{PSO_2H} [\muM]')
268           subplot(2,1,2),plot(2:Ly-1,f2(2:Ly-1,Lx)'),
                 title('PSO_2H at membrane')
269           xlabel('L_y [nm]')
270           ylabel('c_{PSO_2H} [\muM]')
271           pause(0.1)
272     end
273
274     if mod(it,5e4)==0 | it==nprint | it==nmax
275           saveas(gcf,[pwd,...
276               '/prova4/plot7/iteration no.',...
277               num2str(it),' and t=',...
278               num2str(t),' s.fig'])
279
280     end
281 end
282
283 save('data_problem4.mat')
```

A.5 Code problem 5

Resolving code for equations in system (2.12):

```
1 %problem 5
2
3 clear all
4 close all
5 clc
6
7 Lx = 150;            %[nm]
```

```
8  Ly = 50;              %[nm]
9  dt = 1.3e-8;          %[s]
10 dx = 10e-9;           %[m]
11
12
13 %H2O2 parameters
14 D_h2o2=1.8e-9;         %[m^2/s]
15 dV=1.02e-14;           %[m^3/s]
16 A=1.3e-10;             %[m^2]
17 c0prx2=10;             %[uM], 0.01-100uM
18 c0_h2o2=10;            %[uM], 0.01-100uM
19 k1=1e2;                %[uM^-1 s^-1]
20
21 %reaction parameters
22 k2=1.7;                %[s^-1]
23 k3=2.1e-2;             %[s^-1]
24 k4=1.2e-2;             %[uM^-1 s^-1]
25 k5=1e-4;               %[s^-1]
26 D=10e-10;              %[m^2/s]
27 Dm=10e-12;             %[m^2/s]
28
29 %membrane parameters
30 epsilon=10;
31 kass0=1e7;
32 kdis0=1e6;
33
34 r1=D_h2o2*dt/(dx*dx);
35 r2=D*dt/(dx*dx);
36 rm=Dm*dt/(dx*dx);
37
38 tmax = 10e-3; %the lower the time, the greater the
      detail
39 nmax = tmax/dt;
40 nprint=100;
41
42 %initial conditions
43 t=0;
44
45 f1=zeros(Ly,Lx);
46 f2=c0prx2*ones(Ly,Lx);
47 f3=zeros(Ly,Lx);
48 f4=zeros(Ly,Lx);
49 f5=zeros(Ly,Lx);
50
```

```
h2=zeros(Ly,1);
h3=zeros(Ly,1);
h4=zeros(Ly,1);
h5=zeros(Ly,1);

%time iteration
for it=1:nmax
    g1=zeros(Ly,Lx); %new matrix g() to include
       following times
    g2=zeros(Ly,Lx);
    g3=zeros(Ly,Lx);
    g4=zeros(Ly,Lx);
    g5=zeros(Ly,Lx);
    i2=zeros(Ly,1);
    i3=zeros(Ly,1);
    i4=zeros(Ly,1);
    i5=zeros(Ly,1);

    for i=2:Ly-1
        soma_k1=0;
        soma_ka2=0;
        soma_ka3=0;
        soma_ka4=0;
        soma_k4=0;
        soma_ka5=0;
        kass=0;
        kdis=0;

        if i>20 && i<30
            kass=kass0;
        end
        for j=2:Lx-1
            membrana=0;
            if j== Lx-1
                membrana=1;
            end
            delta=0;
            if j>=Lx-epsilon
                delta = 1;
                soma_k1=soma_k1+k1*h2(i)*f1(i,j);
                soma_ka2=soma_ka2+kass*f2(i,j);
                soma_ka3=soma_ka3+kass*f3(i,j);
                soma_ka4=soma_ka4+kass*f4(i,j);
                soma_k4=soma_k4+k4*h3(i)*f1(i,j);
```

```
94                  soma_ka5=soma_ka5+kass*f5(i,j);
95
96              end
97              g1(i,j)=f1(i,j)+r1*((f1(i+1,j)...
98                  +f1(i-1,j)+f1(i,j+1)+f1(i,j-1)...
99                  -4*f1(i,j)))-k1*dt*f1(i,j)...
100                 *f2(i,j)-k4*dt*f1(i,j)*f3(i,j)...
101                 +delta*(-k1/epsilon*dt*f1(i,j)...
102                 *h2(i)-k4/epsilon*dt*f1(i,j)*h3(i));
103             g2(i,j)=f2(i,j)+r2*((f2(i+1,j)...
104                 +f2(i-1,j)+f2(i,j+1)+f2(i,j-1)...
105                 -4*f2(i,j)))-k1*dt*f1(i,j)...
106                 *f2(i,j)+k3*dt*f4(i,j)+delta*dt...
107                 *(-kass*f2(i,j))...
108                 +kdis*dt*h2(i)/epsilon;
109             g3(i,j)=f3(i,j)+r2*((f3(i+1,j)...
110                 +f3(i-1,j)+f3(i,j+1)+f3(i,j-1)...
111                 -4*f3(i,j)))+k1*dt*f1(i,j)...
112                 *f2(i,j)-k4*dt*f1(i,j)*f3(i,j)...
113                 -k2*dt*f3(i,j)+k5*dt*f5(i,j)...
114                 +delta*dt*(-kass*f3(i,j))...
115                 +kdis*dt*h3(i)/epsilon;
116             g4(i,j)=f4(i,j)+r2*((f4(i+1,j)...
117                 +f4(i-1,j)+f4(i,j+1)+f4(i,j-1)...
118                 -4*f4(i,j)))+k2*dt*f3(i,j)...
119                 -k3*dt*f4(i,j)+delta*dt...
120                 *(-kass*f4(i,j))...
121                 +kdis*dt*h4(i)/epsilon;
122             g5(i,j)=f5(i,j)+r2*((f5(i+1,j)...
123                 +f5(i-1,j)+f5(i,j+1)+f5(i,j-1)...
124                 -4*f5(i,j)))+k4*dt*f3(i,j)...
125                 *f1(i,j)-k5*dt*f5(i,j)+delta*dt...
126                 *(-kass*f5(i,j))...
127                 +kdis*dt*h5(i)/epsilon;
128         end
129         i2(i)=h2(i)+rm*(h2(i+1)+h2(i-1)-2*h2(i))...
130             -kdis*dt*h2(i)+soma_ka2*dt...
131             -soma_k1/epsilon*dt+k3*h4(i)*dt;
132         i3(i)=h3(i)+rm*(h3(i+1)+h3(i-1)-2*h3(i))...
133             -kdis*dt*h3(i)+soma_ka3*dt...
134             -k2*dt*h3(i)+k5*dt*h5(i)...
135             +soma_k1/epsilon*dt-soma_ka4/epsilon*dt;
136         i4(i)=h4(i)+rm*(h4(i+1)+h4(i-1)-2*h4(i))...
137             -kdis*dt*h4(i)+soma_ka4*dt...
```

```
138             +k2*dt*h3(i)-k3*dt*h4(i);
139         i5(i)=h5(i)+rm*(h5(i+1)+h5(i-1)-2*h5(i))...
140             -kdis*dt*h5(i)+soma_ka5*dt...
141             -k5*dt*h5(i)+soma_k4/epsilon*dt;
142
143     end;
144
145     %boundary conditions
146
147     for j=2:Lx-1
148         g1(1,j)=g1(Ly-1,j);
149         g1(Ly,j)=g1(2,j);
150
151         g2(1,j)=g2(Ly-1,j);
152         g2(Ly,j)=g2(2,j);
153
154         g3(1,j)=g3(Ly-1,j);
155         g3(Ly,j)=g3(2,j);
156
157         g4(1,j)=g4(Ly-1,j);
158         g4(Ly,j)=g4(2,j);
159
160         g5(1,j)=g5(Ly-1,j);
161         g5(Ly,j)=g5(2,j);
162     end
163
164
165     for i=2:Ly-1
166         g1(i,1)=0;
167         g1(i,Lx)=g1(i,Lx-1)...
168             +dx*(1/D_h2o2*c0_h2o2*dV/A);
169
170         g2(i,1)=c0prx2;
171         g2(i,Lx)=g2(i,Lx-1);
172
173         g3(i,1)=g3(i,2);
174         g3(i,Lx)=g3(i,Lx-1);
175
176         g4(i,1)=g4(i,2);
177         g4(i,Lx)=g4(i,Lx-1);
178
179         g5(i,1)=g5(i,2);
180         g5(i,Lx)=g5(i,Lx-1);
181     end
```

```
182
183
184     t=t+dt
185     f1=g1;  %using matrix g as initial matrix in
            next iteration
186     f2=g2;
187     f3=g3;
188     f4=g4;
189     f5=g5;
190     h2=i2;
191     h3=i3;
192     h4=i4;
193     h5=i5;
194
195
196     %plot1: concentrations in time
197     if mod(it,nprint)==0
198         figure(1)
199         subplot(2,2,1),plot(1:150,f1(25,:),...
200             1:150,f2(25,:)),
201         title('H_2O_2 and PSH')
202         legend('H_2O_2','PSH'),
203         ylabel('c_{H_2O_2}, c_{PSH} [\muM]')
204         xlabel('L_x [nm]')
205         subplot(2,2,2),plot(1:150,f3(25,:),'r'),
                title('PSOH'),
206         ylabel('c_{PSOH} [\muM]')
207         xlabel('L_x [nm]')
208         subplot(2,2,3),plot(1:150,f4(25,:),'c'),
                title('PSS'),
209         ylabel('c_{PSS} [\muM]')
210         xlabel('L_x [nm]')
211         subplot(2,2,4),plot(1:150,f5(25,:),'m'),
                title('PSO_2H'),
212         ylabel('c_{PSO_2H} [\muM]')
213         xlabel('L_x [nm]')
214         pause(0.1);
215     end
216
217     if mod(it,5e4)==0 | it==nprint | it==nmax
218         saveas(gcf,[pwd,...
219             '/prova5/plot1/iteration no.',...
220             num2str(it),' and t=',...
221             num2str(t),' s.fig'])
```

```
222     end
223
224
225
226     %plot2: f1 concentration
227     if mod(it,nprint)==0
228          figure(2)
229          subplot(2,1,1),plot(2:Lx-1,f1(Ly/2,2:Lx-1))
230          title('H_2O_2')
231          xlabel('L_x [nm]')
232          ylabel('c_{H_2O_2} [\muM]')
233          subplot(2,1,2),plot(2:Ly-1,f1(2:Ly-1,Lx)'),
                title('H_2O_2 at membrane')
234          xlabel('L_y [nm]')
235          ylabel('c_{H_2O_2} [\muM]')
236          pause(0.1);
237     end
238
239     if mod(it,5e4)==0 | it==nprint | it==nmax
240          saveas(gcf,[pwd,...
241              '/prova5/plot2/iteration no.',...
242              num2str(it),' and t=',...
243              num2str(t),' s.fig'])
244     end
245
246
247     %plot3: f2 concentration
248     if mod(it,nprint)==0
249          figure(3)
250          subplot(2,1,1),plot(2:Lx-1,f2(Ly/2,2:Lx-1))
251          title('PSH')
252          xlabel('L_x [nm]')
253          ylabel('c_{PSH} [\muM]')
254          subplot(2,1,2),plot(2:Ly-1,f2(2:Ly-1,Lx)'),
                title('PSH at membrane')
255          xlabel('L_y [nm]')
256          ylabel('c_{PSH} [\muM]')
257          pause(0.1);
258     end
259
260     if mod(it,5e4)==0 | it==nprint | it==nmax
261          saveas(gcf,[pwd,...
262              '/prova5/plot3/iteration no.',...
263              num2str(it),' and t=',...
```

```
264             num2str(t),' s.fig'])
265     end
266
267
268     %plot4: f3 concentration
269     if mod(it,nprint)==0
270         figure(4)
271         subplot(2,1,1),plot(2:Lx-1,f3(Ly/2,2:Lx-1))
272         title('PSOH')
273         xlabel('L_x [nm]')
274         ylabel('c_{PSOH} [\muM]')
275         subplot(2,1,2),plot(2:Ly-1,f3(2:Ly-1,Lx)'),
                title('PSOH at membrane')
276         xlabel('L_y [nm]')
277         ylabel('c_{PSOH} [\muM]')
278         pause(0.1);
279     end
280
281     if mod(it,5e4)==0 | it==nprint | it==nmax
282         saveas(gcf,[pwd,...
283             '/prova5/plot4/iteration no.',...
284             num2str(it),' and t=',...
285             num2str(t),' s.fig'])
286     end
287
288
289     %plot5: f4 concentration
290     if mod(it,nprint)==0
291         figure(5)
292         subplot(2,1,1),plot(2:Lx-1,f4(Ly/2,2:Lx-1))
293         title('PSS')
294         xlabel('L_x [nm]')
295         ylabel('c_{PSS} [\muM]')
296         subplot(2,1,2),plot(2:Ly-1,f2(2:Ly-1,Lx)'),
                title('PSS at membrane')
297         xlabel('L_y [nm]')
298         ylabel('c_{PSS} [\muM]')
299         pause(0.1);
300     end
301
302     if mod(it,5e4)==0 | it==nprint | it==nmax
303         saveas(gcf,[pwd,...
304             '/prova5/plot5/iteration no.',...
305             num2str(it),' and t=',...
```

```
306             num2str(t),' s.fig'])
307     end
308
309
310     %plot6: f5 concentration
311     if mod(it,nprint)==0
312         figure(6)
313         subplot(2,1,1),plot(2:Lx-1,f5(Ly/2,2:Lx-1))
314         title('PSO_2H')
315         xlabel('L_x [nm]')
316         ylabel('c_{PSO_2H} [\muM]')
317         subplot(2,1,2),plot(2:Ly-1,f2(2:Ly-1,Lx)'),
                title('PSO_2H at membrane')
318         xlabel('L_y [nm]')
319         ylabel('c_{PSO_2H} [\muM]')
320         pause(0.1);
321     end
322
323     if mod(it,5e4)==0 | it==nprint | it==nmax
324         saveas(gcf,[pwd,...
325             '/prova5/plot6/iteration no.',...
326             num2str(it),' and t=',...
327             num2str(t),' s.fig'])
328     end
329
330
331     %f1 pcolor
332     if mod(it,nprint)==0
333         figure(7)
334         pcolor(f1),shading interp,colorbar
335         xlabel('L_x [nm]'),ylabel('L_y [nm]')
336         title('H_2O_2 concentration')
337         pause(0.1);
338     end
339
340     if mod(it,5e4)==0 | it==nprint | it==nmax
341         saveas(gcf,[pwd,...
342             '/prova5/pcolor/f1/iteration no.',...
343             num2str(it),' and t=',...
344             num2str(t),' s.fig'])
345     end
346
347
348     %f2 pcolor
```

```
349     if mod(it,nprint)==0
350         figure(8)
351         pcolor(f2),shading interp,colorbar
352         xlabel('L_x [nm]'),ylabel('L_y [nm]')
353         title('PSH concentration')
354         pause(0.1);
355     end
356 
357     if mod(it,5e4)==0 | it==nprint | it==nmax
358         saveas(gcf,[pwd,...
359             '/prova5/pcolor/f2/iteration no.',...
360             num2str(it),' and t=',...
361             num2str(t),' s.fig'])
362     end
363 
364 
365     %f3 pcolor
366     if mod(it,nprint)==0
367         figure(9)
368         pcolor(f3),shading interp,colorbar
369         xlabel('L_x [nm]'),ylabel('L_y [nm]')
370         title('PSOH concentration')
371         pause(0.1);
372     end
373 
374     if mod(it,5e4)==0 | it==nprint | it==nmax
375         saveas(gcf,[pwd,...
376             '/prova5/pcolor/f3/iteration no.',...
377             num2str(it),' and t=',...
378             num2str(t),' s.fig'])
379     end
380 
381 
382 
383     %f4 pcolor
384     if mod(it,nprint)==0
385         figure(10)
386         pcolor(f4),shading interp,colorbar
387         xlabel('L_x [nm]'),ylabel('L_y [nm]')
388         title('PSS concentration')
389         pause(0.1);
390     end
391 
392     if mod(it,5e4)==0 | it==nprint | it==nmax
```

```
393           saveas(gcf,[pwd,...
394               '/prova5/pcolor/f4/iteration no.',...
395               num2str(it),' and t=',...
396               num2str(t),' s.fig'])
397       end
398
399
400       %f5 pcolor
401       if mod(it,nprint)==0
402           figure(11)
403           pcolor(f5),shading interp,colorbar
404           xlabel('L_x [nm]'),ylabel('L_y [nm]')
405           title('PSO_2H concentration')
406           pause(0.1);
407       end
408
409       if mod(it,5e4)==0 | it==nprint | it==nmax
410           saveas(gcf,[pwd,...
411               '/prova5/pcolor/f5/iteration no.',...
412               num2str(it),' and t=',...
413               num2str(t),' s.fig'])
414       end
415
416   end
417
418   save('data_problem5.mat')
```

Bibliography

[1] J. Dulak, A. Jozkowicz, and A. Loboda. *Angiogenesis and Vascularisation. Cellular and Molecular Mechanisms in Health and Diseases.* Springer, 2013.

[2] T. H. Adair and J. P. Montani. *Angiogenesis.* Morgan & Claypool Life Sciences, 2011.

[3] A. Hoeben et al. Vascular Endothelial Growth Factor and Angiogenesis. *Pharmacological Reviews*, 56(4), 2004.

[4] J. L. Mehta and N. S. Dhalla. *Biochemical Basis and Therapeutic Implications of Angiogenesis.* 2013.

[5] C. A. Reinhart-King. *Mechanical and Chemical Signaling in Angiogenesis.* Springer, 2013.

[6] D. H. Kang et al. Peroxiredoxin II Is an Essential Antioxidant Enzyme that Prevents the Oxidative Inactivation of VEGF Receptor-2 in Vascular Endothelial Cells. *Molecular Cell*, 44.

[7] R. M. Johnson, G. Goyette, Y. Ravindranath, and Y. S. Ho. Hemoglobin Autoxidation and Regulation of endogenous H_2O_2 Levels in Erythrocytes. *Free Radical Biology & Medicine*, 39, 2005.

[8] R. Benfeitas, G. Selvaggio, F. Antunes, P. Coelho, and A. Salvador. Hydrogen Peroxide Metabolism and Sensing in Human Erythrocytes: a Validated Kinetic Model and Reappraisal of the Role of Peroxiredoxin II. *Free Radical Biology & Medicine*, 74, 2014.

[9] L. W. Clegg & F. M. Gabhann. Site Specific Phosphorylationof VEGFR2 is Mediated by Receptor Trafficking: Insights from Computational Model. *PLOS Computational Biology*, 11(6), 2015.

[10] A. Chillà et al. Proteomic Identification of VEGF-dependent Protein Enrichment to Membrane Caveolar-raft Microdomains in Endothelial Progenitor Cells. *Molecular and Cellular Proteomics*, 12(7), 2013.

[11] F. M. Low, M. B. Hampton, and C. C. Winterbourn. Peroxiredoxin 2 and Peroxide Metabolism in the Erythrocyte. *Antioxidants & Redox Singaling*, 10(9), 2008.

[12] A. J. Bray. Theory of Phase Ordering Kinetics. *Advances in Physics*, 43(3), 1994.

[13] Q. Han. *A Basic Course in Partial Differential Equations*. American Mathematical Society, 2011.

[14] Z. Rubinstein. *A Course in Ordinary and Partial Differential Equations*. Academic Press, 1969.

[15] Y. Pinchover and J. Rubinstein. *An Introduction to Partial Differential Equations*. Cambridge University Press, 2005.

[16] P. L. DeVries. *A First Course in Computational Physics*. John Wiley & Sons, 1994.

[17] M. Claus & G. Breier. *Mechanisms of Angiogenesis*. Birkhauser Verlag, 2005.

[18] S. A. Mousa & P. J. Davis. *Angiogenesis Modulation in Health and Disease*. Springer, 2013.

[19] I. Domingues et al. VEGFR2 Translocates to the Nucleus to Regulate its own Transcription. *PLOS One*, 6(9), 2011.