

ADAPTIVE NUMERICAL SIMULATION OF REACTION - DIFFUSION SYSTEMS

Chamakuri Nagaiah



Faculty of Mathematics
Otto-von-Guericke University, Magdeburg.

ADAPTIVE NUMERICAL SIMULATION OF REACTION - DIFFUSION SYSTEMS

Dissertation

zur Erlangung des akademischen Grades

Doktor rerum naturalium

(Dr. rer. nat.)

von: M. Tech., M. Sc., Chamakuri Nagaiah

geb. am: 03.06.1979 in Maddirala, Andhra Pradesh, India

genehmigt durch die Fakultät für Mathematik

der Otto-von-Guericke Universität, Magdeburg, Deutschland

Gutachter: Prof. Dr. Gerald Warnecke

Priv.-Doz. Dr. Martin Falcke

Eingereicht am: 08.01.2007

Verteidigung am: 30.03.2007

ADAPTIVE NUMERICAL SIMULATION OF REACTION - DIFFUSION SYSTEMS

Dissertation

Submitted for the academic degree

Doktor rerum naturalium

(Dr. rer. nat)

by: M. Tech.,M. Sc., Chamakuri Nagaiah

born on : 03.06.1979 in Maddirala, Andhra Pradesh, India

approved from the Faculty of Mathematics

Otto-von-Guericke University, Magdeburg, Germany

Referees: Prof. Dr. Gerald Warnecke

Priv.-Doz. Dr. Martin Falcke

Submitted on: 08.01.2007

Defended on: 30.03.2007

To my parents

Table of Contents

Table of Contents	vii
Abstract	ix
1 Introduction	1
2 Mathematical Modeling of Heat and Mass Transfer in Fluidized Beds	7
2.1 Introduction of basic variables and assumptions	10
2.2 Balance equations	13
2.2.1 Boundary conditions	20
2.3 Two dimensional model for the liquid injection into the fluidized bed . . .	22
2.3.1 Spraying mass balance equation	25
2.4 Three dimensional model for the liquid injection into the fluidized bed . . .	26
2.4.1 Spraying mass balance equation	28
2.5 Invariant regions	29
3 Model Equations in Intracellular Calcium Dynamics	35
3.1 Introduction to intracellular Ca^{2+} dynamics	37
3.2 Governing equations	39
3.2.1 Deterministic equations in 2D	39
3.2.2 Deterministic equations in 3D	40
3.3 Stochastic behavior of intracellular Ca^{2+} dynamics	42
3.3.1 Stochastic channel model	43
3.4 Hybrid stochastic and deterministic model	47
4 Discretization of Reaction-Diffusion Systems	51
4.1 Mathematical notations and function spaces	51
4.2 The basic aspects of the finite element method	54
4.2.1 Mass lumping	57
4.3 Spatial discretization	58
4.3.1 Semi discretization in space	58

4.4	Time discretization	61
4.4.1	One step ϑ schemes	61
4.4.2	Linearly implicit Runge-Kutta methods	62
4.5	Grid adaptivity and error estimators	64
4.5.1	The Z^2 error indicator	66
4.6	Solution of algebraic equations	70
4.6.1	Inexact Newton method	71
4.6.2	Solution of linear equations	72
4.7	Domain decomposition methods	74
5	Numerical Results of Heat and Mass Transfer in Fluidized Beds	77
5.1	Numerical results in 1D and 2D	77
5.1.1	One-dimensional simulation results with uniform liquid distribution	77
5.1.2	Two-dimensional simulation results with non-uniform liquid distribution	89
5.2	Numerical results in 3D	97
5.2.1	Experiment-1	98
5.2.2	Experiment-2	102
5.3	Parallel numerical results	107
6	Numerical Results for the Intracellular Ca^{2+} Dynamics	111
6.1	Numerical results in 2D	111
6.1.1	Opening of one channel deterministically in one cluster	112
6.1.2	Numerical results of the stochastic channel transition in one cluster	121
6.1.3	Numerical results with many clusters	123
6.2	Numerical results in 3D	129
6.3	Numerical results using domain decomposition methods	136
7	Summary	139
A	Simulation Parameters for Heat and Mass Transfer in Fluidized Beds	143
B	Hybrid Algorithm and Simulation Parameters used for Ca^{2+} Dynamics	151
	Bibliography	155

Abstract

The aim of this work is to find efficient and reliable numerical solutions of two complex problems under consideration. In the first application problem, an improved continuum model has been derived to describe the temperature and concentration distributions in gas-solid-fluidized beds with spray injection. The model equations for the nozzle spray are also reformulated to achieve reliable numerical solutions. The model equations are strongly coupled and semilinear partial differential equations with boundary conditions. The model equations are more flexible to compute the numerical solution on unstructured meshes than previous models. Solutions to these equations are approximated using a finite element method for the spatial discretization and an implicit Euler method in time. A study has been conducted to see the behavior of process parameters for heat and mass transfer in fluidized beds. The numerical results demonstrate that the method has a convergence order that agrees with theoretical considerations. The numerical results are validated with experimental results for two cases in three space dimensions. From parallelized numerical results, using domain decomposition methods, we show that good parallel efficiency is achieved with different numbers of processors.

The second application problem concerns the adaptive numerical simulation of intracellular calcium dynamics. The modeling of diffusion, binding and membrane transport of calcium ions in cells leads to a system of reaction-diffusion equations. The strongly localized temporal behavior of calcium concentration due to opening and closing of channels as well as their spatial localization are effectively treated by an adaptive finite element method. The discrete approximation of deterministic equations produces a system of stiff ordinary differential equations with multiple time scales. The time scales are handled using linearly implicit time stepping methods with an adaptive step size control. The opening and closing of channels is typically a stochastic process. A hybrid method is adopted to couple the deterministic and stochastic equations. The adaptive numerical convergence of solutions is studied with different cluster arrangements. The deterministic equations are solved with parallel numerical methods to reduce the computational time using domain decomposition methods. A good parallel efficiency is achieved with different numbers of processors.

Zusammenfassung

Gegenstand der vorliegenden Arbeit ist es, effiziente und zuverlässige numerische Lösungen für zwei betrachtete komplexe Probleme zu finden. Für das erste Anwendungsproblem wurde zur Beschreibung der Temperatur und der Konzentrationsverteilung in Gas/Feststoff-Wirbelschichten mit Einspritzdüse ein verbessertes Kontinuumsmodell hergeleitet. Um zuverlässige numerische Lösungen zu erzielen, wurden die Modellgleichungen für die Spritzdüse neu dargelegt. Sie sind gekoppelte, nichtlineare partielle Differentialgleichungen mit Randbedingungen. Diese Modellgleichungen sind flexibler bei der Berechnung numerischer Lösungen auf unstrukturierten Gittern. Sie wurden mittels einer Finiten Element-Methode für die Ortsdiskretisierung und des impliziten Eulerverfahrens für die Zeitdiskretisierung approximiert. Ferner wurde eine Studie zur Untersuchung des Verhaltens der Prozessparameter des Massen- und Wärmeaustausches in Wirbelschichten durchgeführt. Die numerischen Resultate zeigen, dass die Konvergenzordnung der verwendeten Methode mit theoretischen Betrachtungen übereinstimmt. Durch experimentelle Daten für zwei Fälle in drei Raumdimensionen wurden die numerischen Ergebnisse bestätigt. Unter Verwendung von Gebietszerlegungsmethoden konnte für parallele Rechnungen mit unterschiedlicher Anzahl von Prozessoren eine gute Effizienz erzielt werden.

Das zweite Anwendungsproblem beschäftigt sich mit der adaptiven numerischen Simulation der intrazellularen Dynamik von Kalzium. Die Modellierung der Diffusion, der Bindung sowie des Membrantransports der Kalziumionen in den Zellen führt auf ein System von Reaktions-Diffusionsgleichungen. Das streng lokalisierte temporäre Verhalten der Kalziumkonzentration aufgrund des Öffnens und Schließens von Kanälen einerseits sowie ihre räumliche Lokalisierung andererseits sind effektiv mit einer adaptiven Finiten Element-Methode behandelt worden. Die diskrete Approximierung der deterministischen Gleichungen ergibt ein System gewöhnlicher Differentialgleichungen mit mehreren Zeitskalen. Unter Verwendung linear impliziter Zeitschrittverfahren mit adaptiver Schrittweitensteuerung wurden diese Zeitskalen behandelt. Das Öffnen und Schließen der Kanäle ist typischerweise ein stochastischer Prozess. Zur Kopplung der deterministischen und stochastischen Gleichungen wurde eine gemischte Methode eingesetzt. Die numerische Konvergenz der

Lösungen ist ausführlich mit verschiedenen Klustereinteilungen untersucht worden. Die deterministischen Gleichungen wurden mit parallelen numerischen Methoden unter Verwendung von Gebietszerlegungstechniken gelöst, um die benötigte Rechenzeit zu reduzieren. Eine hohe Effizienz konnte mit unterschiedlicher Anzahl von Prozessoren erreicht werden.

Acknowledgement

I would like to express my sincere and deep gratitude to my pioneering supervisor Prof. Dr. Gerald Warnecke, without his guidance this work would not have been accomplished. I am greatly indebted to him for providing me with advice and encouragement during the course of my research. He was always open to new ideas and I appreciate the freedom he gave me while working on my research.

I would like to thank Jun.-Prof. Dr.-Ing. Stefan Heinrich, Dr.-Ing. Mirko Peglow and Prof. Dr.-Ing. Lothar Mörl for their support during my graduate work in modeling of heat and mass transfer in fluidized beds.

My sincere thanks to Priv.-Doz. Dr. Martin Falcke for giving me the opportunity to work in the area of cell biology and his continuous help. In particular, working in interdisciplinary areas I have enjoyed the joint work with Dr. Sten Rüdiger, who gave me the excellent insight into the real cell biology. I have benefited immensely from his help to get the right perspective.

I wish to thank Prof. S. Sundar for his encouragement. I also like to thank the faculty and staff of the Institute for Analysis and Numerics and those who helped to make this work successful. Thanks also to Dr. Walfred Grambow for his help in technical assistance.

I would like to acknowledge the German Research Foundation (Deutsche Forschungsgemeinschaft) for providing funding through research programm Graduiertenkolleg 828 and under the project grants "Wa 633/16-1, Fa 350/6-1". I would also like to thank all the members of the Graduiertenkolleg.

Last but not least, my whole hearted gratitude to my family for their immeasurable support throughout my life without their unconditional love, encouragement and support, I could never have come this far. And also my special thanks to all of my friends for their moral support.

Chapter 1

Introduction

This work is devoted to the reliable solution of specific complex problems described by partial differential equations of advection, diffusion and reaction type. They are explained in Chapter 2 and Chapter 3. The problems under consideration originate from various processes of nature from biology and of chemical engineering. In this work, our focus is on both fields. The two specific problems, which are considered in the present work, have similar type of mathematical model equations. Due to the high complexity of such systems, finding exact solutions is currently unknown. Therefore, numerical methods are essential to obtain the approximate solution of such problems. Furthermore, the computational capabilities allow the incorporation of more and more detailed physics into the models. The problems considered in this work are highly coupled and it is challenging to get the numerical solution with moderate computational time.

In this work, the attention is focused on two specific problems. The first problem is described by the heat and mass transfer in fluidized beds with liquid injection. This process is very complex in reality and the modeling of such a process is described by the mass and the energy balances of the particles, the sprayed liquid, as well as air. The basic modeling of this problem can be found in the theses of Heinrich [47] and also Henneberg [51]. The focus of the current study is on the derivation of a continuum model and the numerical treatment of such a system of coupled semilinear parabolic and hyperbolic equations, especially for fluidized bed spray granulation (FBSG), to determine the thermal conditions in the fluidized bed. This work contains an improvement of the modeling and a more sophisticated numerical method is presented in comparison to the model presented in Heinrich [49]. The model equations form a complicated system of coupled semilinear partial differential equations. The model equations for the nozzle spray in two and three dimensions are reformulated using the available previous models. The model equations are

made mesh independent in order to be more flexible with different meshes. The previously used structured finite difference method is replaced by a finite element method allowing unstructured grids. This flexibility allows the computation of a fluidized bed apparatus with non rectangular geometry as well as local mesh adaption in the solution process. If we include more than one spray nozzle in three space dimensions then the numerical computations with a single processor are very difficult. This simulation may take several days or months. Due to the high computational cost, parallel programming methods are used to reduce the computational time for such type of complex applications. Specifically, the domain decomposition methods are used to obtain accurate and efficient results.

The second application problem is concerned with the efficient and reliable numerical solutions of intracellular calcium dynamics. In the calcium dynamics, one of the important task of the cellular information processes is the calcium signalling task. Specific signals can trigger a sudden increase in the cytoplasmic calcium level by opening channels in the endoplasmic reticulum. The calcium released by one channel diffuses into the cytosol. This diffusivity increases the opening probability of neighboring channels. This way of coupling of channels by calcium diffusion causes the spatial spread of release. The opening and closing of channels occurs in microseconds. This process leads to complicated spatio-temporal signals in the cell. Due to fluctuations in binding and unbinding of IP_3 and Ca^{2+} , the opening and closing of channels is stochastic. Here, the most important task is to understand this complicated signalling task and to develop accurate models as well as models for the stochastic channel transitions. This type of models can be found in Gillespie [37], DeYoung-Keizer [29], Dixon et al. [30], Falcke [34], Thul and Falcke [90], and Thul [89]. The current work addresses the numerical treatment of such processes. The simulation of a single event and several events leads to many challenging numerical problems.

Here we will outline some of the problems which are encountered in the numerical simulations. At first, due to the multiple length scales of channels and clusters in the membrane, suitable numerical methods are mandatory. Especially, finite element methods or finite volume methods seem to be suitable. In this work we have chosen the finite element method for solving this problem. To capture the original structure of the cell, adaptive grid refinement is necessary to provide efficient and fast numerical solutions. The adaptive grid refinement uses a posteriori error estimators, which will be mentioned in next paragraph. The high and fast concentration changes upon opening and closing of channels have a strong impact on the stochastic dynamics of channel binding and unbinding. The stochastic solver is based on the Gillespie method, but the usual Gillespie method solves

stochastic processes where the propensities are constant during the subsequent transitions. However, the concentration and propensities are changes based on the channel opening and closing. For this purpose we have adopted a new hybrid algorithm which couples the deterministic and stochastic equations, see Alfonsi et al. [2]. Two different time stepping methods are considered for solving the deterministic and stochastic processes. The release of calcium through channel opening or closing occurs on the order of microseconds. These small time scales cannot be ignored, therefore an efficient time stepping method to capture these fast changes are needed. For this purpose the linearly implicit Runge-Kutta methods, which are very suitable for solving stiff ordinary differential equations, are used. The opening of channels occurs in order of microseconds and when all channels are closed then the time step size is nearly in order of milliseconds. For handling such fast changing step sizes an automatic time step procedure is suitable.

For solving the problems with complicated geometries, as explained above, the finite element methods are more suitable than finite difference methods. The current work uses the standard Galerkin method for the spatial discretization. A system of ordinary differential equations is obtained after space discretization. We have mainly concentrated on implicit methods for solving these equations. In the first problem, we considered the implicit Euler method for solving the system of ordinary differential equations. After time discretization we get a system of nonlinear equations which can be solved using nonlinear solvers. In the second problem we used linearly implicit methods of Rosenbrock type [79] which are constructed by calculating the exact Jacobian. These methods offer several advantages. They completely avoid the solution of nonlinear equations, which means that no Newton iteration has to be controlled. More detailed expositions of these methods can be found in Hairer and Wanner [45], Lang [57], as well as Schmitt and Weiner [82]. Also these methods are more suitable to use adaptive time steps. An automatic step size selection procedure ensures that the step size is as large as possible while guaranteeing the desired precision.

A posteriori error estimates can be used to judge the quality of a numerical approximation and to determine an adaptive strategy to improve the accuracy where it is needed, see Babuška and Rheinboldt [3], Verfürth [93], as well as Zienkiewicz and Zhu [98]. The main advantage of the adaptive methods is that they lead to substantial saving in computational work for a given tolerance. Here we have used the error estimators based on the averaging of the solution. The large class of problems in three space dimensions include million of unknowns in computational domain. On sequential machines these computations

are tedious and may take several months to get a numerical solution. Specifically, solving the algebraic system consumes much CPU time in computations. For this purpose parallel computing is mandatory. Even with the use of modern parallel computers, the sheer size of the resulting system puts too much demand on the available capacity in terms of memory usage and computational time in three dimensional problems.

The remaining chapters are organized as follows.

Chapter 2 gives a brief explanation of the mathematical modeling of heat and mass transfer in fluidized beds. The convection-diffusion-reaction equations are obtained with the help of the mass and the energy balances of the air, particles and liquid. These equations are coupled and form a semilinear system with boundary conditions. Next, the nozzle spray model is derived using some assumptions which are explained in Section 2.3 for two and three dimensions. This is an improved model and more sophisticated numerical methods are applied in comparison to Heinrich et al. [48], Heinrich and Mörl [49]. The modeling is made mesh independent in order to be more flexible with different meshes. This flexibility allows the computation of an apparatus with nonrectangular geometry as well as local mesh adaption in the solution process. Finally, necessary conditions are derived for the balance equations to check the solution bounds using invariant regions.

A brief introduction of the process of intracellular calcium dynamics is given in Chapter 3. The model equations in two and three dimensions are shown in Section 3.2, which have been taken from Falcke [34], Thul and Falcke [90], and Thul [89]. These model equations in the cytosolic and the endoplasmic regions represent a reaction-diffusion system. These equations form a highly coupled and semilinear system. A stochastic model has been adopted for the gating of subunits which is explained in Section 3.3. This model is based on the DeYoung-Keizer model for the subunit dynamics, see [29]. A brief explanation of this model is also given in this chapter. Furthermore, an efficient method, the so-called Gillespie algorithm is explained, which determines the time of each transition by using random numbers. Finally, a hybrid method is presented which couples the deterministic and the stochastic equations.

In Chapter 4 we focus on the discretization of the partial differential equations in space and time. First, a brief introduction of the basic finite element methods and the space discretization using the standard Galerkin method is given. In the next section, the time discretizations based on the single step ϑ -method and linearly implicit methods are explained. In general, implicit methods are more suitable for solving stiff ordinary differential equations. Specifically, linearly implicit methods are very suitable for adaptive

time step size control. For the space adaptivity a posteriori error estimators are essential which are explained in Section 4.5. In particular, the error estimator of Zienkiewicz and Zhu [98] is recalled which is based on the average of the local gradients of the solution. In the next section, an overview of the solution of algebraic equations is given. Finally, the basic concepts of domain decomposition methods are explained. They are employed in parallel numerical simulations using the programming package UG [12].

The numerical solutions of heat and mass transfer in fluidized beds with liquid injection are presented in Chapter 5. The simulation results of complex correlations of mass and liquid flow rates, mass and heat transfer, drying, transient two-dimensional air humidity, air temperature, particle wetting, liquid film temperature and particle temperature are shown in the first section. Comparisons of simulation results with different linear solvers and experimental order of convergence are given. In Section 5.2, the three dimensional numerical solutions of air humidity, air temperature, degree of wetting, liquid film temperature and particle temperature are depicted with different degree of net depositions. The simulation results have been validated with transient measurement result of the air temperature which are shown in the work of Heinrich [47] and Henneberg [51]. The numerical solutions using domain decomposition methods are discussed in the final section.

The adaptive numerical solutions of intracellular calcium dynamics with different numbers of cluster arrangements are presented in Chapter 6. Here we focus on the convergence results of numerical solutions using different grid structures and results based on grid adaptivity for deterministic channel opening. The simulation results obtained with linearly implicit methods are discussed. The next section depicts the numerical results based on stochastic channel transitions with different numbers of cluster arrangements. In the next section the three dimensional simulation results are presented. In the final section, the parallel numerical results using the domain decomposition methods are presented with the opening of channels in the clusters in a prescribed deterministic way. Furthermore, the parallel efficiency of different processors for this particular problem is also discussed.

Finally, a brief summary and future tasks are presented in Chapter 7.

Chapter 2

Mathematical Modeling of Heat and Mass Transfer in Fluidized Beds

The traditional importance of heat and mass transfer in chemistry, physics, and engineering, and the recent development of various reaction-diffusion processes in biology, ecology, and biochemistry have led to many physical interesting and mathematically challenging problems using semilinear parabolic and hyperbolic equations. From the process engineering point of view, the fabrication and subsequent treatment of disperse products are very important. This is due to the fact that 60 % of all products of the chemical industry are particles, see [96].

Fluidized bed spray granulation is a process used for the production of granular high-quality, free-flowing, low-dust and low-attrition solids originating from liquid products, like solutions, suspensions, melts and emulsions. This is achieved

- by converting the suspended solids contained in the liquids into granulates,
- by transforming the powder like accumulating solids into granulates,
- by coating or by fixing the solid granulates in a matrix,
- or by agglomerating the solid particles.

Definition: A fluidizing gas passes upward through a porous plate into the fluidized bed as shown in Figure. 2.1. At a low flow rate, fluid merely percolates through the void spaces between stationary particles. With an increase in flow rate, particles move apart and a few are seen to vibrate. At a still higher velocity, a point is reached when the particles are all just suspended in the upward flowing gas. And the liquid to be granulated is usually sprayed with a jet into the fluidized bed of solid particles, whereby some amount of liquid

forms a layer on the particles. The spraying in can occur from the top down, from the bottom up, or sideways with a jet submerged in a chosen position. The solvent evaporates in the hot, unsaturated fluidizing gas and the solid grows in layers on the particle surface. This process is called *granulation* or *layered growth process*.

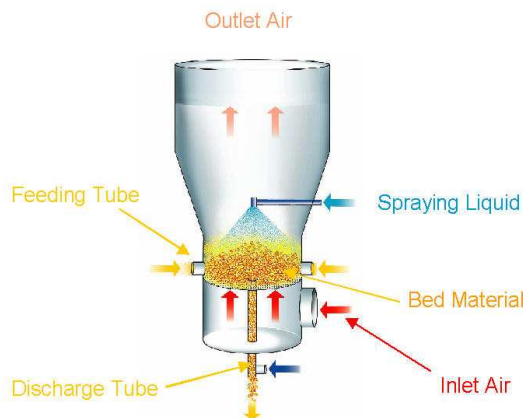


Figure 2.1: Fluidized bed apparatus

The modeling of heat and mass transfer in gas-solid-fluidized beds with spray injection which are widely used for the formation of particles from liquid solutions or suspensions as well as for the coating of particles with solid layers for the production of functional surfaces to enhance their handling properties, e.g. solvability properties, controlled release or protection for chemical reactions. Such a fluidized bed spray granulation (FBSG) system involves high heat and mass transfer and mixing properties, as well as the coupling of wetting, drying, particle enlargement, homogenization and separation processes. In FBSG, the liquid is sprayed with a nozzle as droplets on solid particles. The droplets are deposited on the particles and distributed through spreading. The solvent evaporates in the hot, unsaturated fluidization gas, thereby the solid grows in layers on the particle surface. The process conditions in the injection zone have a strong influence on the local particle volume concentrations, particle velocities, deposition of the liquid droplets and solidification of the solid content of the liquid and subsequent product quality.

So, understanding the mechanism occurring in the injection zone is essential in order to achieve and control desired product qualities. In spite of the common use of fluidized beds for agglomeration, granulation and coating tasks only a few investigations on the injection into fluidized beds are available in the literature. Contributions to this research have been reported by [13, 52, 61, 66, 67, 73, 97].

The FBSG is usually upscale by calculating the dependency of the water evaporation from the air inlet temperature. But this coarse balancing is not sufficient to estimate temperature and moisture gradients of the particles as well as necessary heat and mass transfer surfaces. More details on the experimental and theoretical work on temperature and concentration distributions in liquid sprayed fluidized beds can be found in the literature, see e.g. [48, 49].

Another important application of liquid injection into fluidized beds is the heterogeneous gas-phase catalysis, e.g. Fluid Catalytic Cracking / FCC of hydrocarbons, where reactants are injected in the liquid state and thus released vapor will perform the respective chemical reactions, see Bruhns and Werther [25]. For fluidized bed reactors, a detailed knowledge of the mechanisms of liquid injection is required, not only for engineering design but also for safe and economic operation in an industrial scale.

In comparison to Heinrich et al. [48], Heinrich and Mörl [49] and Henneberg [51] we use an improved model as well as a much more sophisticated numerical methods. The modeling was made mesh independent in order to be more flexible with meshes. The previously structured finite difference method was replaced by a finite element method allowing unstructured grids in 2D and 3D. This flexibility allows the computation of an apparatus with non rectangular geometry as well as local mesh adaption in the solution process. Further, we allow parallelization to speed up computations for complex applications.

In the remaining part of this chapter we derive our model equations in the most general form, the integral conservation law form, and then recast them into divergence form, which is natural for finite volume schemes and finite element methods. The main focus is on solving the model equations, in particular, the convection-reaction-diffusion equations with appropriate boundary conditions. The concepts of convection-reaction-diffusion equations are prevalent in our construction of numerical methods and the recurring use of these model equations allows us to develop a consistent frame work for analysis of consistency, accuracy, stability and convergence. The model equations we study have coupled and semilinear behavior. We will show the total balances, which will be used later in order to check the numerical accuracy of computed steady states. In the next subsection we will give the model equations for the sprayed liquid in two and three space dimensions. In the final subsection we will explain the invariant regions used to check the necessary conditions for the balance equations.

2.1 Introduction of basic variables and assumptions

In the following we will use some constants and notations that are introduced in Appendix A.

Wetted particles: The liquid is sprayed onto the fluidized bed particles in the form of little droplets. It is assumed that the drops spread on the particles as a film with constant thickness F . Assuming that the particles are 100% wetted, the *maximum liquid concentration* is

$$\kappa_{L,max} = A^* \epsilon^* F \rho_L.$$

The *wetting efficiency*

$$\phi = \frac{A_{wetted}}{A_{wetted} + A_{unwetted}} = \frac{\kappa_L}{\kappa_{L,max}}.$$

marks the rate of wetted particle surface to overall particle surface.

Partial pressures: The separation of the gas phase into its components, air and vapor, is described through partial pressures. The constant system pressure is given as a sum of the partial pressures according to Dalton's law. Each component of the gas phase satisfied

$$p_i = \frac{\kappa_i}{\epsilon} R_i T,$$

where $R_i = \frac{R}{M_i}$ is the specific gas constant, and for 2 components i, j

$$\frac{\kappa_i}{\kappa_j} = \frac{R_j p_j}{R_i p_i}.$$

Average gas constants: Let's assume an ideal gas mixture with a mass of m_i of partial pressure of p_i a gas constant R_i , that occupies a volume V at a temperature T . Then for each component holds

$$p_i V = m_i R_i T.$$

The overall mass m and pressure p are given as the sums of the corresponding values of each component:

$$m = \sum_i m_i, \quad p = \sum_i p_i.$$

If the equation $pV = mRT$ holds also for the gas mixture, then the average gas constant \bar{R} for the gas mixture is given by

$$\bar{R} = \frac{\sum_i m_i R_i}{\sum_i m_i} = \frac{\sum_i p_i}{\sum_i p_i / R_i}.$$

Air humidity: The humid air is a mixture of air and vapor. the air humidity Y_A is defined as

$$Y_A = \frac{\kappa_V}{\kappa_A},$$

and for ideal gas holds

$$Y_A = \frac{M_V}{M_A} \frac{p_V}{p - p_V}.$$

Air density: The density of air at a pressure p and a temperature T with the specific (average) gas constant of air at $R_A = 287.22 \text{ J}/(\text{kgK})$ is

$$\rho_A = \frac{p}{R_A T}.$$

While the concentration is given by

$$\kappa_A = \frac{\epsilon p_A}{R_A T} = \epsilon \frac{p_A}{p} \rho_A.$$

The density of air with a humidity Y_A is

$$\rho_Y = \frac{p}{R_Y T}.$$

whereas the gas constant for humid air would be

$$R_Y = \frac{R_A + Y_A R_V}{1 + Y_A}.$$

Enthalpy of humid air: The volume-based enthalpy of humid air h_A is

$$h_A = \kappa_A c_{pA} \theta_A + \kappa_V c_{pV} \theta_A + \kappa_V \Delta h_{V0}.$$

Evaporation flow: An adiabatic saturation of air with vapor in the fluidized bed may be assumed. The water mass flow \dot{m}_{ev} that evaporates at the interface A between humid air and water, or condensates for $\dot{m}_{ev} < 0$, depends on the present vapor pressure p_V and on the saturated vapor pressure $p_{V,\infty}$ in the following manner

$$\dot{m}_{ev} = \beta A \frac{p M_V}{RT} \ln \frac{p - p_V}{p - p_{V,\infty}}. \quad (2.1)$$

The *saturated vapor pressure* $p_{V,\infty}$ of the water is a function of the temperature may be approximated through an "Antoine equation", see Schlünder and Tsotsas [81],

$$p_{V,\infty}(\theta_L) = \exp\left(23.462 - \frac{3978.205}{233.349 + \theta_L}\right).$$

The *mass transfer coefficient* β describes the influence of the particle flow on the vapor transport. An approximation of the Sherwood number is used for its calculation, see Appendix A. For small partial vapor pressures $p_V < p$, in other words temperatures well under water's boiling point, Eq. (2.1) may be simplified by the approximation $\ln(1+x) \approx x$,

$$\dot{m}_{ev} = \beta A \frac{p M_V}{RT} \ln \frac{p_{V,\infty} - p_V}{p - p_{V,\infty}}. \quad (2.2)$$

The *adiabatic saturation humidity*

$$Y_{sat} = \frac{M_V}{M_A} \frac{p_{V,\infty}}{p - p_{V,\infty}},$$

where $M_V/M_A = 0.622$, with $M_V = 0.01802 \text{ kg/mol}$, $M_A = 0.02896 \text{ kg/mol}$, is the humidity for which liquid film and humid air are in equilibrium, and depends through the saturated vapor pressure $p_{V,\infty}$ on the liquid film temperature θ_L . With the approximation $\frac{p-p_V}{p-p_{V,\infty}} \approx 1$, Eq. (2.2) can be written as follows

$$\dot{m}_{ev} \approx \beta A \frac{p M_A}{RT} (Y_{sat} - Y_A) = \beta \rho_A A (Y_{sat} - Y_A).$$

The wetted surface, related to the volume element, is $A^* \epsilon^* \phi$, and so consequently

$$\dot{\kappa}_{ev} = \beta \rho_A A^* \epsilon^* \phi (Y_{sat} - Y_A),$$

which holds for the evaporation flow (evaporated mass flow) per volume unit.

Enthalpy flow of the evaporated water: It is assumed that the required heat for the evaporation is taken from the liquid film, while the required heat for the temperature rise from θ_L to θ_A comes from the air. Under this condition, the evaporated water flow, $\dot{\kappa}_{ev}$ takes with it also the volume based enthalpy flow by the evaporation of the liquid film

$$\dot{q}_{ev} = \dot{\kappa}_{ev} (\Delta h_{V0} + c_{pV} \theta_L).$$

Heat transfer at the interfaces: Figure 2.2 illustrates, alongside with the enthalpy flow of evaporated water, the enthalpy flows between unwetted particles and air, between liquid film and air, as well as particles and liquid film. Heat is transported on the interfaces of

each phase, particle-air (PA), liquid film-air (LA), particle-liquid film (PL), particle-wall (PW), air-wall (AW), and wall-environment (WE), through convection, conduction and radiation. The *surface-based heat flow* \dot{q}_{ij}^A over an interface A_{ij} (from i to j) is described by a heat transfer coefficient α_{ij}

$$\dot{q}_{ij}^A = \frac{dq_{ij}^A}{dA} = \alpha_{ij}(\theta_i - \theta_j).$$

2.2 Balance equations

Our model equations can be written in the following general form,

$$\frac{d}{dt} \int_V u dV + \int_S s(u) \cdot \underline{n} ds = \int_V v dV. \quad (2.3)$$

In this equation, $u : \Omega \times [0, T] \rightarrow \mathbb{R}^m$ is a vector valued function containing the set of variables which are conserved, i.e. concentration and temperature per unit volume. The equation is a statement of the conservation of these quantities in a finite region of space with volume V and surface area S over a finite interval of time $[0, T]$. The vector \underline{n} is a unit vector normal to the surface pointing outward, $s(u)$ is a set of vectors, or tensors, containing the flux of u per unit area per unit time, and v is the rate of production of u per unit volume per unit time. The flux vector $s(u)$ may have a variety of forms describing different mechanisms.

Advection: $s(u) = \underline{w}u$, where \underline{w} is velocity vector.

Heat convection (Fourier's law): $s(u) = -\lambda \nabla u$, where λ is heat conduction coefficient.

Dispersion: $s(u) = -D \nabla u$, where D is the dispersion matrix.

Applying the Gauss's theorem to the flux integral in Eq. (2.3), the equation can be rewritten as,

$$\int_V \frac{\partial u}{\partial t} dV = - \int_V \nabla \cdot s(u) dV + \int_V v dV. \quad (2.4)$$

General conceptions about the pneumatics of gas-solid fluidized beds must take into account a gas phase and a suspension phase with particles and gas, which affect one another. Here the gas is viewed as an ideal plug flow and the particles are viewed as an intermixed flow.

Simplifying assumptions:

- The air possesses the character of a plug flow (PFTR).
- Perfect back mixing of the particles (CSTR) for the one-dimensional model with uniform liquid distribution.
- Only the first drying period is observed.
- Consideration of axial and radial dispersion coefficients and thus of non-ideal particle mixing for the two-dimensional model with non-uniform liquid distribution.
- The entire injected liquid is deposited onto the particles as film with constant thickness F .
- Only liquid reaches the fluidized bed through the jet employed.
- Intra particular heat transfer resistance can be neglected.

The modeling of the problem is obtained by the balance of the mass and energy of the air, of the solid as well as of the liquid contained in the fluidized bed. A similar type of modeling can be found in Heinrich et al. [48], Heinrich and Mörl [49] and Henneberg [51]. Their model includes mesh size and is not flexible with different unstructured grids. The new model does not depend on the mesh size and it is flexible with unstructured grids. Therefore, we were able to add mesh adaption as a new feature of the numerical method. We have tested this extensively in 2 and 3 space dimensions. The advantage of the present source term is flexible for higher dimensions. The balance inside the fluidized bed zone delivers a partial differential equation for each balance variable. The balance variables are understood as functions of space and time. The following model equations are in Cartesian coordinates.

Mass balance of air

A differential volume element is depicted in Figure (2.2). The water loading of the entering humid air increases during the flow, because of the evaporation flow given off from the liquid film on the particles. The vapor will be transformed into the air as a function of the conditions of mass transfer. For the mass balance of air, the following factors have to be considered,

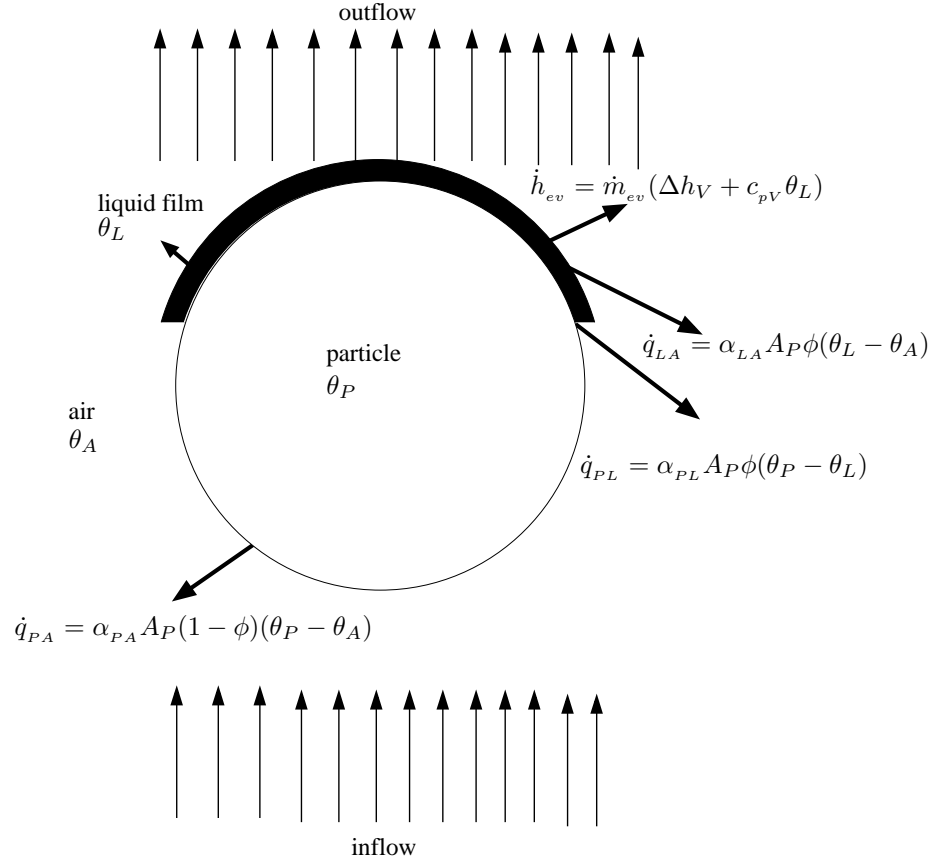


Figure 2.2: Heat and mass transfer on a wetted particle

Source term: Evaporation flow \dot{m}_{ev} .

Transport term: Convection in the plug flow $s(u) = \dot{m}_{WA}$.

Balance equation:

$$\int_V \frac{\partial m_{WA}}{\partial t} dV = - \int_V \frac{\partial \dot{m}_{WA}}{\partial z} H_{fb} dV + \int_V \dot{m}_{ev} dV.$$

$$\int_V \frac{\partial (Y_A m_A)}{\partial t} dV = - \int_V \frac{\partial (Y_A \dot{m}_A)}{\partial z} H_{fb} dV + \int_V \beta \rho_A A_P \phi (Y_{sat} - Y_A) dV.$$

The divergence of this equation is,

$$m_A \frac{\partial Y_A}{\partial t} = -\dot{m}_A H_{fb} \nabla \cdot Y_A + \beta \rho_A A_P \phi (Y_{sat} - Y_A),$$

$$\frac{\partial Y_A}{\partial t} = -W(\theta_A) \nabla \cdot Y_A + R_1(\theta_A) \phi (Y_{sat}(\theta_L) - Y_A), \quad (2.5)$$

where

$$\begin{aligned} W(\theta_A) &= -\frac{\dot{m}_A}{m_A(\theta_A)} H_{fb}(\theta_A), \\ m_A(\theta_A) &= \rho_A(\theta_A) \epsilon V_{fb}, \\ R_1(\theta_A) &= \frac{A_P \beta(\theta_A) \rho_A(\theta_A)}{m_A(\theta_A)}. \end{aligned}$$

Energy balance of air

Figure 2.2 shows the heat and mass transfers, which cause a change of the air enthalpy. In the balance area, a heat flow occurs between the unwetted portion of the solid particles and air as well as between the liquid film of the wetted part of the particles and air. In addition the enthalpy of the water evaporated from the liquid film increases the air energy. This enthalpy flow consists of the mass flow of the evaporated water, the enthalpy of the liquid, the evaporation enthalpy and the enthalpy of the vapor in the air. Heat transfer coefficients between liquid film and air α_{LA} , as well as particles and air α_{PA} are calculated according to Gnielinski [40]. Both heat transfer coefficients are assumed to be equal due to the same flow conditions, i.e. $\alpha_{PA} = \alpha_{LA} = \alpha$.

Source term: Enthalpy of the evaporation flow \dot{h}_{ev} .

Sink: Heat transfer between air and liquid film \dot{q}_{LA} as well as between air and particle \dot{q}_{PA} .

Transport term: Convection in the plug flow $s(u) = \dot{h}_{WA}$.

The balance of the enthalpy results in an equation for the temporal derivative of the air temperature.

Balance equation:

$$\int_V \frac{\partial h_{WA}}{\partial t} dV = - \int_V \frac{\partial \dot{h}_{WA}}{\partial z} H_{fb} dV - \int_V \dot{q}_{LA} dV - \int_V \dot{q}_{PA} dV + \int_V \dot{h}_{ev} dV. \quad (2.6)$$

where

$$\begin{aligned} h_{WA} &= h_A + h_V, \\ &= m_A c_{pA} \theta_A + m_V \Delta h_V + m_V c_{pV} \theta_A, \\ &= m_A c_{pA} \theta_A + m_A Y_A \Delta h_V + m_A Y_A c_{pV} \theta_A. \end{aligned}$$

$$\begin{aligned}
\frac{\partial h_{WA}}{\partial t} &= m_A c_{pA} \frac{\partial \theta_A}{\partial t} + m_A \Delta h_V \frac{\partial Y_A}{\partial t} + m_A c_{pV} \left\{ Y_A \frac{\partial \theta_A}{\partial t} + \theta_A \frac{\partial Y_A}{\partial t} \right\}, \\
&= m_A (c_{pA} + c_{pV} Y_A) \frac{\partial \theta_A}{\partial t} + m_A (\Delta h_V + c_{pV} \theta_A) \frac{\partial Y_A}{\partial t}. \\
\frac{\partial \dot{h}_{WA}}{\partial z} &= \dot{m}_A (c_A + c_{st} Y_A) \frac{\partial \theta_A}{\partial z} + \dot{m}_A (\Delta h_V + c_{st} \theta_A) \frac{\partial Y_A}{\partial z}.
\end{aligned}$$

Substitute $\frac{\partial h_{WA}}{\partial t}$, $\frac{\partial \dot{h}_{WA}}{\partial z}$ in the balance equation (2.6)

$$\begin{aligned}
&\int_V m_A (c_{pA} + c_{pV} Y_A) \frac{\partial \theta_A}{\partial t} dV + \int_V m_A (\Delta h_V + c_{pV} \theta_A) \frac{\partial Y_A}{\partial t} dV \\
&= - \int_V H_{fb} \dot{m}_A (c_{pA} + c_{pV} Y_A) \frac{\partial \theta_A}{\partial z} dV - \int_V H_{fb} \dot{m}_A (\Delta h_V + c_{pV} \theta_A) \frac{\partial Y_A}{\partial z} dV \\
&\quad - \int_V \alpha_{AL} A_P \phi (\theta_A - \theta_L) dV - \int_V \alpha_{PA} A_P (1 - \phi) (\theta_A - \theta_P) dV \\
&\quad\quad\quad + \int_V \dot{m}_{ev} (\Delta h_V + c_{pV} \theta_A) dV.
\end{aligned}$$

If we substitute the expression $\frac{\partial Y_A}{\partial t}$ from Eq. (2.5), then the evaporation term will get canceled. Finally, the divergence form of the energy balance of the air, in terms of temperature of air is

$$\frac{\partial \theta_A}{\partial t} = -W(\theta_A) \frac{\partial \theta_A}{\partial z} - Q_1(Y_A, \theta_A) \{ (1 - \phi)(\theta_A - \theta_P) + \phi(\theta_A - \theta_L) \}, \quad (2.7)$$

where $Q_1 = \frac{\alpha(\theta_A) A_P}{m_A(\theta_A)(c_{pA}(\theta_A) + c_{pV}(\theta_A) Y_A)}$.

Mass balance of water

The maximum liquid mass per volume element is defined using the effective particle surface, the thickness of the liquid film and the density of the liquid.

$$\rho_{L,max} = \phi \frac{m_{L,max}}{V_{fb}} = A^* \epsilon^* F \rho_{water}. \quad (2.8)$$

The liquid mass flow sprayed into the fluidized bed per volume element, the evaporation flow and the mixing of the bed material affect the liquid content of the volume element.

Source term: Mass flow of the drop deposition from the nozzle \dot{m}_{LV} .

Sink: Evaporation flow \dot{m}_{ev} .

Transport term: This term is determined by the particle dispersion, with liquid on the particle surface $s(u) = -D \nabla m_{WP}$.

Balance equation:

$$\int_V \frac{\partial m_{WP}}{\partial t} dV = \int_V \nabla \cdot (D\nabla m_{WP}) dV - \int_V \dot{m}_{ev} dV + \int_V \dot{m}_{LV} dV,$$

where $m_{WP} = A_P \phi F \rho_L$. Substitute this term in the above equation

$$\int_V \frac{\partial (A_P \phi F \rho_L)}{\partial t} dV = \int_V \nabla \cdot (D\nabla (A_P \phi F \rho_L)) dV - \int_V \beta \rho_A A_P \phi (Y_{sat} - Y_A) dV + \int_V \dot{m}_{LV} dV.$$

The divergence of this equation is,

$$\begin{aligned} A_P F \rho_L \frac{\partial \phi}{\partial t} &= A_P F \rho_L \nabla \cdot (D\nabla \phi) - \beta \rho_A A_P \phi (Y_{sat} - Y_A) + \dot{m}_{LV}, \\ \frac{\partial \phi}{\partial t} &= \nabla \cdot (D\nabla \phi) - \frac{\beta \rho_A}{\rho_L F} \phi (Y_{sat} - Y_A) + \frac{\dot{m}_{LV}}{A_P F \rho_L}, \\ \frac{\partial \phi}{\partial t} &= \nabla \cdot (D\nabla \phi) - R_2(\theta_A, \theta_L) \phi (Y_{sat}(\theta_L) - Y_A) + S_1(\theta_L) \dot{m}_{LV}, \end{aligned} \quad (2.9)$$

where

$$\begin{aligned} R_2(\theta_A, \theta_L) &= \frac{\beta(\theta_A) \rho_A(\theta_A)}{\rho_L(\theta_L) F}, \\ S_1(\theta_L) &= \frac{1}{A_P F \rho_L(\theta_L)}. \end{aligned}$$

Energy balance of water

A corrective factor is introduced, which specifies the ratio of the heat transport coefficient between the particle and the liquid film to the heat transfer coefficient in the fluidized bed, assuming that the heat transfer coefficient between air and particles is equal to the heat transfer coefficient between air and liquid film:

$$f = \frac{\alpha_{PL}}{\alpha}.$$

The average liquid film temperature is influenced by the axial and radial dispersion, the heat flow between the air and the liquid film as well as between the particle and liquid film, by the enthalpy loss as a result of evaporation, and the enthalpy flow brought in by the liquid sprayed into the fluidized bed.

Source term: Enthalpy of the drop deposition from the nozzle \dot{h}_s , heat transfer between liquid film and air \dot{q}_{LA} as well as between particle and liquid film \dot{q}_{PL} .

Sink: Evaporation flow \dot{h}_{ev} .

Transport term: Dispersion $s(u) = -D\nabla h_L$.

Balance equation:

$$\int_V \frac{\partial h_L}{\partial t} dV = \int_V \nabla \cdot (D\nabla h_L) dV + \int_V \dot{q}_{LA} dV + \int_V \dot{q}_{PL} dV + \int_V \dot{h}_s dV - \int_V \dot{h}_{ev} dV, \quad (2.10)$$

where

$$h_L = m_{LC_L}\theta_L = V_L\rho_{LC_L}\theta_L = A_L F\rho_{LC_L}\theta_L = A_P\phi F\rho_{LC_L}\theta_L \quad \text{and} \quad \dot{h}_s = \dot{m}_{LV}c_{L}\theta_{L,in}.$$

Substitute these terms in the balance equation (2.10), then the divergence of this equation is,

$$\begin{aligned} A_P F\rho_{LC_L} \frac{\partial(\phi\theta_L)}{\partial t} &= A_P F\rho_{LC_L} \nabla \cdot (D\nabla(\phi\theta_L)) + \alpha A_P \{\phi(\theta_A - \theta_L) + f\phi(\theta_P - \theta_L)\} \\ &\quad + \dot{m}_{LV}c_L\theta_{L,in} - \beta\rho_A A_P\phi(Y_{sat} - Y_A)(\Delta h_V + c_{pV}\theta_A). \end{aligned} \quad (2.11)$$

Divide the term $A_P F\rho_{LC_L}$ both sides, finally we obtain the following differential equation for liquid film temperature

$$\begin{aligned} \frac{\partial(\phi\theta_L)}{\partial t} &= \nabla \cdot (D\nabla(\phi\theta_L)) + Q_2(\theta_A, \theta_L) \{\phi(\theta_A - \theta_L) + f(\theta_A)\phi(\theta_P - \theta_L)\} \\ &\quad - R_3(\theta_A, \theta_L)\phi(Y_{sat}(\theta_L) - Y_A)(\Delta h_V + c_{pV}(\theta_A)\theta_A) + S_2(\theta_L)\dot{m}_{LV}, \end{aligned} \quad (2.12)$$

where $Q_2(\theta_A, \theta_L) = \frac{\alpha(\theta_A)}{F\rho_{LC_L}}$, $R_3(\theta_A, \theta_L) = \frac{\beta(\theta_A)\rho_A(\theta_A)}{F\rho_{LC_L}}$, and $S_2(\theta_L) = \frac{\theta_{L,in}}{A_P F\rho_L}$.

Energy balance of the particles

The temporal change of the enthalpy of the solid is determined by the heat exchange between particles and air, between particles and liquid as well as by the intensity of the dispersion.

Source term: Heat transfer between particle and air \dot{q}_{PA} .

Sink: Heat transfer between particle and liquid film \dot{q}_{PL} .

Transport term: Particle dispersion $s(u) = -D\nabla h_P$.

Balance equation:

$$\int_V \frac{\partial h_P}{\partial t} dV = \int_V \nabla \cdot (D\nabla h_P) dV + \int_V \dot{q}_{PA} dV - \int_V \dot{q}_{PL} dV,$$

where $h_P = m_P c_{pP} \theta_P$. The divergence of this equation is,

$$m_P c_{pP} \frac{\partial \theta_P}{\partial t} = m_P c_{pP} \nabla \cdot (D \nabla \theta_P) + \alpha_{PA} A_P (1 - \phi) (\theta_A - \theta_P) - \alpha_{PL} A_P \phi (\theta_P - \theta_L). \quad (2.13)$$

Divide the term $m_P c_{pP}$ both sides, we obtain a differential equation for particle temperature

$$\frac{\partial \theta_P}{\partial t} = \nabla \cdot (D \nabla \theta_P) + Q_3(\theta_A) \{ (1 - \phi) (\theta_A - \theta_P) - f(\theta_A) \phi (\theta_P - \theta_L) \}. \quad (2.14)$$

where $Q_3(\theta_A) = \frac{\alpha(\theta_A) A_P}{m_P c_{pP}}$.

Total balances

The following balances will be used later in order to check the numerical accuracy of computed steady states.

Air humidity: The outlet air humidity can be calculated using the total mass balance around the fluidized bed, see Heinrich [47]. The outlet air humidity depends on inlet air humidity, mass flow rate of the inlet air and mass flow rate of the liquid

$$Y_{A,out} = Y_{A,in} + \frac{\dot{m}_L}{\dot{m}_A}. \quad (2.15)$$

Air temperature: The air temperature at the outlet is determined using an enthalpy balance in the fluidized bed. The balance is

$$\theta_{A,out} = \frac{\dot{m}_A (c_{pA} \theta_{A,in} + Y_{A,in} (c_{pV} \theta_{A,in} + \Delta h_{V,0})) + \dot{m}_L c_{pL} \theta_{L,in} - \dot{m}_A Y_{A,out} \Delta h_{V,0}}{\dot{m}_A (c_{pA} + Y_{A,out} c_{pV})}. \quad (2.16)$$

2.2.1 Boundary conditions

The boundary of the fluidized bed zone is divided into 3 partial surfaces. The balance at each boundary surface delivers the corresponding boundary conditions of the partial differential equation system. The flow defined inside a fluidized bed region Ω , should be a continuous vector field in $\bar{\Omega}$. The set $\bar{\Omega}$ denotes the closure of Ω , that is the union of the domain Ω with its boundary Γ : $\bar{\Omega} = \Omega \cup \Gamma$ and $\Gamma = \Gamma_D \cup \Gamma_N$. Here $\Gamma_i \times [0, \tau]$ indicates the boundary surface from initial time 0 seconds to final time τ seconds where $i = D, N$; D indicates Dirichlet and N indicates Neumann boundary conditions. The vector \underline{n} denotes the outward normal unit vector at the boundary Γ , λ characterizes the heat conduction coefficient on the apparatus surface and D is the dispersion matrix.

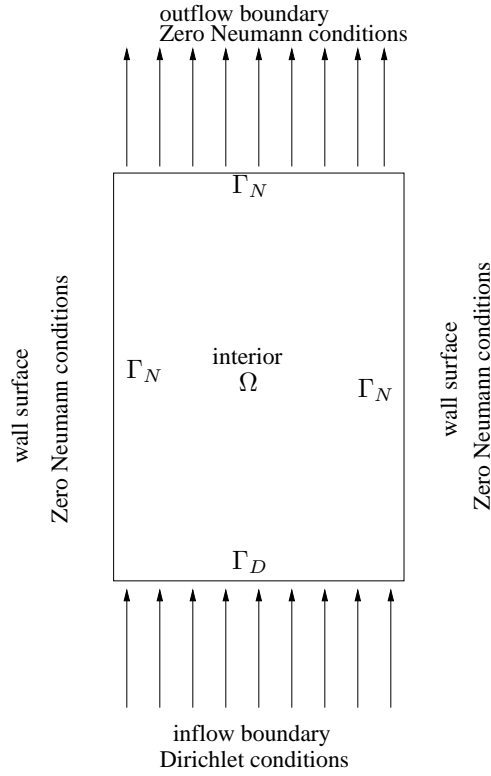


Figure 2.3: Coordinate system of the nozzle

The bottom surface:

Vapor in the gas flow is transported from the air distributor to the top boundary of the fluidized bed zone. The humidity of air Y_A and the air temperature θ_A are determined at the boundary by the inflow of air humidity $Y_{A,in}$ and inflow of air temperature $\theta_{A,in}$

$$\begin{aligned} Y_{A|bot} &= Y_{A,in} \text{ on } \Gamma_D \times [0, \tau], \\ \theta_{A|bot} &= \theta_{A,in} \text{ on } \Gamma_D \times [0, \tau]. \end{aligned}$$

The boundary of the bed zone is not influenced by the particles and liquid flow. So that the flow over the bottom boundary for degree of wetting ϕ , liquid film temperature θ_L and particle temperature θ_P is zero

$$\begin{aligned} \underline{n} \cdot D\nabla\phi &= 0 \text{ on } \Gamma_N \times [0, \tau], \\ \underline{n} \cdot D\nabla\theta_L &= 0 \text{ on } \Gamma_N \times [0, \tau], \\ \underline{n} \cdot D\nabla\theta_P &= 0 \text{ on } \Gamma_N \times [0, \tau], \end{aligned}$$

where $Y_{A,in}, \theta_{A,in} \in \mathbb{R}$ are given constant data.

The apparatus wall surface:

The apparatus wall has an effect on the temperature and concentration progressions in the fluidized bed. In direct contact, it can absorb or radiate heat. The heat flow is determined by the heat transfer coefficient α . The temperature gradient at the wall is approximated by the temperature from the interior to temperature at the wall. For the numerical solution we make the simplifying assumption that the wall temperature is constant, so that the temperature gradient at the wall is zero

$$\begin{aligned}\underline{n} \cdot D\nabla\theta_L &= 0 \text{ on } \Gamma_N \times [0, \tau], \\ \underline{n} \cdot D\nabla\theta_P &= 0 \text{ on } \Gamma_N \times [0, \tau].\end{aligned}$$

Further there is no flux of degree of wetting,

$$\underline{n} \cdot D\nabla\phi = 0 \text{ on } \Gamma_N \times [0, \tau].$$

The top surface:

This boundary is an outflow boundary, so Eq. (2.5) and Eq. (2.7) do not have a boundary condition. The diffusive equations need a boundary condition. Here we also assume that no particles and liquid leave the apparatus. Therefore, it makes sense to assume that there is no flux of degree of wetting ϕ , liquid film temperature θ_L and particle temperature θ_P .

$$\begin{aligned}\underline{n} \cdot D\nabla\phi &= 0 \text{ on } \Gamma_N \times [0, \tau], \\ \underline{n} \cdot D\nabla\theta_L &= 0 \text{ on } \Gamma_N \times [0, \tau], \\ \underline{n} \cdot D\nabla\theta_P &= 0 \text{ on } \Gamma_N \times [0, \tau].\end{aligned}$$

2.3 Two dimensional model for the liquid injection into the fluidized bed

The temperature and concentration distributions inside the fluidized bed are determined by the spatial distribution of the sprayed liquid and by liquid evaporation. The spraying area depends on the spraying angle and the penetration depth of the liquid droplets atomized by the nozzle, whereby the spraying angle is influenced by the nozzle characteristics. The penetration depth is determined by the intensity of the liquid drop deposition on the particles. The intensity of the drop deposition can be described by the *deposition efficiency* ϕ_{dep} . As an initial step, we assume that the drop deposition is constant for the calculation. The spray jet is considered to be a homogeneous conical drop flow. In the following, the

polar coordinate system with coordinates (r, θ) based on the nozzle axis was used, see Figure 2.4.

Drop penetration depth: The *drop concentration* $\kappa_{LV,dr}$ in the spray jet is related to the initial concentration $\kappa_{LV,dr,0}$ and depends on the distance to the nozzle exit s_{dr} . For a one-dimensional consideration, under the assumption of a parallel velocity field according to Löffler [62] one obtains

$$-\frac{\kappa_{LV,dr}}{\kappa_{LV,dr,0}} = \frac{3}{2} \frac{1 - \epsilon}{\epsilon} \frac{\varphi_{dep}}{d_p} s_{dr}. \quad (2.17)$$

The penetration depth rises with the increase of the fluidized bed porosity as well as with the increase of the particle diameter. The deposition efficiency can be calculated as $\varphi_{dep} = \eta h$, where η is the *impingement efficiency* and h is the *adhesion efficiency*. The interested reader is referred to Heinrich and Mörl [49] for more details about impingement efficiency, adhesion efficiency and deposition efficiency.

Linear drop path: It is assumed that the liquid drops entering the fluidized bed move with a constant velocity, until they adhere onto a particle. In this way, the drop velocity field \underline{w}_{dr} is radially symmetric to the nozzle center: $\underline{w}_{dr} = w_{dr} \underline{e}_r$ where \underline{e}_r is the unit vector in radial direction.

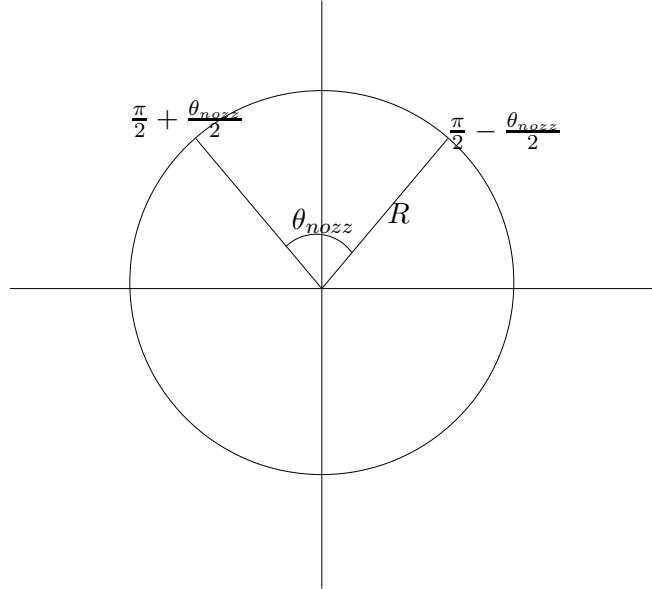


Figure 2.4: Coordinate system of the nozzle

Radially symmetric drop distribution: The following is assumed to calculate the spatial distribution of the drop concentration $\kappa_{LV,dr}(r, \vartheta)$:

- The entering liquid within a cone with an angle θ_{nozz} around the nozzle axis is evenly distributed. The drop concentration outside the cone is considered to be zero.
- The variables influencing the drop deposition are locally independent.

The drop concentration within the jet region only depends on the distance from the nozzle, see Figure 2.4. Thus it follows that

$$\dot{\kappa}_{LV,dr}(\mathbf{x}) = \begin{cases} \tilde{\kappa}(r, \vartheta) & \text{if } \frac{\pi}{2} - \frac{\theta_{nozz}}{2} \leq \vartheta \leq \frac{\pi}{2} + \frac{\theta_{nozz}}{2}, \\ 0 & \text{otherwise.} \end{cases} \quad (2.18)$$

Drop balance: Here we use the following notations:

$\underline{w}_{dr} = (w_r, 0)$, the average drop path length $s_{dr} \in \mathbb{R}^+$ is constant,

$$\tilde{\kappa}(r, \vartheta) = \left\{ \tilde{\kappa}(r, \vartheta) \mid 0 < r \leq R, \frac{\pi}{2} - \frac{\theta_{nozz}}{2} \leq \vartheta \leq \frac{\pi}{2} + \frac{\theta_{nozz}}{2} \right\} \in \mathbb{R}^+.$$

The liquid flow transported through the spray jet is equal to the product of the drop concentration $\kappa_{LV,dr}$ and the drop path velocity field \underline{w}_{dr} . Here we assume that the drop path velocity field depends only on the height of the fluidized bed. The balance equation of the drop concentration is in polar coordinates

$$\frac{\partial \kappa_{LV,dr}}{\partial t} = -\nabla \cdot (\kappa_{LV,dr} \underline{w}_{dr}) - \tilde{\kappa},$$

where $\tilde{\kappa}$ is the flux of drop concentration lost due to deposition on particles. Here we assumed that the drop concentration is independent of time. So the balance equation is

$$\nabla \cdot (\kappa_{LV,dr} \underline{w}_{dr}) + \tilde{\kappa} = 0. \quad (2.19)$$

For $\kappa_{LV,dr} \underline{w}_{dr} = \kappa_{LV,dr} w_{dr} \underline{e}_r$ one obtains

$$\begin{aligned} \nabla \cdot (\kappa_{LV,dr} \underline{w}_{dr}) &= \nabla \cdot (\kappa_{LV,dr} w_{dr} \underline{e}_r), \\ &= w_{dr} (\underline{e}_r \cdot \nabla \kappa_{LV,dr} + \kappa_{LV,dr} \cdot \nabla \underline{e}_r), \\ &= w_{dr} \left(\frac{\partial \kappa_{LV,dr}}{\partial r} + \frac{\kappa_{LV,dr}}{r} \right). \end{aligned}$$

Drop path length: Assume that

- the liquid drops and the bed particles are evenly formed spheres, and
- the deposition efficiency φ_{dep} is constant and locus independent for all liquid drops.

Then an average drop path length is obtained approaching the problem geometrically as in Eq. (2.17)

$$s_{dr} = \frac{2}{3} \frac{\epsilon}{1 - \epsilon} \frac{d_p}{\varphi_{dep}}. \quad (2.20)$$

Using this condition, the following holds

$$\tilde{\kappa} = \frac{w_{dr}}{s_{dr}} \kappa_{LV,dr}. \quad (2.21)$$

Using the above equations, the following differential equation is obtained

$$\frac{d\kappa_{LV,dr}}{dr} + \left(\frac{1}{r} + \frac{1}{s_{dr}} \right) \kappa_{LV,dr} = 0.$$

Solving the above differential equation, we obtain the drop concentration with parameterized group of solutions via $C(\vartheta) \in \mathbb{R}$,

$$\kappa_{LV,dr}(r, \vartheta) = C(\vartheta) \frac{\exp\left(\frac{-r}{s_{dr}}\right)}{r}.$$

2.3.1 Spraying mass balance equation

The function $C(\vartheta)$ is determined by the sprayed liquid mass per unit time \dot{m}_L . The liquid flow enters the region

$$\Omega_n = \left\{ (r, \vartheta) \mid 0 < r \leq R, \frac{\pi}{2} - \frac{\theta_{nozz}}{2} \leq \vartheta \leq \frac{\pi}{2} + \frac{\theta_{nozz}}{2} \right\}$$

with a radius R around the nozzle center

$$\int_{\Omega_n} \tilde{\kappa} dA = - \int_{\Omega_n} \nabla \cdot (\kappa_{LV,dr} \underline{w}_{dr}) dA.$$

Integrating both sides

$$\begin{aligned} \dot{m}_{dr}(R) &= - \int_{\pi/2 - \theta_{nozz}/2}^{\pi/2 + \theta_{nozz}/2} \int_0^R w_{dr} \left(\frac{\partial \kappa_{LV,dr}}{\partial r} + \frac{\kappa_{LV,dr}}{r} \right) r dr d\vartheta, \\ &= - \int_{\pi/2 - \theta_{nozz}/2}^{\pi/2 + \theta_{nozz}/2} w_{dr} C \left(\exp\left(\frac{-R}{s_{dr}}\right) - 1 \right) d\vartheta, \\ &= C w_{dr} \theta_{nozz} \left(1 - \exp\left(\frac{-R}{s_{dr}}\right) \right). \end{aligned}$$

For the boundary value we get $R \rightarrow \infty$, the liquid flow must be equal to the sprayed liquid mass \dot{m}_L

$$\begin{aligned} \dot{m}_L &= \lim_{R \rightarrow \infty} \dot{m}_{dr}(R) = C w_{dr} \theta_{nozz} \lim_{R \rightarrow \infty} \left(1 - \exp\left(\frac{-R}{s_{dr}}\right) \right), \\ &= C w_{dr} \theta_{nozz}. \end{aligned} \quad (2.22)$$

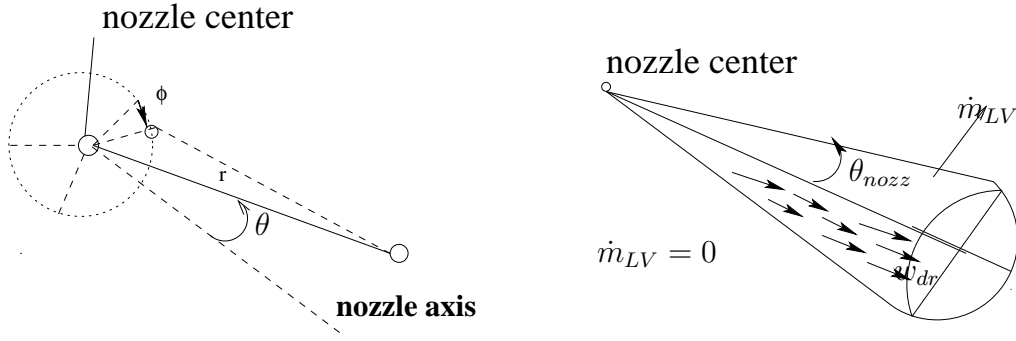


Figure 2.5: Spherical coordinate system of the nozzle and nozzle jet

As a result, we get

$$C = \frac{\dot{m}_L}{\theta_{nozz} w_{dr}}.$$

With the help of Eq. (2.21), the spatial distribution of the drop deposition is obtained

$$\begin{aligned} \tilde{\kappa}(r, \vartheta) &= \frac{w_{dr}}{s_{dr}} C \frac{\exp\left(\frac{-r}{s_{dr}}\right)}{r}, \\ &= \frac{\dot{m}_L}{\theta_{nozz}} \frac{\exp\left(\frac{-r}{s_{dr}}\right)}{r s_{dr}}. \end{aligned} \quad (2.23)$$

2.4 Three dimensional model for the liquid injection into the fluidized bed

In the following, the spherical coordinate system with coordinates (r, θ, φ) based on the nozzle axis was used, see Figure 2.5.

Radially symmetric drop distribution: The following is assumed to calculate the spatial distribution of the drop concentration $\kappa_{LV,r}(r, \theta, \varphi)$:

- The entering liquid within a cone with an angle θ_{nozz} around the nozzle axis is evenly distributed. The drop concentration outside the cone is considered to be zero.
- The variables influencing the drop deposition are locally independent.

The drop concentration within the jet sphere is only dependent on the distance from the nozzle, see Figure 2.5. Thus it follows that

$$\dot{\kappa}_{LV,dr}(\mathbf{x}) = \begin{cases} \tilde{\kappa}(r, \vartheta, \varphi) & \text{if } \frac{\pi}{2} - \frac{\theta_{nozz}}{2} \leq \vartheta \leq \frac{\pi}{2} + \frac{\theta_{nozz}}{2}, \\ 0 & \text{otherwise.} \end{cases} \quad (2.24)$$

Drop balance: Here we use the following notations:

$\underline{w}_{dr} = (w_r, 0, 0)$, average drop path length $s_{dr} \in \mathbb{R}^+$ is constant,

$$\tilde{\kappa}(r, \vartheta, \varphi) = \{\tilde{\kappa}(r, \vartheta, \varphi) \mid 0 < r \leq R, \frac{\pi}{2} - \frac{\theta_{nozz}}{2} \leq \vartheta \leq \frac{\pi}{2} + \frac{\theta_{nozz}}{2}, 0 \leq \varphi \leq 2\pi\} \in \mathbb{R}^+.$$

The liquid flow transported through the spray jet is equal to the product of the drop concentration $\kappa_{LV,dr}$ and the drop path velocity field \underline{w}_{dr} . Here we assumed that drop path velocity field depends only on the height of the fluidized bed. The balance equation of the drop concentration is in sphere coordinates

$$\frac{\partial \kappa_{LV,dr}}{\partial t} = -\nabla \cdot (\kappa_{LV,dr} \underline{w}_{dr}) - \tilde{\kappa}.$$

Where $\tilde{\kappa}$ is the flux of drop concentration lost due to deposition on particles. Here we assumed that the drop concentration is independent of time. So the balance equation is

$$\nabla \cdot (\kappa_{LV,dr} \underline{w}_{dr}) + \tilde{\kappa} = 0. \quad (2.25)$$

For $\kappa_{LV,dr} \underline{w}_{dr} = \kappa_{LV,dr} w_{dr} \underline{e}_r$ holds.

$$\begin{aligned} \nabla \cdot (\kappa_{LV,dr} \underline{w}_{dr}) &= \nabla \cdot (\kappa_{LV,dr} w_{dr} \underline{e}_r), \\ &= w_{dr} (\underline{e}_r \cdot \nabla \kappa_{LV,dr} + \kappa_{LV,dr} \cdot \nabla \underline{e}_r), \\ &= w_{dr} \left(\frac{\partial \kappa_{LV,dr}}{\partial r} + \frac{2\kappa_{LV,dr}}{r} \right). \end{aligned}$$

Drop path length: Assuming that,

- the liquid drops and the bed particles are evenly formed spheres, and
- the deposition efficiency φ_{dep} is constant and locus independent for all liquid drops.

An average drop path length is obtained approaching the problem geometrically as in Eq. (2.17),

$$s_{dr} = \frac{2}{3} \frac{\epsilon}{1 - \epsilon \varphi_{dep}} \frac{d_p}{s}. \quad (2.26)$$

Using this condition, the following holds

$$\tilde{\kappa} = \frac{w_{dr}}{s_{dr}} \kappa_{LV,dr}. \quad (2.27)$$

Using the above equations, the following differential equation is obtained

$$\frac{d\kappa_{LV,dr}}{dr} + \left(\frac{2}{r} + \frac{1}{s} \right) \kappa_{LV,dr} = 0,$$

Solving the above differential equation we obtain the drop concentration with parameterized group of solutions via $C(\vartheta) \in \mathbb{R}$

$$\kappa_{LV,dr}(r, \vartheta) = C(\vartheta) \frac{\exp\left(\frac{-r}{s_{dr}}\right)}{r^2}.$$

2.4.1 Spraying mass balance equation

The function $C(\vartheta, \varphi)$ is determined by the sprayed liquid mass per unit time \dot{m}_L . The liquid flow enters in the conical region

$$\Omega_n = \{(r, \vartheta, \varphi) \mid 0 < r \leq R, \frac{\pi}{2} - \frac{\theta_{nozz}}{2} \leq \vartheta \leq \frac{\pi}{2} + \frac{\theta_{nozz}}{2}, 0 \leq \varphi \leq 2\pi\}$$

with a radius R around the nozzle center:

$$\int_{\Omega_n} \tilde{\kappa} dA = - \int_{\Omega_n} \nabla \cdot (\kappa_{LV,dr} \underline{w}_{dr}) dA.$$

Integrating both sides

$$\begin{aligned} \dot{m}_{dr}(R) &= - \int_0^{2\pi} \int_{\pi/2 - \theta_{nozz}/2}^{\pi/2 + \theta_{nozz}/2} \int_0^R w_{dr} \left(\frac{\partial \kappa_{LV,dr}}{\partial r} + \frac{\kappa_{LV,dr}}{r} \right) r^2 \sin \vartheta dr d\vartheta d\varphi, \\ &= - \int_0^{2\pi} \int_{\pi/2 - \theta_{nozz}/2}^{\pi/2 + \theta_{nozz}/2} w_{dr} C \left(\exp\left(\frac{-R}{s_{dr}}\right) - 1 \right) \sin \vartheta d\vartheta d\varphi, \\ &= \int_0^{2\pi} C w_{dr} \left(1 - \exp\left(\frac{-R}{s_{dr}}\right) \right) 2 \sin\left(\frac{\theta_{nozz}}{2}\right) d\varphi, \\ &= 4C\pi w_{dr} \left(1 - \exp\left(\frac{-R}{s_{dr}}\right) \right) \sin\left(\frac{\theta_{nozz}}{2}\right). \end{aligned}$$

For the boundary value we take $R \rightarrow \infty$, the liquid flow must be equal to the sprayed liquid mass \dot{m}_L

$$\begin{aligned} \dot{m}_L &= \lim_{R \rightarrow \infty} \dot{m}_{dr}(R) = 4C\pi w_{dr} \sin\left(\frac{\theta_{nozz}}{2}\right) \lim_{R \rightarrow \infty} \left(1 - \exp\left(\frac{-R}{s_{dr}}\right) \right), \\ &= 4C\pi w_{dr} \sin\left(\frac{\theta_{nozz}}{2}\right). \end{aligned} \quad (2.28)$$

As a result, we get

$$C = \frac{\dot{m}_L}{4\pi w_{dr} \sin\left(\frac{\theta_{nozz}}{2}\right)}.$$

With the help of Eq. (2.27), the spatial distribution of the drop deposition is obtained as

$$\begin{aligned} \tilde{\kappa}(r, \vartheta, \varphi) &= \frac{w_{dr}}{s_{dr}} C \frac{\exp\left(\frac{-r}{s_{dr}}\right)}{r^2}, \\ &= \frac{\dot{m}_L}{4\pi \sin\left(\frac{\theta_{nozz}}{2}\right)} \frac{\exp\left(\frac{-r}{s_{dr}}\right)}{r^2 s_{dr}}. \end{aligned} \quad (2.29)$$

2.5 Invariant regions

In this section we give condition for an invariant region for the balance equations (2.5), (2.7), (2.9), (2.12) and (2.14). It provides a suitable theoretical foundation and frame work for studying the large time behavior of solutions. First we introduce a short hand notation for the system.

If we set

$$\begin{aligned}
 u &= (Y_A, \theta_A, \phi, \theta_L, \theta_P), \\
 \mathbf{M} &= \text{diag}(-W, -W, 0, 0, 0), \\
 \mathbf{D} &= \text{diag}(0, 0, D, D, D), \\
 \mathbf{F}(u) &= \left(R_1\phi(Y_{sat} - Y_A), -Q_1\{(1 - \phi)(\theta_A - \theta_P) + \phi(\theta_A - \theta_L)\}, \right. \\
 &\quad \left. - R_2\phi(Y_{sat} - Y_A) + S_1\dot{m}_{LV}, \right. \\
 &\quad \left. Q_2\{\phi(\theta_A - \theta_L) + f_\alpha\phi(\theta_P - \theta_L)\} - R_3\phi(Y_{sat} - Y_A)(\Delta h_V + c_{pV}\theta_A) + S_2\dot{m}_{LV}, \right. \\
 &\quad \left. Q_3\{(1 - \phi)(\theta_A - \theta_P) - f_\alpha\phi(\theta_P - \theta_L)\} \right).
 \end{aligned}$$

Then we can write the set of Eqs. (2.5), (2.7), (2.9), (2.12) and (2.14), as

$$u_t = \mathbf{D}u_{xx} + \mathbf{M}u_x + \mathbf{F}(u) \quad (2.30)$$

with the initial data

$$u(x, 0) = u_0(x), \quad x \in \Omega. \quad (2.31)$$

$$\left. \begin{aligned}
 &\text{Here } R_1, R_2, R_3, Q_1, Q_2, Q_3, S_1, S_2, \dot{m}_{LV}, f_\alpha, \Delta h_v, c_{st} > 0, \text{ and } 0 \leq Y_A \leq Y_{sat}, \\
 &0 \leq \theta_A \leq \chi, 0 \leq \phi \leq 1, 0 \leq \theta_L \leq \chi, \text{ and } 0 \leq \theta_P \leq \chi, \text{ where } \chi > 0 \text{ is inlet} \\
 &\text{temperature of air.}
 \end{aligned} \right\} \quad (2.32)$$

Definition 2.1. A closed subset $I \subset \mathbb{R}$ is called a (*positively*) *invariant region* for the local solution defined by system (2.30), (2.31), if any solution $u(x, t)$ having all of its boundary and initial values in I , satisfies $u(x, t) \in I$ for all $x \in \Omega$ and for all $t \in [0, T)$, see Smoller [85], page 199.

The invariant regions I we consider will be made up of the intersection of "half spaces", i.e. we consider regions I of the form

$$I = \cap_{i=1}^m \{u \in S : G_i(u) \leq 0\}, \quad (2.33)$$

where G_i are smooth real-valued functions defined on open subsets of S , and for each i , the gradient ∇G_i never vanishes.

Definition 2.2. The smooth function $G : \mathbb{R}^n \rightarrow \mathbb{R}$ is called *quasi-convex* at u if whenever the gradient vanishes, i.e. $\nabla G_u(\eta) = 0$, then the Hessian is non-negligible, i.e. $\mathcal{H}G_u(\eta, \eta) \geq 0$.

Theorem 2.5.1. Let I be defined by (2.33), and suppose that for all $t \in \mathbb{R}^+$ and for every $u_0 \in \partial I$, so $G_i(u) = 0$ for some i , the following conditions hold:

1. ∇G_i at u_0 is a left eigenvector of $\mathbf{D}(u_0, x)$, and $\mathbf{M}(u_0, x)$, for all $x \in \omega$.
2. if $\nabla G_i \mathbf{D}(u_0, x) = \mu \nabla G_i$, with $\mu \neq 0$, then G_i is quasi-convex at u_0 .
3. $\nabla G_i \cdot \mathbf{F} \leq 0$ at u_0 , for all $t \in \mathbb{R}^+$.

Then I is invariant for (2.30).

Proof. See Smoller [85], page 200. □

Theorem 2.5.2. Let I be defined by (2.33), and suppose that I is an invariant region for (2.30), where $\mathbf{F} = \mathbf{F}(u, t)$ and \mathbf{D} is a positive definite matrix. Then the following conditions hold at each point u_0 on ∂I , say, $G_i(u_0) = 0$:

1. ∇G_i at u_0 is a left eigenvector of $\mathbf{D}(u_0, x)$, for all $x \in \omega$.
2. G_i is quasi-convex at u_0 .
3. $\nabla G_i \cdot \mathbf{F} \leq 0$ for all $t \geq 0$.

Proof. See Smoller [85], page 204. □

Our observation is that under some additional restrictions to be determined below there exists a *bounded* invariant region of type (2.33) for this initial value problem (2.30), (2.31). It consists of points between the curves $G_1(u) = -Y_A$, $G_2(u) = -\theta_A$, $G_3(u) = -\phi$, $G_4(u) = -\theta_L$, $G_5(u) = -\theta_P$, $G_6(u) = Y_A - Y_{sat}$, $G_7(u) = \theta_A - \chi$, $G_8(u) = \phi - 1$, $G_9(u) = \theta_L - \chi$ and $G_{10}(u) = \theta_P - \chi$. In particular

$$I = \{Y_A, \theta_A, \phi, \theta_L, \theta_P : 0 \leq Y_A \leq Y_{sat}, 0 \leq \theta_A, \theta_L, \theta_P \leq \chi, 0 \leq \phi \leq 1\}.$$

Here we want to show that I is an invariant region for the full system. The gradients are

$$\nabla G_1 = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \nabla G_2 = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \nabla G_3 = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \end{pmatrix}, \nabla G_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{pmatrix}, \nabla G_5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{pmatrix},$$

$$\nabla G_6 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \nabla G_7 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \nabla G_8 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \nabla G_9 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \nabla G_{10} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

To see that I is an invariant region, we first note that all ∇G_i are left eigenvectors of $\mathbf{D}(u_0, x)$. Thus if we set $G_1(u) = -Y_A$, then this implies that

$$\nabla G_1 \cdot F_1 |_{Y_A=0} = -R_1 \phi(Y_{sat} - Y_A) |_{Y_A=0} = -R_1 \phi Y_{sat} \leq 0.$$

Similarly we can check for other curves $G_i(u)$. If we set $G_2(u) = -\theta_A$, then this implies that

$$\begin{aligned} \nabla G_2 \cdot F_2 |_{\theta_A=0} &= (-1) \{-Q_1 \{(1 - \phi)(\theta_A - \theta_P) + \phi(\theta_A - \theta_L)\}\} |_{\theta_A=0}, \\ &= Q_1 \{(1 - \phi)(-\theta_P) + \phi(-\theta_L)\} \leq 0, \quad \text{in } I. \end{aligned}$$

If we set $G_3(u) = -\phi$, then this implies that

$$\begin{aligned} \nabla G_3 \cdot F_3 |_{\phi=0} &= (-1) \{-R_2 \phi(Y_{sat} - Y_A) + S_1 \dot{m}_{LV}\} |_{\phi=0}, \\ &= -S_1 \dot{m}_{LV} \leq 0 \quad \text{in } I. \end{aligned}$$

If we set $G_4(u) = -\theta_L$, then this implies that

$$\begin{aligned} \nabla G_4 \cdot F_4 |_{\theta_L=0} &= (-1) \left\{ Q_2 \{ \phi(\theta_A - \theta_L) + f_\alpha \phi(\theta_P - \theta_L) \} - R_3 \phi(Y_{sat} - Y_A) (\Delta h_V + c_{pV} \theta_A) \right. \\ &\quad \left. + S_2 \dot{m}_{LV} \right\} |_{\theta_L=0}, \\ &= (-1) \left\{ Q_2 \{ \phi(\theta_A) + f_\alpha \phi(\theta_P) \} - R_3 \phi(Y_{sat} - Y_A) (\Delta h_V + c_{pV} \theta_A) + S_2 \dot{m}_{LV} \right\}. \end{aligned}$$

a necessary condition that I is an invariant region is

$$R_3 \phi(Y_{sat} - Y_A) (\Delta h_V + c_{pV} \theta_A) \leq Q_2 \{ \phi \theta_A + f_\alpha \phi \theta_P \} + S_2 \dot{m}_{LV},$$

which is equivalent to

$$Y_{sat} - Y_A \leq \frac{1}{R_3 \phi (\Delta h_V + c_{pV} \theta_A)} \left\{ Q_2 \{ \phi \theta_A + f_\alpha \phi \theta_P \} + S_2 \dot{m}_{LV} \right\},$$

or

$$Y_A \geq Y_{sat} - \frac{1}{R_3\phi(\Delta h_V + c_{pV}\theta_A)} \left\{ Q_2\{\phi\theta_A + f_\alpha\phi\theta_P\} + S_2\dot{m}_{LV} \right\} \quad \text{at } \theta_L = 0. \quad (2.34)$$

This is an extra condition on the solution. If we set $G_5(u) = -\theta_P$, then this implies that

$$\begin{aligned} \nabla G_5 \cdot F_5 |_{\theta_P=0} &= (-1) \left\{ Q_3\{(1-\phi)(\theta_A - \theta_P) - f_\alpha\phi(\theta_P - \theta_L)\} \right\} |_{\theta_P=0}, \\ &= (-1) Q_3\{(1-\phi)\theta_A - f_\alpha\phi(-\theta_L)\} \leq 0 \quad \text{in } I. \end{aligned}$$

Next, we set $G_6(u) = Y_A - Y_{sat}$, and calculate $\nabla G_6 \cdot F_1 |_{Y_A=Y_{sat}}$. Then this implies that

$$\nabla G_6 \cdot F_1 |_{Y_A=Y_{sat}} = R_1\phi(Y_{sat} - Y_A) |_{Y_A=Y_{sat}} = 0 \quad \text{in } I.$$

If we set $G_7(u) = \theta_A - \chi$, then this implies that

$$\begin{aligned} \nabla G_7 \cdot F_2 |_{\theta_A=\chi} &= -Q_2\{(1-\phi)(\theta_A - \theta_P) + \phi(\theta_A - \theta_L)\} |_{\theta_A=\chi}, \\ &= -Q_2\{(1-\phi)(\chi - \theta_P) + \phi(\chi - \theta_L)\} \leq 0 \quad \text{in } I. \end{aligned}$$

If we set $G_8(u) = \phi - 1$, then this implies that

$$\begin{aligned} \nabla G_8 \cdot F_3 |_{\phi=1} &= \{-R_2\phi(Y_{sat} - Y_A) + S_1\dot{m}_{LV}\} |_{\phi=1}, \\ &= -R_2(Y_{sat} - Y_A) + S_1\dot{m}_{LV}. \end{aligned}$$

A necessary condition that I is an invariant region is

$$\begin{aligned} -R_2(Y_{sat} - Y_A) + S_1\dot{m}_{LV} &\leq 0, \\ Y_A &\leq Y_{sat} - \frac{S_1}{R_2}\dot{m}_{LV} \quad \text{at } \phi = 1. \end{aligned} \quad (2.35)$$

This is an extra condition on the solution. If we set $G_9 = \theta_L - \chi$, then this implies

$$\begin{aligned} \nabla G_9 \cdot F_4 |_{\theta_L=\chi} &= \left\{ Q_2\{\phi(\theta_A - \theta_L) + f_\alpha\phi(\theta_P - \theta_L)\} - R_3\phi(Y_{sat} - Y_A)(\Delta h_V + c_{pV}\theta_A) + S_2\dot{m}_{LV} \right\} |_{\theta_L=\chi}, \\ &= \left\{ Q_2\{\phi(\theta_A - \chi) + f_\alpha\phi(\theta_P - \chi)\} - R_3\phi(Y_{sat} - Y_A)(\Delta h_V + c_{pV}\theta_A) + S_2\dot{m}_{LV} \right\}. \end{aligned}$$

A necessary condition that I is an invariant region is

$$\begin{aligned} \nabla G_9 \cdot F_4 &= \left\{ Q_2\{\phi(\theta_A - \chi) + f_\alpha\phi(\theta_P - \chi)\} - R_3\phi(Y_{sat} - Y_A)(\Delta h_V + c_{pV}\theta_A) + S_2\dot{m}_{LV} \right\} \leq 0, \end{aligned}$$

$$\begin{aligned}
R_3\phi(Y_{sat} - Y_A)(\Delta h_V + c_{pV}\theta_A) &\geq Q_2\{\phi(\theta_A - \chi) + f_\alpha\phi(\theta_P - \chi)\} + S_2\dot{m}_{LV}, \\
Y_{sat} - Y_A &\geq \frac{1}{R_3\phi(\Delta h_V + c_{pV}\theta_A)} \left\{ Q_2\{\phi(\theta_A - \chi) + f_\alpha\phi(\theta_P - \chi)\} + S_2\dot{m}_{LV} \right\}, \\
Y_A &\leq Y_{sat} - \frac{1}{R_3\phi(\Delta h_V + c_{pV}\theta_A)} \left\{ Q_2\{\phi(\theta_A - \chi) + f_\alpha\phi(\theta_P - \chi)\} + S_2\dot{m}_{LV} \right\} \quad \text{at } \theta_L = \chi.
\end{aligned} \tag{2.36}$$

This is an extra condition on the solution. If we set $G_{10} = \theta_P - \chi$, then this implies

$$\begin{aligned}
\nabla G_{10} \cdot F_5 |_{\theta_P = \chi} &= Q_3\{(1 - \phi)(\theta_A - \theta_P) - f_\alpha\phi(\theta_P - \theta_L)\} |_{\theta_P = \chi}, \\
&= Q_3\{(1 - \phi)(\theta_A - \chi) - f_\alpha\phi(\chi - \theta_L)\}.
\end{aligned}$$

A necessary condition that I is an invariant region is

$$\begin{aligned}
Q_3\{(1 - \phi)(\theta_A - \chi) - f_\alpha\phi(\chi - \theta_L)\} &\leq 0, \\
f_\alpha\phi(\chi - \theta_L) &\geq (1 - \phi)(\theta_A - \chi), \\
\chi - \theta_L &\geq \frac{1}{f_\alpha\phi}(1 - \phi)(\theta_A - \chi), \\
\theta_L &\leq \chi - \frac{1}{f_\alpha\phi}(1 - \phi)(\theta_A - \chi) \quad \text{at } \theta_P = \chi.
\end{aligned} \tag{2.37}$$

This is an extra condition on the solution. We derived some necessary conditions which must hold if I is an invariant region. In order to obtain invariant region for the full system (2.30) above conditions should satisfy. Accordingly, we can make following lemma.

Lemma 2.5.3. *Let I is defined as in (2.33) and*

$$\begin{aligned}
Y_A &\geq Y_{sat} - \frac{1}{R_3\phi(\Delta h_V + c_{pV}\theta_A)} \left\{ Q_2\{\phi\theta_A + f_\alpha\phi\theta_P\} + S_2\dot{m}_{LV} \right\} \quad \text{at } \theta_L = 0, \\
Y_A &\leq Y_{sat} - \frac{S_1}{R_2}\dot{m}_{LV} \quad \text{at } \phi = 1, \\
Y_A &\leq Y_{sat} - \frac{1}{R_3\phi(\Delta h_V + c_{pV}\theta_A)} \left\{ Q_2\{\phi(\theta_A - \chi) + f_\alpha\phi(\theta_P - \chi)\} + S_2\dot{m}_{LV} \right\} \quad \text{at } \theta_L = \chi, \\
\theta_L &\leq \chi - \frac{1}{f_\alpha\phi}(1 - \phi)(\theta_A - \chi) \quad \text{at } \theta_P = \chi.
\end{aligned}$$

Under these conditions, the system (2.30) has an invariant region.

Proof. Proof of this lemma is given by the above derivations. □

In our computation we checked these conditions.

Chapter 3

Model Equations in Intracellular Calcium Dynamics

”Almost everything that we do is controlled by Ca^{2+} - how we move, how our hearts beat and how our brains process information and store memories”, see Berridge et al. [17]. The above statement illustrates the universal significance of Ca^{2+} in cell signalling. To do all of this, Ca^{2+} acts as an intracellular messenger, relaying information within cells to regulate their activity. Calcium signalling is an important part of cellular information processes. It regulates multiple cellular functions such as gene expression, secretion, muscle contraction or synaptic plasticity. The Ca^{2+} signal employed by a variety of processes is a transient increase of the concentration in the cytosol [16, 18, 74]. Increase of $[\text{Ca}^{2+}]$ is due to entry through the cell membrane or to Ca^{2+} release from internal storage compartments, specifically the endoplasmic reticulum (ER) and the sarcoplasmic reticulum. It leads to the formation of spatio-temporal signals in the form of waves of high Ca^{2+} concentration traveling across the cell [76, 65, 35] and the global oscillations [15, 78]. In general, the information transmitted by these signals arrives as a stimulus at the plasma membrane and is translated into intracellular Ca^{2+} oscillations by well known pathways.

Ca^{2+} performs its signaling tasks by a transient rise in the cytosol Ca^{2+} concentration. It increases for a short time and drops again. This may be a single event or may result in the type of periodic patterns which are shown in Figure 3.1. It depicts oscillations of the Ca^{2+} concentrations in hepatocytes, i.e. liver cells. The first train of oscillations are caused by treating the cells with adenosine triphosphate (ATP). As soon as the cells are exposed to PPADS (pyridoxalphosphate-6-azophenyl-2',4'-disulphonic acid) the pattern of the oscillations changes. The amplitude decreases, whereas the frequency increases. This is a typical example for a cellular response. The opening and closing of a single channel

is essentially a stochastic process. Therefore, it is important for understand of calcium signalling, to develop accurate models for the stochastic transitions of single channel states. Recordings of channel currents are then evaluated for open probabilities, mean open and mean closed times. These data then lead to models of Ca^{2+} and IP_3 binding such as the DeYoung-Keizer (DYK) model. Here we will adopt a model which is similar to the DYK model, but contains a further conformational change associated to the opening of the channel. One can use a standard method to compile a list of all transitions of the channel (such as those of the DYK-type model) and to determine a sufficiently small time step width δt . One can then determine the occurrence of each of the transitions during a specific time step by comparing a computer random number with the product of the corresponding rate and δt . Here we used more efficient method, called the Gillespie algorithm [37]. This method determines the time of each transition by using a random number. Thus it needs a number of random numbers which are equivalent to the number of transitions, which are far less than for the standard method.

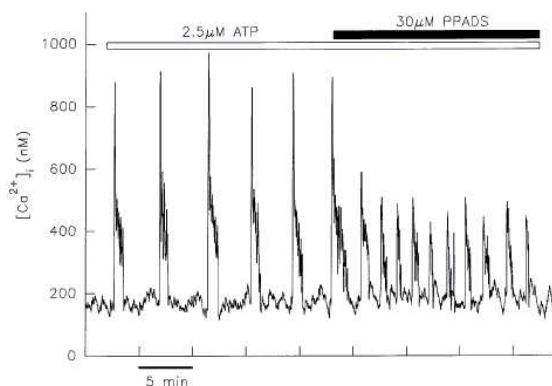


Figure 3.1: ATP-induced Ca^{2+} oscillations in rat hepatocyte. Figure from Dixon et al. [30]

We organized this chapter as follows: In first section we give brief introduction regarding intracellular Ca^{2+} dynamics. In second section we present the mathematical modeling equations in two and three dimensions. This we explain the choice of parameter values. The subsequent section will give the introduction to different stochastic models and the hybrid stochastic-deterministic algorithm used for our simulation results. Further we explain the DeYoung-Keizer (DYK) model for channel dynamics and the Gillespie method which determine when the next event will occur and what kind of event it is. Simulation results are shown in Chapter 6.

3.1 Introduction to intracellular Ca^{2+} dynamics

Intracellular Ca^{2+} dynamics is the dynamics of Ca^{2+} transport through the plasma membrane, release from and uptake by intracellular stores and binding to buffer proteins. Release is determined by the conduction properties of the release channels and the diffusion characteristics of the storage compartments and the receiving compartments. Opening and closing of Ca^{2+} channels control the release. These channels are packed into clusters on the membrane of intracellular storage compartments like the endoplasmic reticulum (ER) or the sarcoplasmic reticulum (SR) containing 10-50 channels, see Figure 3.2. The maximal number of channels in a cluster is not very well known but it is estimated to be in the range of 20-30, see [26, 32]. These clusters in turn are randomly scattered across the ER membrane. The average distance of clusters is typically larger than the Ca^{2+} diffusion length, see [65, 88]. Stochastic behavior, i.e. random opening and closing of channels, manifests itself as spontaneous release events through single channels or several channels in a cluster, see [21, 65].

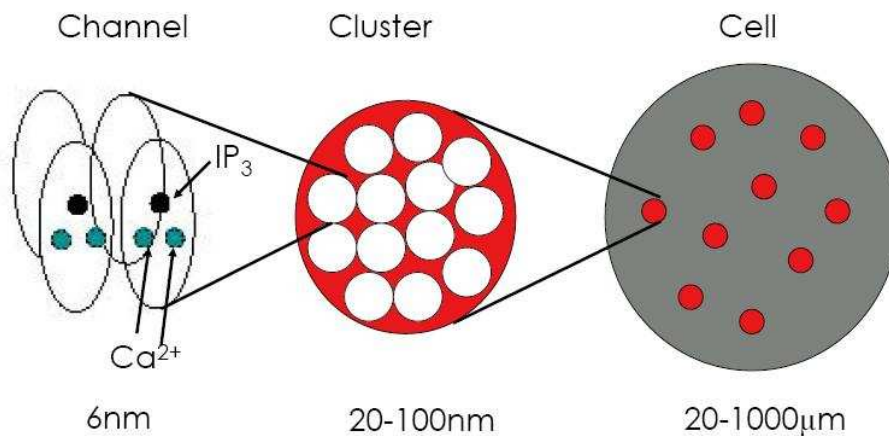


Figure 3.2: The structure of clusters and channels.

A channel type, which control the release of Ca^{2+} , present in the ER membrane of many cells is the inositol 1,4,5-trisphosphate (IP_3) receptor channel (IP_3R). The opening probability of the IP_3R depends on the Ca^{2+} on the cytosolic side of the channel and the IP_3 concentration, see [87, 74] for reviews. It increases nonlinearly with the IP_3 concentration and the Ca^{2+} concentration. Thus, Ca^{2+} released by one channel diffuses in the cytosol

and thus increases the opening probability of neighboring channels. This provides a self-amplifying release mechanism and the coupling of channels by Ca^{2+} diffusion causes the spatial spread of release, which is called Calcium Induced Calcium Release (CICR). The IP_3R channel releases Ca^{2+} from the endoplasmic reticulum (ER) upon an increase of IP_3 concentration in the cytosol, see Figure 3.3. Further, remarkable feature of the conductive channel property is the strong nonlinearity of the Ca^{2+} feedback. For large Ca^{2+} concentrations, which occur in the vicinity of an open channel, Ca^{2+} inhibits its own release. Another element of intracellular Ca^{2+} handling are buffers. Buffers are proteins binding most of the Ca^{2+} in a cell (up to 99%). They present in the cytosol as well as in ER and other storage compartments. Buffers are considered as mobile or immobile depending on their diffusion characteristics. The rate constants of Ca^{2+} binding and dissociation allow for a distinction between slow and fast buffers. Buffering of most of the free Ca^{2+} is one of the basic phenomena in intracellular Ca^{2+} dynamics. It influences the time scales and sets the diffusion length scales.

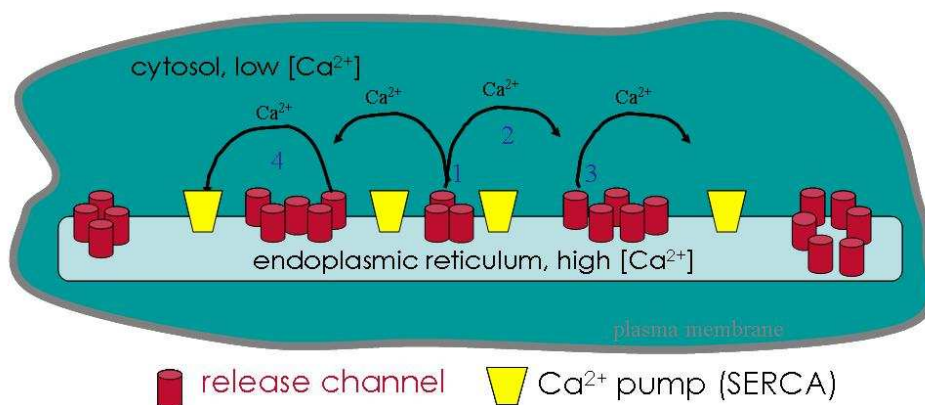


Figure 3.3: The structure of clusters and channels.

The dependence of the opening probability of the release channels on cytosolic Ca^{2+} creates communication between channels and allows for the formation of spatio-temporal patterns of intracellular Ca^{2+} release. These patterns show a hierarchy of the phenomena. The elementary event of cytosolic Ca^{2+} dynamics is the opening of a single channel with the ensuing Ca^{2+} release is called *blip*. The next larger event is a *puff* and a puff is the opening of several closely packed channels. Puffs are generic elements of Ca^{2+} signalling, see [14, 91]. They can cooperate to set off a wave traveling through the cell.

3.2 Governing equations

3.2.1 Deterministic equations in 2D

In a cell, the Ca^{2+} is transported through channels and by pumps, diffuses in the cytosol as well as in the endoplasmic reticulum and reacts with buffers. A stationary profile can be reached by assuming the removal of Ca^{2+} from the cytosol by pumps or imposing boundary conditions, which guarantee that the current entering through the channel is equal to the current leaving through the volume surface. The release process is described by the reaction-diffusion equations for the cytosolic Ca^{2+} concentration c and the Ca^{2+} concentration E in the ER as well as the buffer concentrations b_i and $b_{E,j}$, in the cytosol and ER respectively. We have $i = s, d, m$ and $j = s, m$, where s denotes a stationary, d a dye and m a mobile buffers. These equations are in cartesian coordinates. As a simplification we do not consider the full three-dimensional cytosolic and ER space in this subsection but instead consider thin sheets below and above an idealized planar ER membrane of finite extension. All concentrations are therefore two-dimensional in space. More details regarding 2D modeling can be found in Falcke [34].

The equations include diffusion of free Ca^{2+} described by $D\Delta c$, diffusion of dye buffers b_d denoted by $D_d\Delta b_d$, diffusion of mobile buffers b_m described $D_m\Delta b_m$ and the reactions of stationary buffer b_s , dye buffer b_d and mobile buffer b_m with free Ca^{2+} given by $k_i^+(B_i - b_i)c - k_i^-b_i$ where $i = s, d, m$. The total concentration of stationary, dye and mobile buffer B_i , $i = s, d, m$ is usually homogeneous before the experiments begin. Therefore, the concentration of free buffer, i.e. buffer with no Ca^{2+} bound, can be expressed as $(B_i - b_i)$ at any point in space. The reaction diffusion equations in 2 space dimensions are

$$\frac{\partial c}{\partial t} = D_c\Delta c + (P_l + P_c(r))(E - c) - P_p\frac{c^2}{K_d^2 + c^2} - \sum_i H_i(c, b_i), \quad (3.1)$$

$$\frac{\partial E}{\partial t} = D_E\Delta E + \gamma \left[(P_l + P_c(r))(E - c) - P_p\frac{c^2}{K_d^2 + c^2} \right] - \sum_j K_j(c, b_{E,j}), \quad (3.2)$$

$$\frac{\partial b_i}{\partial t} = D_{b,i}\Delta b_i + H_i(c, b_i) \quad i = s, m, d, \quad (3.3)$$

$$\frac{\partial b_{E,j}}{\partial t} = D_{E,j}\Delta b_{E,j} + K_j(E, b_{E,j}) \quad j = s, m. \quad (3.4)$$

The transport through the ER membrane comprises three contributions. Calcium is moved from the ER into the cytosol through a leak current $P_l(E - c)$, and the channels $P_c(r)(E - c)$. The latter term will be discussed in more detail below. Calcium is resequenced into the

ER by pumps modeled by term proportional to P_p . The action of pumps was found to be cooperative in calcium. The parameter K_d is the dissociation constant of the pumps.

The term proportional to P_c in Eqs. (3.1) and (3.2) models the current through an open channel. This current was found to depend on the cross-membrane difference. For differences found in cell-physiological conditions, the current can be approximated by a linear dependence on $(E - c)$. The current is modeled as a source with constant density in a specified channel cluster region. A model to calculate the cluster radius is proposed by Thul and Falcke [90]. The radius R_n of the cluster n with $N_{\text{open},n}$ open channels is then determined by

$$R_n = R_S \sqrt{N_{\text{open},n}}.$$

The position of the cluster is given by a fixed position \mathbf{x}_n . Then the flux term is given by

$$P_c(\mathbf{r}_n) = \begin{cases} P_{ch} & \text{if } \|\mathbf{r}_n - \mathbf{x}_n\| < R_n \text{ for a cluster } i, \\ 0 & \text{otherwise.} \end{cases}$$

Note that in a model including the dynamics of channel gating the number of open channels is time-dependent. Here, however the number of open channels $N_{\text{open},n}$ is determined by stochastic channel dynamics during a simulation. The corresponding value of P_{ch} can be found in Table B.1.

The amount of buffer in the cytosol and the ER that is bound to calcium is given by b_i or $b_{E,j}$, respectively. All buffers are assumed to be distributed homogeneously in the initial state. Immobile buffers are modeled by setting their diffusion coefficient to zero. Total buffer concentrations in the cytosol and the ER are denoted by B_i or G_j , respectively. Experimentally, the total amount in some buffers is known quite well. However, the amount of some other buffers such as the stationary buffer, comprising contributions from different calcium stores such as mitochondria, is not well known. The buffer binding and unbinding of calcium is modeled by the usual mass-action kinetic terms

$$H_i = k_{b,i}^+(B_i - b_i)c - k_{b,i}^-b_i, \quad (3.5)$$

$$K_j = k_{E,j}^+(G_j - b_{E,j})E - k_{E,j}^-b_{E,j}. \quad (3.6)$$

3.2.2 Deterministic equations in 3D

We want to simulate the release of Ca^{2+} in a cube volume divided by the luminal membrane perpendicular to the cylinder axis. The smaller part represents the ER and the larger

part, the cytosol. The channel is a pore in the center of the ER membrane with radius R_s , see Figure 3.4. The initial condition is the stationary Ca^{2+} - distribution resulting from the pumps and the leak flux P_l . No flux boundary conditions were applied at the outer surface of the cylinder. We chose cartesian coordinates for our simulations with the positive z-direction pointing from top to bottom in Figure 3.4. For more details regarding 3D modeling can be found in Thul and Falcke [90], Thul [89].

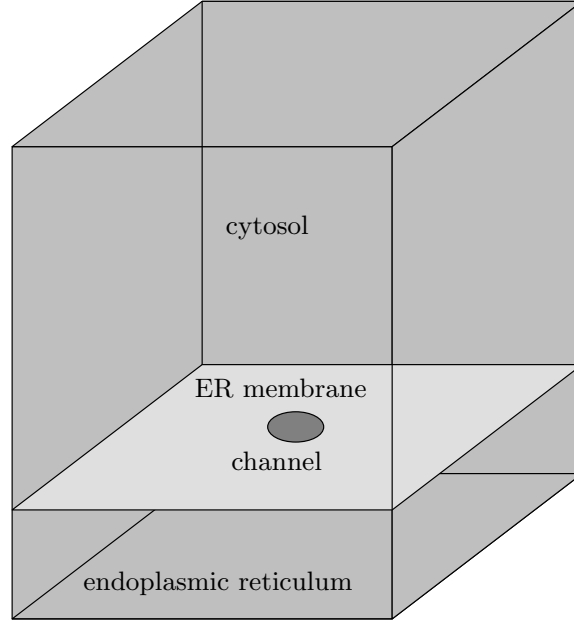


Figure 3.4: Volume within which release was simulated.

The reaction-diffusion equations in the 3 dimensions are

$$\frac{\partial c}{\partial t} = D_c \Delta c - \sum_i H_i(c, b_i), \quad (3.7)$$

$$\frac{\partial E}{\partial t} = D_E \Delta E - \sum_j K_j(c, b_{E,j}), \quad (3.8)$$

$$\frac{\partial b_i}{\partial t} = D_{b,i} \Delta b_i + H_i(c, b_i), \quad i = s, m, d \quad (3.9)$$

$$\frac{\partial b_{E,j}}{\partial t} = D_{E,j} \Delta b_{E,j} + K_j(E, b_{E,j}). \quad j = s, m \quad (3.10)$$

The major difference lies in the flux in these equations as compared to two dimensional equations, which was a source in 2D and now becomes a boundary condition. The flux J

through the membrane separating ER and cytosol is given by

$$J = (P_l + P_c(r))(E - c) - P_p \frac{c^2}{K_d^2 + c^2}, \quad r \leq R. \quad (3.11)$$

Here, the values of E and c have to be taken at the membrane. The *cluster radius* is denoted by R , P_l is the coefficient of the *leak flux density* and P_p the *maximal pump strength*. The constants in Eq. (3.11) will be determined in the next section. The currents are incorporated into the volume dynamics by setting the boundary condition at the ER membrane to

$$D \frac{\partial c}{\partial z} = -\frac{1}{\gamma} D_E \frac{\partial E}{\partial z} = J. \quad (3.12)$$

3.3 Stochastic behavior of intracellular Ca^{2+} dynamics

The process causing random behavior in intracellular Ca^{2+} dynamics is the transition between the different states of the channel subunits and the channel. Channels open and close randomly. The opening and closing probability depends on the state of the channel subunits. The opening probability is the highest, if a minimum number of subunits are activated and very low otherwise. The findings of Mak et al. [63, 64] suggest that the change of opening probability is due to a change of the average opening rate whereas the closing rate is more or less constant. In the following, an event resulting in an opening or closing of the channel will be called *channel transition*. In the following subsections we study the fascinating behavior of the intracellular Ca^{2+} dynamics and stochastic models.



Figure 3.5: Cryo-electron microscopic images of purified type I IP_3 receptor from a mouse cerebellum. Figure from Jiang et al. [54].

3.3.1 Stochastic channel model

In this subsection, the brief introduction of stochastic model for the gating of subunits is explained. This model is based on the DeYoung-Keizer (DYK) model for the subunit dynamics, see [29], we will briefly discuss this model here. It is known that a subunit consists of binding sites for Ca^{2+} and IP_3 . However, the exact number of binding sites is still under investigation. Based on the results of Bezprozvanny et al. [19], DeYoung and Keizer [29] proposed a model for a single subunit. The model by DeYoung and Keizer was set up as a deterministic model and used later on as a stochastic scheme by Falcke et al. [34, 36]. The subunit consists of three binding sites: an activating and an inhibitory Ca^{2+} as well as an activating IP_3 binding sites. Therefore the state of a subunit can be specified by a binary triplet ijk . The indices are represented by the IP_3 binding site, the Ca^{2+} activating and the Ca^{2+} inhibiting binding site respectively. An index is equal to 1 if an ion is bound and 0 if not. Hence, for example, the state 110 refers to IP_3 and Ca^{2+} bound to the activating sites, respectively, and no Ca^{2+} attached to the inhibiting binding site. We assume that the channel is open, if at least three of the subunits are activated, i.e. they have bound Ca^{2+} and IP_3 at the activating site. The resulting nine states of a subunit are shown in Figure 3.6. In 2D numerical simulations, the standard DYK model is considered for a subunit dynamics which consists of 8 different states, i.e. see Figure 3.6 with excluding the "open" state. An additional state, called "open", is introduced to DYK model which is considered for subunit dynamics in our 3D numerical simulations. In Figure 3.6, the binding rate constants for IP_3 activation are given by a_1p and a_3p , whereas a_2c and a_4c refer to Ca^{2+} inhibition. The activation of Ca^{2+} is controlled by a_5c . The dissociation rates for the above processes are denoted by b_1 through b_5 . The reactions that occur at a subunit are binding and unbinding of Ca^{2+} and IP_3 . They determine the state of one subunit. In an ensemble of subunits these processes lead to a fraction X_{ijk} of subunits in a state ijk . If the ensemble is large enough and homogeneous, these fractions can be described by rate equations. For instance, the time evolution of X_{110} is governed by

$$\dot{X}_{110} = -[b_5 + a_2c + b_1] X_{110} + a_5cX_{100} + b_2X_{111} + a_1pX_{010}, \quad (3.13)$$

with p being the IP_3 concentration and c the Ca^{2+} concentration. The negative term represents the processes that reduce the value of X_{110} . This can result from unbinding of IP_3 with rate b_1 , unbinding from the activating Ca^{2+} site with rate b_5 and binding to the inhibiting Ca^{2+} binding site with rate a_2c . The remaining three terms control the increase of X_{110} . This happens for example through binding with rate a_5c to the activating Ca^{2+}

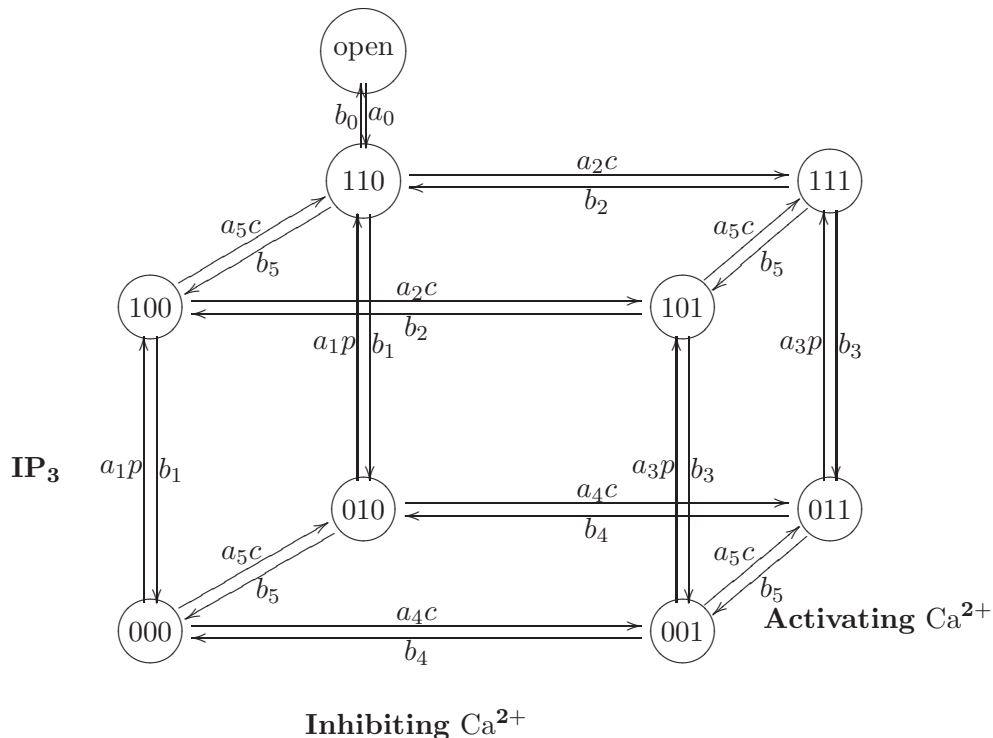


Figure 3.6: Transition scheme of the DeYoung-Keizer model

site of a subunit that is in the state 100. Together with the remaining rate equations, the state of the ensemble is fully characterized. We may discard one of these equations and use instead the conservation law

$$\sum_{\{ijk\} \in [0,1]^3} X_{ijk} = 1. \quad (3.14)$$

It states that each subunit belongs to one of the fractions X_{ijk} and that the number of subunits is conserved. In general, the Ca^{2+} concentration is not constant in time, so that a closed solution for the fractions X_{ijk} is not accessible.

The binding and dissociation of Ca^{2+} and IP_3 as well as the conformational change are stochastic events rendering the opening and closing of the channel a stochastic process. That stochastic process is coupled to the concentration of cytosolic Ca^{2+} since the binding probabilities per unit time depend on it and vice versa the number of open channels determines the concentration fields.

The state of active elements can be described as either activated or deactivated. As their

number per cluster is rather small and as they are tightly packed, each element occupies a non negligible spatial fraction. Therefore the state of a cluster is well characterized by the area occupied by activated units. We refer to it as the active area of a cluster. Usually this is not a connected patch. Its size equals the sum of the areas of all activated units.

In this work the stochastic solver is based on the Gillespie method [38]. The brief explanation of this method is presented here. We start by establishing out notation and briefly reviewing the details of stochastic chemical kinetics relevant to our work. According to the Gillespie method, suppose the volume V contains a spatially homogeneous mixture of X_j molecules of chemical species S_j , $j = 1, \dots, N$, that can interact through M specified reaction events R_i , $i = 1, \dots, M$. The state of the system, (X_1, \dots, X_N) , consists of the *numbers* of molecules of the chemical species. Then we may assert the existence of M constants r_i , $i = 1, \dots, M$, which depend only on the physical properties of the molecules, such that

$$\begin{aligned} r_i dt &= \text{average probability that a particular combination of } R_i \text{ events} \\ &\text{will react accordingly in the next infinitesimal time interval } dt. \end{aligned} \quad (3.15)$$

By "average" here mean that, if we multiply $r_i dt$ by the total number of distinct combinations of R_i reaction events in V at time t , we will obtain the probability that an R_i event will occur somewhere inside V in the next infinitesimal time interval $(t, t + dt)$.

The reaction probability density function: If we are given that the system is in the state (X_1, \dots, X_N) at time t , then essentially all we need to know "when will be the next event take place and what kind of event will it be?". These are two important questions that arise due to the stochastic nature of the reactions. One can expect that these two questions will be answered in some "probabilistic" sense. According to the Gillespie method [38], the probabilistic function defined by

$$\begin{aligned} P(\tau, i) dt &= \text{probability that, given the state } (X_1, \dots, X_N) \text{ at time } t, \text{ the} \\ &\text{next event in } V \text{ will occur in the infinitesimal time interval} \\ &(t + \tau, t + \tau + dt), \text{ and will be an } R_i \text{ event.} \end{aligned} \quad (3.16)$$

We call $P(\tau, i)$ the *reaction probability density function* on the space of the continuous variable τ ($0 \leq \tau < \infty$) and the discrete variable i , $i = 1, \dots, M$. The probability that the event R_i occurs in the next infinitesimal time interval $[t, t + \tau]$ is given in terms of the propensity function and equals $a_i(t, c)dt$. The reaction propensities, a_i , are the probabilistic

rates of the reactions. In our model the propensities are proportional to the occupation number h_i of the corresponding subunit state, defined as

$$h_i = \text{number of distinct } R_i \text{ event combinations in} \\ \text{the state } (X_1, \dots, X_N).$$

Hence we set $a_i = h_i r_i$, where r_i is defined as above Eq. (3.15). Given the actual time t , the probability that the next stochastic event occurs in the infinitesimal time interval $[t + \tau, t + \tau + dt]$ and is an R_i event is given by

$$P(\tau, i)dt = \begin{cases} a_i \exp(-a_0 \tau) dt & \text{if } 0 \leq \tau < \infty \text{ and } i = 1, \dots, M, \\ 0 & \text{otherwise.} \end{cases} \quad (3.17)$$

where $a_i = h_i r_i$, where $i = 1, \dots, M$ and $a_0 = \sum_j^M a_j$ is the sum of all propensities. One can find a more detailed derivation of this function in [37].

In the above we observed that essentially what is needed to simulate the time evolution of a chemically reacting system is some way of specifying when the next reaction event will occur and what kind of event it will be. To give a more precise mathematical meaning, what is needed is a method for generating a pair (τ, i) from the set of random pairs whose probability density function is $P(\tau, i)$ like the one given in Eq. (3.17). To generate the pair (τ, i) based on the Gillespie method [37], called "direct" method. It is based on the fact that any two-variable probability density function can be written as the product of two one-variable probability density functions, a procedure known as "conditioning", in the following form

$$P(\tau, i) = P_1(\tau) \cdot P_2(i|\tau). \quad (3.18)$$

Here, $P_1(\tau)$ is the probability that the next event will occur between times $t + \tau$ and $t + \tau + dt$, and $P_2(i|\tau)$ is the probability that the next event will be an R_i event, given that the next event occurs at time $t + \tau$. These two equations express the two one-variable density functions

$$P_1(\tau) = a_0 \exp(-a_0 \tau) \quad \text{if } 0 \leq \tau < \infty \quad (3.19)$$

$$P_2(i|\tau) = \frac{a_i}{a_0} \quad \text{if } i = 1, \dots, M. \quad (3.20)$$

Note that both of these one-variable density functions are properly normalized i.e. $\int_0^\infty P_1(\tau) d\tau = 1$ and $\sum_{i=1}^M P_2(i|\tau) = 1$.

However, our objective in the stochastic simulation algorithm to generate a random pair (τ, i) according to the probability density function in Eq. (3.17). From this method first generate a random value τ according to Eq. (3.19) by simply drawing a random number r_1 from the uniform distribution in the unit interval and taking

$$a_0 \cdot \tau = \ln(1/r_1). \quad (3.21)$$

Then we generate a random integer i according to Eq. (3.20) by drawing a random number r_2 from the uniform distribution in the unit interval and taking

$$\sum_{j=1}^i a_j \leq a_0 \cdot r_2 < \sum_{j=1}^{i+1} a_j. \quad (3.22)$$

In this way we can find the next event R_i and it will occur after time τ .

3.4 Hybrid stochastic and deterministic model

The Gillespie method based on the assumption that during successive stochastic events the propensities a_i do not change. Indeed, over those successive stochastic events, there must be a significant activity in all reaction channels. However, when linking the stochastic channel dynamics to the calcium dynamics, we expect the propensity a_i to change in time due to its dependence on the local calcium concentration c . This effect will be particularly strong for openings and closings of channels, since after such events the local calcium concentration c changes dramatically by orders of magnitude. So the propensities can change too rapidly over small time intervals. Moreover, purely stochastic simulations are computationally very expensive when there are many reactions involved in chemical system. There are some efficient methods available in literature. Still this area is very active in the current research.

To overcome those problems, we adopted a hybrid method which is recently introduced by Alfonsi et al. [2]. In their hybrid algorithm, the stochastic reaction equations are partitioned into deterministic and stochastic equation, to reduce the computational time and increase the efficiency. To adapt this hybrid algorithm to current problem, we used the spatial-temporal equations are deterministic and the opening/closing of channels are considered as stochastic part. Here we will give the brief explanation of the hybrid method. The reaction probability for each event R_i is specified in terms of the propensity function $a_i = a_i(t, c(t))$. Based on physical laws and the idea that chemical reactions are essentially

random processes, the stochastic formulation of chemical reactions is given in terms of a Markov jump process $c(t) \in \mathbb{N}^N$, see [39]. Its characterization is based on the probability $a_i(t, c)dt$ of event R_i occurring in the next infinitesimal time interval $[t, t + dt]$. Denoting by $T_i(t)$ the time at which event R_i first occurs after t , this can be written as

$$\mathbf{P}[T_j(t) \in [t, t + dt] | c(t)] = a_j(c(t), t)dt.$$

More details can be found in Alfonsi et al. [2]. Within their setting the time τ to the next stochastic event is determined by solving

$$g_i(t + \tau | t) = \int_t^{t+\tau} a_i(c(t), s) ds = \xi, \quad (3.23)$$

with $\xi = \ln(1/r_1)$, where the sum of propensities a_0 may explicitly depend both on time and the local calcium concentration. The function $g_i(t + \tau | t)$ is non-decreasing for $t + \tau > t$, since the propensities a_i are non-negative by definition. Note that the above equation simplifies to the equation determining τ in Eq. (3.21) in the case of constant a_0 . To determine the time of next reaction τ , condition Eq. (3.23) is conveniently rewritten in differential form by introducing a variable $g(t)$ and solving

$$\dot{g}(s) = a_0(c, s). \quad (3.24)$$

with initial condition $g(0) = 0$, along with the deterministic equations for c and buffers. This is the stochastic equation considered in our numerical simulations. To calculate the propensities we follow the dynamics of DYK model. A reaction then occurs whenever $g(s)$ reaches the value ξ . As before, the specific event R_i is determined based on a second random number r_2 solving Eq. (3.22) with propensities evaluated at the event time $t + \tau$. This way one can determine the next reaction event and when it occurs.

we would like to give the brief outline of the algorithm here. A special feature of the calcium system is that not all stochastic events change the open/close state of a channel. A channel transition has a major impact on the local calcium concentration c , while non-channel transitions do not change the local calcium concentration. During the computation of the deterministic part of the calcium dynamics the stochastic events are traced via Eq. (3.23) respectively Eq. (3.24). During the simulation the stochastic system is updated for every stochastic time step dt . The time step dt is determined using the first random number generation, see Eq. (3.21), and by fulfilling the requirement $a_0 dt \leq 1$, where a_0 is the sum of the propensities. Using the second random number the reaction event R_i

is determined, see Eq. (3.22). In this way we can determine the next reaction event R_i and it will occur after the time τ . If a non-channel transition occurs, the stochastic event is performed. The stochastic channel dynamics is updated correspondingly, while there is no influence on the calcium concentration. On the other hand, if a channel transition takes place, both the channel and the calcium dynamics do change. This typically requires a readjustment of the deterministic time step. The algorithmic realization of our hybrid approach is given in Appendix B, see Algorithm 3.

Note that $\Delta t_{channel}$ should be smaller than or similar to the time scale of stochastic transitions after a channel opening/closing. This is needed since we linearly interpolate the deterministic solution to determine stochastic transitions between succeeding deterministic time steps. Therefore, fast changes of the deterministic variables after a channel opening/closing need to be approximated numerically at time scales comparable to the stochastic transitions. Here we typically use two time steps in the hybrid simulation. One time step is for updating the deterministic solution, which is determined by the adaptive time step criteria of ODE solver and second time step is used for updating stochastic solver, which is calculated by the Gillespie algorithm. The step by step process of the hybrid algorithm can be found in Algorithm 3.

Chapter 4

Discretization of Reaction-Diffusion Systems

In this chapter we explain the discretization and solution of partial differential equations of the type which are given in Chapter 2 and Chapter 3. This chapter is organized as follows, in the first section we give the basic definition of some function spaces and associated norms which are essential in the formulation of the finite element method. In the second section we explain the basic aspects of the finite element method. One of the advantages of the finite element method is that it can be used with relative ease to find approximations to solutions of differential equations in divergence form on general domains. We will first consider a so-called *semi-discrete* analogue of the full system where we have discretized in space using continuous piecewise linear finite elements, explained in third section. We shall see that the semi-discrete problem is an initial value problem for a system of ordinary differential equations. To obtain a fully discrete problem we will then discretize in time, explained in fourth section. In the fifth section we give an introduction to solving the resulting algebraic equations. In the sixth section we explain the adaptive grid refinement and the Z^2 error estimator. Finally, the basic idea of domain decomposition methods is explained.

4.1 Mathematical notations and function spaces

The process of spatial discretization by the finite element method is based on the discrete representation of a weak integral form of the partial differential equations to be solved. The formulation and subsequent discretization of such an integral form requires the definition of some function spaces and associated norms. Standard books on mathematical concepts,

such as, for instance, Adams [1], Nair [72], provide a detailed exposition of the mathematical concepts, which are the basis of the finite element method.

Consider a spatial domain $\Omega \subset \mathbb{R}^d$ with piecewise smooth boundary Γ . Here, $d = 1, 2$ or 3 denotes the number of space dimensions. We shall use the notation

$$f : \bar{\Omega} \longrightarrow \mathbb{R}$$

to state that for each spatial point $\mathbf{x} \in \bar{\Omega}$, $f(\mathbf{x}) \in \mathbb{R}$. The set $\bar{\Omega}$ denotes the closure of Ω , that is the union of the domain Ω with its boundary Γ : $\bar{\Omega} = \Omega \cup \Gamma$.

Definition 4.1. A function $f : \Omega \longrightarrow \mathbb{R}$ is said to be of class $C^m(\Omega)$ if all of its derivatives on Ω up to order m exist and are continuous functions. For instance, the notation $f(\mathbf{x}) \in C^m(a, b)$ indicates that $f(\mathbf{x})$ possesses m continuous derivatives for $\mathbf{x} \in (a, b)$.

In finite element analysis we work with functional equations in integral and, thus, we are interested in functions belonging to larger spaces than C^m . As we will see, instead of requiring the m -th derivative to be continuous, we will require that first derivatives are square integrable. In fact, finite element functions should possess generalized derivatives, i.e., derivatives in the sense of distributions, and some integrability properties. Such classes of functions are particular examples of Sobolev function spaces.

Definition 4.2. Let $\Omega \subset \mathbb{R}^d$ the class of all measurable functions u

$$L^p(\Omega) = \left\{ u : \Omega \longrightarrow \mathbb{R} \mid \int_{\Omega} |u|^p \, d\mathbf{x} < \infty \right\}, \quad 1 \leq p < \infty,$$

and

$$L^\infty(\Omega) = \left\{ u : \Omega \longrightarrow \mathbb{R} \mid \text{ess sup}_{\mathbf{x} \in \Omega} |u(\mathbf{x})| < \infty \right\}.$$

This space is equipped with norm

$$\|u\|_{L^p(\Omega)} := \left(\int_{\Omega} |u|^p \, d\mathbf{x} \right)^{1/p}, \quad 1 \leq p < \infty,$$

and

$$\|u\|_{L^\infty(\Omega)} := \text{ess sup}_{\mathbf{x} \in \Omega} |u(\mathbf{x})|.$$

Definition 4.3. If $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}_0^d$ is a d -tuple of nonnegative integers α_j , we call α a *multi-index* and denote by x^α the monomial $x_1^{\alpha_1} \dots x_d^{\alpha_d}$. A multi-index has the *degree* $|\alpha| = \sum_{j=1}^d \alpha_j$. We define the derivative

$$D^\alpha u = \partial_{x_1}^{\alpha_1} \dots \partial_{x_d}^{\alpha_d} := \frac{\partial^{|\alpha|} u}{\partial^{\alpha_1} x_1 \dots \partial^{\alpha_d} x_d}.$$

Next we give a particular class of Sobolev spaces, those of square integrable functions and derivatives.

Definition 4.4. For any nonnegative integer k , we define the **Sobolev space** $H^k(\Omega)$ using multi-index notation

$$H^k(\Omega) = \{u \in L^2(\Omega) \mid D^\alpha u \in L^2(\Omega) \quad \forall |\alpha| \leq k\}.$$

Therefore, $H^k(\Omega)$ consists of square integrable functions all of whose derivatives of order up to k are also square integrable. $H^k(\Omega)$ is equipped with norm

$$\|u\|_k = \left(\int_{\Omega} \sum_{|\alpha| \leq k} |D^\alpha u|^2 dx \right)^{1/2}.$$

Note that $L^2(\Omega)$ is, in fact, a Sobolev space, $H^0(\Omega) = L^2(\Omega)$, while the Sobolev space for $k = 1$ is defined by

$$H^1(\Omega) = \left\{ u \in L^2(\Omega) \mid \frac{\partial u}{\partial x_i} \in L^2(\Omega), i = 1, \dots, d \right\}.$$

This space is equipped with the inner product

$$\langle u, v \rangle_1 = \int_{\Omega} \left(uv + \sum_{i=1}^d \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_i} \right) d\Omega,$$

and its induced norm is

$$\|u\|_1 = \sqrt{\langle u, u \rangle_1}.$$

We shall also frequently use the subspace

$$H_0^1(\Omega) = \{v \in H^1(\Omega) \mid v = 0 \text{ on } \Gamma\},$$

the elements of which possess a square integrable first derivative over the domain Ω and whose trace vanishes on its boundary Γ . Moreover, its inner product and norm coincide with those of $H^1(\Omega)$.

Remark 4.1. H_0^1 is usually defined as the closure of $C_0^\infty(\Omega)$, the set of all continuous functions with continuous derivatives whose support is a bounded subset of Ω , with respect to the norm of $\|\cdot\|_1$. This is, $H_0^1(\Omega)$ is the set of all functions u in $H^1(\Omega)$ such that u is the limit in $H^1(\Omega)$ of a sequence $\{u_s\}_{s=1}^\infty$ where all u_s are in $C_0^\infty(\Omega)$, see Adams [1].

4.2 The basic aspects of the finite element method

In this section we present the properties of the classical finite element approximation. In order to apply the standard Galerkin method, we face, by definition, the problem of constructing finite-dimensional subspaces V_h of spaces V such as $H^1(\Omega), H_0^1(\Omega), \dots$. The *finite element method in its simplest form* is a specific process of constructing subspaces V_h , which are then called *finite element spaces*. This construction is characterized by *three basic aspects*: the existence of a triangulation of the polygonal set Ω , the construction of a finite dimensional subspace consisting of piecewise-polynomials, and the existence of a basis of functions having small support. Here we are giving the definition like Ciarlet [28].

Let us consider $d = 2$. The *first basic aspect* is that a *triangulation* \mathcal{T}_h is established over the set $\bar{\Omega}$, i.e., the set $\bar{\Omega}$ is subdivided into a finite number of subsets T , in such a way that the following properties are satisfied:

$$(T1) \quad \bar{\Omega} = \bigcup_{T \in \mathcal{T}_h} T.$$

(T2) For each $T \in \mathcal{T}_h$, the set is closed and its interior $int(T)$ is nonempty and connected.

(T3) For each distinct $T_1, T_2 \in \mathcal{T}_h$, one has $int(T)_1 \cap int(T)_2 = \emptyset$.

(T4) If $F = T_1 \cap T_2 \neq \emptyset$, T_1 and T_2 distinct elements of \mathcal{T}_h , then F is a common edge, or vertex of T_1 and T_2 .

(T5) $diam(T) \leq h$ for each $T \in \mathcal{T}_h$.

Then \mathcal{T}_h is called a *triangulation* of $\bar{\Omega}$, see Quarteroni and Valli [75], Ciarlet [27].

A *second basic aspect* of the finite element method consists of determining a finite dimensional space V_h which should result in a suitable approximation of the infinite dimensional space V .

Here the point is that the function $v_h \in V_h$ are piecewise-polynomials, i.e., for each $T \in \mathcal{T}_h$ the space

$$P_T := \{v_h|_T | v_h \in V_h\}$$

consists of algebraic polynomials. Analogously for $d = 1$ we would be considering subintervals, for $d = 3$ tetrahedra instead of triangles.

To be more precise, let us denote by $\mathbb{P}_k, k \geq 0$, the space of polynomials of degree less than or equal to k in the variables x_1, \dots, x_d . The dimension of \mathbb{P} is given by, see Braess

[23], Quarteroni and Valli [75]

$$\dim \mathbb{P}_k = \binom{d+k}{k}. \quad (4.1)$$

The first collection of functions, denoted by V , is composed of *test* or *weighing* functions. It consists of all functions which are square integrable, have square integrable first derivatives over the computational domain Ω , and vanish on the Dirichlet portion, Γ_D , of the boundary. It is defined as follows

$$S = \{v \in H^1(\Omega) \mid v = 0 \text{ on } \Gamma_D\} \equiv H_{\Gamma_D}^k(\Omega). \quad (4.2)$$

The second collection of functions is called the *trial* or *admissible solutions*. This collection is similar to the test functions except that these trial functions are required to satisfy the Dirichlet conditions on Γ_D . This space is denoted by S and defined as follows

$$V = \{u \in H^1(\Omega) \mid u = u_D \text{ on } \Gamma_D\} \equiv S + \{\bar{u}_D\}, \quad (4.3)$$

where \bar{u}_D is any function in $H^1(\Omega)$ such that $\bar{u}_D = u_D$ on Γ_D . However, for homogeneous boundary conditions, $u_D = 0$, trial and test spaces coincide, $V = S = H_0^1(\Omega)$.

Now we will define the finite dimensional subspaces $V_h \subset V$ and $S_h \subset S$. The weighting functions $v_h \in S_h$ vanish on Γ_D . The approximation u_h lies in V_h and satisfies, with the precision given by the mesh size h , the boundary condition u_D on Γ_D , for more details see Donea and Huerta [31], Ciarlet [28]. The subspaces are defined as

$$S_h := \{v_h \in H^1(\Omega) \mid v_h|_T \in \mathbb{P}_k \ \forall T \in \mathcal{T}_h \text{ and } v_h = 0 \text{ on } \Gamma_D\}, \quad (4.4)$$

$$V_h := \{u_h \in H^1(\Omega) \mid u_h|_T \in \mathbb{P}_k \ \forall T \in \mathcal{T}_h \text{ and } u_h = u_D \text{ on } \Gamma_D\}. \quad (4.5)$$

The *third basic aspect* of the finite element method is to construct an efficient basis for the space S_h . The basis should have a support as small as possible with little overlap of neighboring basis functions. This leads to sparse linear systems. By denoting, \mathbf{a}_i , $i = 1, \dots, N$, the global set of nodes in $\bar{\Omega}$, is sufficient to choose $\varphi_j \in S_h$, $j = 1, \dots, N$, such that

$$\varphi_j(\mathbf{a}_i) = \delta_{ij} \equiv \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases} \quad (4.6)$$

We see that the *support* of φ_j , the set of points \mathbf{a} for which $\varphi_j(\mathbf{a}) \neq 0$, consists of the triangles with the common node \mathbf{a}_j , the shaded area in Figure 4.1. These functions are called *shape functions*. We refer the interested reader to Ciarlet [28]. Finally, we give the definition of a finite element.

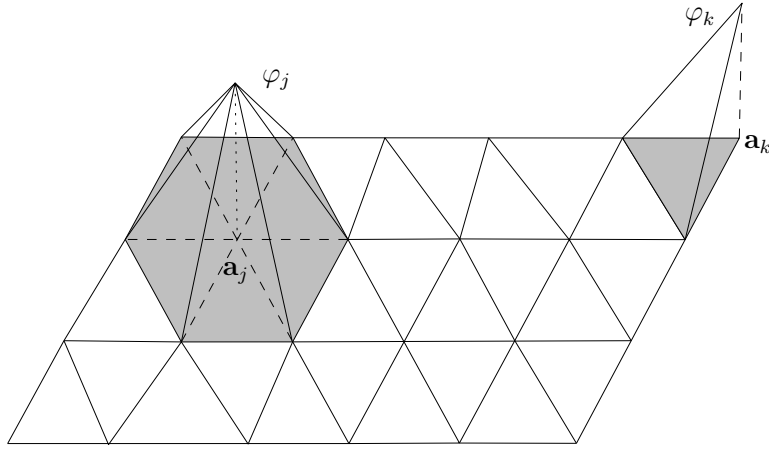


Figure 4.1: The support of basis function

Definition 4.5. A *finite element* in \mathbb{R}^d is a triple (T, P_T, Σ) where

- T is a closed subset of \mathbb{R}^d with a non empty interior and a Lipschitz-continuous boundary, for example a triangle in 2D,
- P_T is a finite-dimensional linear space of functions defined on T ,
- Σ is a set of degrees of freedom,

such that a function $v \in P_T$ is determined by the degrees of freedom Σ .

Definition 4.6. Let h_T be the longest side of a triangle T and ρ_T the diameter of the circle inscribed in T . A family of triangulations \mathcal{T}_h , $h > 0$ is called *regular* if there exists a constant $\sigma \geq 1$ such that

$$\max_{T \in \mathcal{T}_h} \frac{h_T}{\rho_T} \leq \sigma \quad \forall h > 0.$$

This condition means that the triangles $T \in \mathcal{T}_h$ are not allowed to be arbitrarily thin, or equivalently, the angles of the triangles T are not allowed to be arbitrarily small. The constant σ is a measure of the smallest angle in any $T \in \mathcal{T}_h$.

Let \mathbf{a}_i , $i = 1, \dots, N$, be the nodes of \mathcal{T}_h . Given $u \in C^0(\bar{\Omega})$ the *linear interpolant* $I(u) \in V_h$ is defined as the unique piecewise linear function agreeing with u at the nodes of \mathcal{T}_h , i.e.

$$I(u(\mathbf{a}_i)) = u(\mathbf{a}_i) \quad i = 1, \dots, N.$$

The domain $\bar{\Omega}$ is polygonal and $T \in \mathcal{T}_h$. Then to evaluate the integrals one would use a suitably chosen a *numerical quadrature formula* of the form

$$Q(f) := \int_T f(\mathbf{x}) = \sum_{j=1}^q f(\mathbf{x}_{j,T}) w_{j,T} \quad \text{for } f \in C(\bar{\Omega}),$$

where $w_{j,T}$, $j = 1, \dots, q$, are quadrature weights and $\mathbf{x}_{j,T}$ are quadrature points in the element T . We take the quadrature formula has given by

$$Q(f) := \int_{\Omega} I(f) d\mathbf{x} \quad \text{for } f \in C(\bar{\Omega}). \quad (4.7)$$

This corresponds to the trapezoidal rule on each element.

Let T be a d -simplex, and let $\lambda_i(\mathbf{x})$, $1 \leq i \leq d+1$, denote the barycentric coordinates of a point \mathbf{x} with respect to the vertices of the d -simplex. Then for any integer $\alpha_i \geq 0$ one has, see Ciarlet [28]

$$\int_T \lambda_1^{\alpha_1} \dots \lambda_{d+1}^{\alpha_{d+1}} d\mathbf{x} = \frac{\alpha_1! \dots \alpha_{d+1}! d!}{(\alpha_1 + \dots + \alpha_{d+1} + d)!} |T|. \quad (4.8)$$

4.2.1 Mass lumping

The first way to introduce the lumping process is to substitute the mass matrix \mathbf{M} by the matrix $\tilde{\mathbf{M}}$ given by

$$\tilde{\mathbf{M}}_{ij} := \left(\sum_{k=1}^{N_h} \mathbf{M}_{ik} \right) \delta_{ij} = \left[\sum_{k=1}^{N_h} \langle \varphi_i, \varphi_k \rangle \right] \delta_{ij}, \quad (4.9)$$

i.e., $\tilde{\mathbf{M}}$ is the diagonal matrix having the elements $\tilde{\mathbf{M}}_{ii}$ equal to the sum of the elements of \mathbf{M} on the i -th row. Notice at first that trivially $\langle \varphi_i, \varphi_j \rangle_h = 0$ if $i \neq j$, and $\mathbf{a}_i, \mathbf{a}_j$ are not nodes of a common triangle. In that case $\varphi_i \varphi_j$ vanishes at each node of \mathcal{T}_h . Moreover, one can easily check that $\langle \varphi_i, \varphi_k \rangle$ is non-zero only if the nodes \mathbf{a}_i and \mathbf{a}_k belong to the same triangle T . Using Eq. (4.8) we can calculate

$$\int_T \varphi_i^2 d\mathbf{x} = \frac{2! 2!}{(2+2)!} |T| = \frac{1}{6} |T|,$$

and analogously we can calculate

$$\int_T \varphi_i \varphi_k d\mathbf{x} = \frac{1! 1! 2!}{(1+1+2)!} |T| = \frac{1}{12} |T|.$$

The above simple calculation shows that for $\mathbf{a}_i, \mathbf{a}_k \in T$ we have

$$\int_T \varphi_i \varphi_k d\mathbf{x} = \begin{cases} \frac{1}{12} |T|, & i \neq k, \\ \frac{1}{6} |T|, & i = k. \end{cases} \quad (4.10)$$

For each pair $\mathbf{a}_i, \mathbf{a}_k, i \neq k$ there are either exactly two triangles containing \mathbf{a}_i and \mathbf{a}_k or no common triangles. Thus, denoting by \mathbf{D}_i the union of the triangles having \mathbf{a}_i as a vertex, it follows that for a given i

$$\sum_{k \neq i} \langle \varphi_i, \varphi_k \rangle = \sum_{k \neq i} \sum_{T \in \mathcal{T}_h} \int_T \varphi_i \varphi_k \, d\mathbf{x} = \frac{1}{6} |\mathbf{D}_i| .$$

Hence

$$\tilde{M}_{ii} := \sum_{k=1}^{N_h} \langle \varphi_i, \varphi_k \rangle = \frac{1}{3} |\mathbf{D}_i| . \quad (4.11)$$

Analogously, in case of tetrahedra we can show that

$$\tilde{M}_{ii} := \sum_{k=1}^{N_h} \langle \varphi_i, \varphi_k \rangle = \frac{1}{4} |\mathbf{D}_i| . \quad (4.12)$$

4.3 Spatial discretization

The systems of coupled non linear partial differential equations from Chapters 2 and 3 are valid in the domain Ω . The domain $\Omega \subseteq \mathbb{R}^d$ is a convex polygonal subset. The unknowns are functions of space and time with values in $\Omega \times [0, T]$. In this section we give the spatial discretization by the finite element method for solving the coupled partial differential equation system. We will first consider a so-called *semi-discrete* analogue of the full system where we have discretized in space using continuous piecewise linear finite elements. We shall see that the semi-discrete problem is an initial value problem for a system of ordinary differential equations.

Standard finite element texts, such as, for instance, Braess [23], Johnson [55], Shaidurov [83], Quarteroni and Valli [75] provide a detailed exposition of the mathematical concepts, which are the basis of the finite element method.

4.3.1 Semi discretization in space

The system of partial differential equations, which is from Chapters 2 and 3, can be written in the following general form.

$$\begin{aligned} \frac{\partial \mathbf{u}(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\mathbf{A}(\mathbf{x}) \nabla \mathbf{u}(\mathbf{x}, t)) + \mathbf{b}(\mathbf{x}) \cdot \nabla (\mathbf{u}(\mathbf{x}, t)) + \mathbf{r}(\mathbf{u}(\mathbf{x}, t)) &= \mathbf{f}(\mathbf{x}) \quad \text{in } \Omega \times (0, \tau], \\ \mathbf{u}(\mathbf{x}, t) &= \mathbf{u}_0(\mathbf{x}) \quad \text{on } \Omega \times t = 0, \\ \mathbf{u}(\mathbf{x}, t) &= \mathbf{u}_D \quad \text{on } \Gamma_D \times [0, \tau], \\ \mathbf{n} \cdot \mathbf{A}(\mathbf{x}) \nabla \mathbf{u} &= \mathbf{g} \quad \text{on } \Gamma_N \times [0, \tau], \end{aligned} \quad (4.13)$$

where \mathbf{u} is the unknown vector of functions, $\mathbf{A}(\mathbf{x}) > 0$ is the diffusion matrix, $\mathbf{b}(\mathbf{x})$ is the convection velocity, $\mathbf{r}(\mathbf{u})$ is the reaction term and $\mathbf{f}(\mathbf{x})$ is the source function. The function \mathbf{u}_D denotes the prescribed values of \mathbf{u} on the Dirichlet portion Γ_D of the boundary, while the diffusive flux on the Neumann portion Γ_N is \mathbf{g} .

The discretization process using the finite element method is based on a reformulation of the given differential equation in the more general *variational formulation*. Multiplying the above equation for a given time t by $\mathbf{v} \in S_D$, integrating over Ω and using Green's formula, we get the following variational formulation:

Find $\mathbf{u} \in V_D$ s.t: for each $t \in I = [0, T]$

$$\begin{aligned} \langle \frac{\partial \mathbf{u}}{\partial t}, \mathbf{v} \rangle + \langle \mathbf{A}(\mathbf{x}) \nabla \mathbf{u}, \nabla \mathbf{v} \rangle + \langle \mathbf{b}(\mathbf{x}) \cdot \mathbf{u}, \mathbf{v} \rangle + \langle \mathbf{r}(\mathbf{u}), \mathbf{v} \rangle &= \langle \mathbf{f}, \mathbf{v} \rangle + \langle \mathbf{g}, \mathbf{v} \rangle_{\Gamma_N} \quad \text{for all } \mathbf{v} \in S_D, \\ \mathbf{u}(\mathbf{x}, t) &= \mathbf{u}_0(\mathbf{x}) \quad \text{on } \Omega \times t = 0, \\ \mathbf{u}(\mathbf{x}, t) &= \mathbf{u}_D \quad \text{on } \Gamma_D \times [0, T]. \end{aligned} \tag{4.14}$$

Let $V_h \subset V$ and $V_{D_h} \subset V_D$ be finite dimensional subspaces. Replacing the space V by the finite dimensional subspace V_{D_h} we get the following *semi discretization in space*, find $\mathbf{u}_h \in V_h$ s.t:

$$\begin{aligned} \langle \frac{\partial \mathbf{u}_h}{\partial t}, \mathbf{v}_h \rangle + \langle \mathbf{A}(\mathbf{x}) \nabla \mathbf{u}_h, \nabla \mathbf{v}_h \rangle + \langle \mathbf{b}(\mathbf{x}) \cdot \mathbf{u}_h, \mathbf{v}_h \rangle + \langle \mathbf{r}(\mathbf{u}_h), \mathbf{v}_h \rangle &= \langle \mathbf{f}, \mathbf{v}_h \rangle + \langle \mathbf{g}, \mathbf{v}_h \rangle_{\Gamma_N} \\ &\quad \text{for all } \mathbf{v}_h \in S_{D_h}, \\ \mathbf{u}_h(\mathbf{x}, t) &= \mathbf{u}_{0,h}(\mathbf{x}) \quad \text{on } \Omega \times t = 0, \\ \mathbf{u}_h(\mathbf{x}, t) &= \mathbf{u}_D \quad \text{on } \Gamma_D \times [0, T]. \end{aligned} \tag{4.15}$$

Due to the presence of Dirichlet boundary conditions, a distinction must be made between the number of nodal points N of the discretized domain and the number of nodal unknowns, that is the number of equations N_e of the system. We have in the presence of a Dirichlet condition $N_e < N$. Furthermore, we denote by $N_D \subset N$ the subset of nodes on which the Dirichlet condition is given. The weighting functions $\mathbf{v}_h \in S_{D_h}$ vanish on Γ_D . And the subspace S_h is spanned by the basis functions $\{\varphi_i | i \in N \setminus N_D\}$. Specifically we take continuous functions that are piecewise linear on a quasi-uniform triangulation. As basis functions φ_i , we take the shape functions also known as hat functions. We approximate the solution \mathbf{u}_h using the basis functions

$$\mathbf{u}_h(t, x) = \sum_{i \in N \setminus N_D} \mathbf{u}_i(t) \varphi_i(x) + \sum_{i \in N_D} \mathbf{u}_D \varphi_i(x), \tag{4.16}$$

i.e., each $\mathbf{u}_h \in V_h$ can be written in a unique way as a linear combination of the basis functions φ_i and time dependent coefficients. Moreover, the test functions \mathbf{v}_h are defined such that

$$\mathbf{v}_h \in S_h := \text{span}\{\varphi_i\} \quad \text{where } i \in N \setminus N_D. \quad (4.17)$$

Thus, after substitution of Eq. (4.16) into the semi discretization, and testing with each of the basis functions, we get a system of ordinary differential equations in matrix form

$$\mathbf{M}\dot{\mathbf{u}}_h + \mathbf{A}\mathbf{u}_h + \mathbf{B}\mathbf{u}_h + \mathbf{s}(\mathbf{u}_h) = \mathbf{f}, \quad (4.18)$$

where \mathbf{M} is the mass matrix, \mathbf{A} is the stiffness matrix, \mathbf{B} is the matrix depending on the convection velocity and \mathbf{s} is the vector due to the reaction term. The matrices are defined as follows,

$$\begin{aligned} \mathbf{M} &:= \langle \varphi_i, \varphi_j \rangle, & \mathbf{A} &:= \langle \mathbf{A}(\mathbf{x}) \nabla \varphi_i, \nabla \varphi_j \rangle, \\ \mathbf{B} &:= \langle \mathbf{b}(\mathbf{x}) \cdot \nabla \varphi_i, \varphi_j \rangle, & \mathbf{s}(\mathbf{u}_h) &:= \langle \mathbf{r}(\sum_{i=1}^N \mathbf{u}_i(t) \varphi_i(\mathbf{x})), \varphi_j \rangle. \end{aligned}$$

The nonlinear term

$$\mathbf{s}(\mathbf{u}_h) = \int_{\Omega} \mathbf{r} \left(\sum_{i=1}^N \mathbf{u}_i(t) \varphi_i(\mathbf{x}) \right) \varphi_j \, d\mathbf{x}$$

can be approximated using the quadrature formula. This approximation can be found in Knabner and Angermann [56], also see Heineken [46]

$$\begin{aligned} \int_{\omega} \mathbf{r} \left(\sum_{i=1}^N \mathbf{u}_i(t) \varphi_i(\mathbf{x}) \right) \varphi_j \, d\mathbf{x} &\approx Q_{\Omega}^{T_h} \left(\mathbf{r} \left(\sum_{i=1}^N \mathbf{u}_i(t) \varphi_i(\mathbf{x}) \right) \varphi_j \right), \\ &= \int_{\Omega} I_h \left(\mathbf{r} \left(\sum_{i=1}^N \mathbf{u}_i(t) \varphi_i(\mathbf{x}) \right) \varphi_j \, d\mathbf{x} \right), \\ &= \int_{\Omega} \varphi_i \varphi_j \, d\mathbf{x} \, \mathbf{r}(\mathbf{u}_i) = \mathbf{M}\mathbf{r}(\mathbf{u}_i). \end{aligned} \quad (4.19)$$

Then the system (4.18) can be written as follows

$$\mathbf{M}\dot{\mathbf{u}}_h + \mathbf{A}\mathbf{u}_h + \mathbf{B}\mathbf{u}_h + \mathbf{M}\mathbf{r}(\mathbf{u}_h) = \mathbf{f}.$$

Apply mass lumping, we get the lumped mass matrix, which is a diagonal matrix. It can be invertible easily, then we obtain the system of ordinary differential equation in the following form

$$\dot{\mathbf{u}}_h = \tilde{\mathbf{M}}^{-1} \mathbf{F} - \tilde{\mathbf{M}}^{-1} (\mathbf{A}\mathbf{u}_h - \mathbf{B}\mathbf{u}_h) - \mathbf{r}(\mathbf{u}_h), \quad (4.20)$$

where $\tilde{\mathbf{M}}$ is the lumped mass matrix as explained in Subsection 4.2.1.

4.4 Time discretization

The discretization in time of Eq. (4.18) can be accomplished in several possible ways. To start with, we partition the time interval $[0, T]$ into discrete steps

$$0 = t^0, t^1, \dots, t^n = T,$$

that are not necessarily equidistant. The notation for time step is $\tau^i = t^{i+1} - t^i$ and \mathbf{u}_h^i to be the numerical solution at time t^i . Then we look for an approximation to $\mathbf{u}_h(t)$ at each time level t^i . A classical and widespread practice to achieve a full discretization of Eq. (4.18) is to resort to a discretization of the time derivative by a finite difference scheme. We considered the ODE problem

$$\mathbf{M} \frac{\partial \mathbf{u}_h}{\partial t} = \mathbf{F}(\mathbf{u}_h(t)), \quad \mathbf{u}(t^0) = \mathbf{u}^0, \quad (4.21)$$

where $\mathbf{F}(\mathbf{u}_h(t))$ is a continuous function from $[0, T] \times \mathbb{R}$ into \mathbb{R} which is further Lipschitz continuous with respect to \mathbf{u}_h , uniformly in $t \in [0, T]$.

Stiff initial value problems put special demands on the methods to be used for time discretization. First, for stability reasons one has, in order to avoid excessively small time steps, to use so called *implicit methods*, i.e. methods requiring the solution of a system of equations at each time step. Secondly, one would like to use methods which automatically adapt the size of the time steps according to the smoothness of \mathbf{u}_h and thus automatically take smaller time steps in a transient and larger steps when \mathbf{u}_h becomes smoother. In this section we explain the two classical methods for time discretization, first we start with simple θ scheme and next linearly implicit Runge-Kutta methods.

4.4.1 One step ϑ schemes

The one step θ -scheme, see Hairer and Wanner [45], Quarteroni and Valli [75], applied to the semi-discrete system (4.21) yields

$$\mathbf{M} \frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^n}{\tau^n} = \theta \mathbf{F}(\mathbf{u}_h^{n+1}) + (1 - \theta) \mathbf{F}(\mathbf{u}_h^n). \quad (4.22)$$

The θ scheme includes the following methods: If we take $\theta = 0$, this is just *explicit Euler* method or *forward Euler* method. Other choices that are often considered are $\theta = 1/2$ and $\theta = 1$. The method with $\theta = 1/2$ is called the *Crank Nicolson method*, which is second order in time, and with $\theta = 1$ is called the *implicit Euler* method or *backward Euler* method which is first order accurate in time.

Due to the implicitness, the Crank Nicolson method and backward Euler method are more expensive to use than the forward Euler method. Moreover, for stability reasons one has, in order to avoid excessively small time steps, to use so called *implicit methods*, implicit methods can be *unconditionally stable*. Explicit methods are *conditionally stable*.

4.4.2 Linearly implicit Runge-Kutta methods

Here we consider the system (4.21) are to be solved numerically with the help of linearly implicit one-step methods of Rosenbrock type, see Rosenbrock [79]. If the system is stiff, explicit methods are not efficient due to severe time step restrictions for stability reasons. Implicit Runge-Kutta methods, on the other hand, suffer from a serious practical disadvantage in that the solution of the nonlinear implicit equations occurring at each time step. In this case linearly implicit one step methods are suitable while still maintaining computational efficiency and avoiding the necessity to use a nonlinear solver. If one uses linearly implicit methods only linear systems have to be solved.

For computation an s -stage *Rosenbrock* method of order p with embedding of order $\hat{p} \neq p$ has the form, see Hairer and Wanner [45]

$$\left(\frac{1}{\tau^i \gamma} \mathbf{M} - \mathbf{J}\right) \mathbf{k}_j = \mathbf{F} \left(t^i + \tau^i \alpha_j, \mathbf{u}_h^i + \tau^i \sum_{l=1}^{j-1} a_{jl} \mathbf{k}_l \right) - \mathbf{M} \sum_{l=1}^{j-1} \frac{c_{lj}}{\tau^i} \mathbf{k}_l, \quad \text{where } j = 1, \dots, s, \quad (4.23)$$

$$\mathbf{u}_h^{i+1} = \mathbf{u}_h^i + \sum_{l=1}^s m_l \mathbf{k}_l, \quad (4.24)$$

$$\hat{\mathbf{u}}_h^{i+1} = \mathbf{u}_h^i + \sum_{l=1}^s \hat{m}_l \mathbf{k}_l. \quad (4.25)$$

The method coefficients $\gamma, \alpha_j, a_{jl}, c_{jl}, m_l$, and \hat{m}_l are chosen in such a way that some order conditions are fulfilled to obtain a sufficient consistency order where $\alpha_j = \sum_{i=1}^{j-1} \alpha_{ji}$. The Jacobian is given as $\mathbf{J} = \frac{\partial \mathbf{F}(\mathbf{u}_h(t^i))}{\partial \mathbf{u}_h}$. A derivation of these conditions with Butcher series can be found in Hairer and Wanner [45]. We assume $p > \hat{p}$ which is reasonable since one would prefer to continue the integration with the higher order solution \mathbf{u}_h .

Particularly, in our computations we used the *ROS3P* method which was proposed by Lang and Verwer [58]. This method is A-stable with $R(\infty) \approx 0.73$ and third order accurate, very suitable for nonlinear parabolic problems, see for more details Lang [57]. A second order embedding is used for error estimation which is needed for time step control, see Table 4.1 for the set of coefficients.

We also used in one numerical simulations a certain other class of linearly implicit Runge-Kutta methods, namely the W-methods of Steihaug and Wolfbrandt [86]. They are used as a Krylov-W-method, which is especially well suitable for partitioning methods, see for a systematic study in the thesis of Heineken [46]. This method has order $p = 2$, the embedded method is of order $\hat{p} = 1$, see Table 4.2 for set of coefficients. In this case we considered the autonomous problem (4.20) in the simulation.

$\gamma = 7.886751345948129e - 01$	
$a_{21} = 1.267949192431123e + 00$	$c_{21} = 1.607695154586736e + 00$
$a_{31} = 1.267949192431123e + 00$	$c_{31} = 3.464101615137755e + 00$
$a_{32} = 0.0000000000000000e + 00$	$c_{32} = 1.732050807568877e + 00$
$\alpha_1 = 0.0000000000000000e + 00$	$\gamma_1 = 7.886751345948129e - 01$
$\alpha_2 = 1.0000000000000000e + 00$	$\gamma_2 = -2.11324865405187e - 01$
$\alpha_3 = 1.0000000000000000e + 00$	$\gamma_3 = -1.077350269189626e + 00$
$m_1 = 2.0000000000000000e + 00$	$\hat{m}_1 = 2.113248654051871e + 00$
$m_2 = 5.773502691896258e - 01$	$\hat{m}_2 = 1.0000000000000000e + 00$
$m_3 = 4.226497308103742e - 01$	$\hat{m}_3 = 4.226497308103742e - 01$

Table 4.1: Set of coefficients for ROS3P method

$\gamma = 1.0 - \frac{1}{2}\sqrt{2}$	
$a_{21} = 1.0$	$c_{21} = 2.0 + \sqrt{2}$
$a_{31} = 1.0$	$c_{31} = 1.0$
$a_{32} = 0.0$	$c_{32} = 1.0 - \sqrt{2}$
$\alpha_1 = 0.0$	
$\alpha_2 = 1.0$	
$\alpha_3 = 1.0$	
$m_1 = 1.0$	$\hat{m}_1 = \frac{9}{10} - \frac{1}{20}\sqrt{2}$
$m_2 = \frac{1}{2} - \frac{1}{2}\sqrt{2}$	$\hat{m}_2 = \frac{9}{20} - \frac{11}{20}\sqrt{2}$
$m_3 = \frac{1}{2}$	$\hat{m}_3 = \frac{11}{20} + \frac{1}{20}\sqrt{2}$

Table 4.2: Set of coefficients for W-method

In the numerical integration of ordinary differential equations, automatic stepsize control is the most important means to make an integration method efficient. A standard rule

for selecting the stepsize is, e.g., see Hairer et al. [44], Shampine [84]

$$\bar{\tau} := \beta \left(\frac{TOL_t}{\epsilon^{i+1}} \right)^{1/p} \tau^i, \quad \tau_{\text{new}} = \begin{cases} \beta_{\max} \tau^i, & \bar{\tau} > \beta_{\max} \tau^i, \\ \beta_{\min} \tau^i, & \bar{\tau} < \beta_{\min} \tau^i, \\ \bar{\tau}, & \text{otherwise.} \end{cases}$$

where TOL_t is the tolerance to set for time step control, β is a safety factor chosen ≤ 1 and p is related to the order of the integration method. The factors β_{\min} and β_{\max} restrict time step jumps. The time step is rejected if $\epsilon^{i+1} > \zeta \cdot TOL_t$, and new attempt is made with a smaller stepsize. A typical value of ζ is 1.2. The stepsize oscillates violently and much computation time is spent by recalculating the rejected steps and changing the stepsize. This is especially true in the case of stiff differential equations, see for more details Gustafsson et al. [42], Hairer and Wanner [45]. We observed this also in our computations.

A more sophisticated stepsize control proposed by Gustafsson et al. [42], is suitable for stiff differential equations. A new time step τ_{new} , see Gustafsson et al. [42], Gustafsson [41] is computed by

$$\bar{\tau} := \beta \frac{\tau^i}{\tau^{i-1}} \left(\frac{TOL_t}{\epsilon^{i+1}} \right)^{\frac{p_2}{p}} \left(\frac{\epsilon^i}{\epsilon^{i+1}} \right)^{\frac{p_1}{p}} \tau^i, \quad \tau_{\text{new}} = \begin{cases} \beta_{\max} \tau^i, & \bar{\tau} > \beta_{\max} \tau^i, \\ \beta_{\min} \tau^i, & \bar{\tau} < \beta_{\min} \tau^i, \\ \bar{\tau}, & \text{otherwise.} \end{cases} \quad (4.26)$$

The parameter $\beta \in]0, 1]$ is safety factor. In computations we have chosen the parameters $p_1 = 1$ and $p_2 = 1$. The value for TOL_t varied in the numerical computation, see Chapter 6.

It is well suited for our numerical calculations. More details can be found in Chapter 6.

4.5 Grid adaptivity and error estimators

In the numerical solution of reaction-diffusion problems of science or engineering, one often encounters the difficulty that the overall accuracy of the numerical approximation is deteriorated by local singularities such as, e.g., singularities arising from re-entrant corners, interior or boundary layers, or sharp shock-like fronts. An obvious remedy is to refine the mesh near the critical regions, i.e., to place more grid-points where the solution is less regular. Now the question arises, how to identify those regions and how to obtain a good balance between the refined and un-refined regions such that the overall accuracy is optimal. Adaptive procedures try to automatically refine, coarsen, or relocate a mesh and/or adjust the basis to achieve a solution having a specified accuracy in an optimal

fashion. The computation typically begins with a trial solution generated on a coarse mesh with a low order basis. The error of this solution is estimated, if it fails to satisfy the specified accuracy, adjustments are made with the goal of obtaining the desired solution with minimal effort. Common procedures studied for adaptive finite element methods are

- local refinement and/or coarsening of a mesh (h-refinement),
- relocating or moving a mesh (r-refinement),
- locally varying the polynomial degree of the basis (p-refinement).

A *posteriori* error estimates provide accuracy appraisals that are necessary to terminate an adaptive procedure. For stationary problems the estimators can roughly be classified as follows:

- *Residual estimates*: Estimate the error of the computed numerical solution by a suitable norm of its residual with respect to the strong form of the differential equation. These estimators are due to Babuška and Rheinboldt [3] in one dimension and also see for higher dimensions Eriksson and Johnson [33], Verfürth [93] for more details.
- *Solution of local problems*: Solve locally discrete problems similar to, but simpler than, the original problem and the use of appropriate norms of the solutions for the error estimation, see Bank and Weiser [6], Verfürth [93] for more details.
- *Hierarchical basis error estimates*: Evaluate the residual of the computed finite element solution with respect to another finite element space corresponding to higher order elements or to a refined grid, see Bank and Smith [5], Bornemann et al. [22].
- *Averaging methods*: Use some local averaging or post processing technique for error estimation, see Zienkiewicz-Zhu [98], Rodriguez [77].

In our simulations we considered mesh refinement and coarsening based on the Z^2 error indicator of Zienkiewicz and Zhu [98]. The computation typically begins with a trial solution generated on a coarse mesh and successively refine the mesh up to some levels based on the Z^2 error indicator. The full spatial and temporal discretization leads to an approximate solution \mathbf{u}_h with $\mathbf{u}_h(\cdot, t_i) \in V_h$ at the discrete time points t_i , $i = 0, \dots, M$ where the time integration scheme is evaluated. Here we will recall the Z^2 error indicator.

4.5.1 The Z^2 error indicator

We denote by W_h the space of all piecewise linear vector-fields on a triangulation \mathcal{T}_h and set $V_h := W_h \cap C(\Omega, \mathbb{R}^2)$. Define a mesh-dependent scalar product $\langle \cdot, \cdot \rangle : W_h \times W_h \rightarrow \mathbb{R}$ by

$$\langle v, w \rangle_h := \sum_{T \in \mathcal{T}} \frac{|T|}{3} \left\{ \sum_{i=1}^3 v|_T(\mathbf{a}_i) w|_T(\mathbf{a}_i) \right\}. \quad (4.27)$$

Here, $|T|$ denotes the area of T . The quadrature formula

$$\int_T v \, d\mathbf{x} \sim \frac{|T|}{3} \sum_{i=1}^3 v(\mathbf{a}_i)$$

is exact for all linear functions. We have

$$\langle v, w \rangle_h = \int_{\Omega} v \cdot w \, d\mathbf{x}, \quad (4.28)$$

if both arguments are elements of W_h and at least one of them is piecewise constant. Denote by \mathbf{u} and \mathbf{u}_h the unique solution of problems (4.13) and (4.15). Suppose that we dispose of an easily computable approximation $G\mathbf{u}_h$ of $\nabla\mathbf{u}_h$ such that

$$\|\nabla\mathbf{u} - G\mathbf{u}_h\|_{L^2(T)} \leq \alpha \|\nabla\mathbf{u} - \nabla\mathbf{u}_h\|_{L^2(T)}, \quad (4.29)$$

holds with a constant $0 \leq \alpha \leq 1$. We then have

$$\begin{aligned} \frac{1}{1+\alpha} \|G\mathbf{u}_h - \nabla\mathbf{u}_h\|_{L^2(T)} &\leq \|\nabla\mathbf{u} - \nabla\mathbf{u}_h\|_{L^2(T)} \\ &\leq \frac{1}{1-\alpha} \|G\mathbf{u}_h - \nabla\mathbf{u}_h\|_{L^2(T)}, \end{aligned} \quad (4.30)$$

and we may therefore choose $\|G\mathbf{u}_h - \nabla\mathbf{u}_h\|_{L^2(T)}$ as an error estimator. Since $\nabla\mathbf{u}_h$ is a piecewise constant vector-field we may hope that the L^2 -projection onto the continuous, piecewise linear vector-fields satisfies inequality (4.29). The computation of this projection, however, is as expensive as the solution of problem Eq. (4.13). We therefore replace the L^2 -scalar product by an approximation which leads to a more tractable auxiliary problem.

Let $G\mathbf{u}_h \in V_h$ be the $\langle \cdot, \cdot \rangle_h$ -projection of $\nabla\mathbf{u}_h$ onto V_h , i.e.

$$\langle G\mathbf{u}_h, \mathbf{v}_h \rangle_h = \langle \nabla\mathbf{u}_h, \mathbf{v}_h \rangle_h \quad \forall \mathbf{v}_h \in V_h. \quad (4.31)$$

Eqs. (4.27) and (4.28) imply that, see Verfürth [93]

$$G\mathbf{u}_h(\mathbf{a}_i) = \sum_{T \subset \mathbf{D}_a} \frac{|T|}{|\mathbf{D}_a|} \nabla\mathbf{u}_h|_T(\mathbf{a}_i). \quad (4.32)$$

Thus, $G\mathbf{u}_h$ may be computed by a local averaging of $\nabla\mathbf{u}_h$. We finally set

$$\eta_{Z,T} := \|G\mathbf{u}_h - \nabla\mathbf{u}_h\|_{L^2(T)}, \quad (4.33)$$

and

$$\eta_Z := \left\{ \sum_{T \in \mathcal{T}_h} \eta_{Z,T}^2 \right\}^{1/2}. \quad (4.34)$$

The Z^2 indicator $\eta_{Z,T}$ is an estimate for $\|\nabla\mathbf{u}_h^t(\cdot, t_i) - \nabla\mathbf{u}_{i-1}^t(\cdot, t_i)\|_{L^2(T)}$, see Verfürth [93]. Let $\lambda(T) \in \mathbb{N}_0$ be the refinement level of triangle $T \in \mathcal{T}_h$, $\lambda_{max} \in \mathbb{N}_0$ be a given maximum refinement level, and $\phi_1, \dots, \phi_{\lambda_{max}}$ be given real numbers satisfying $0 \leq \phi_1 \dots \leq \phi_{\lambda_{max}}$. With the choice of $\phi_1, \dots, \phi_{\lambda_{max}}$ one controls the structure of the grid. If we set $\phi_1 = \dots = \phi_{\lambda_{max}} = 0$ this leads to a uniform triangulation of level λ_{max} . Here we used the scaled indicator for any triangle $T \in \mathcal{T}_h$

$$\phi_T := \eta_{Z,T} / \sqrt{T}. \quad (4.35)$$

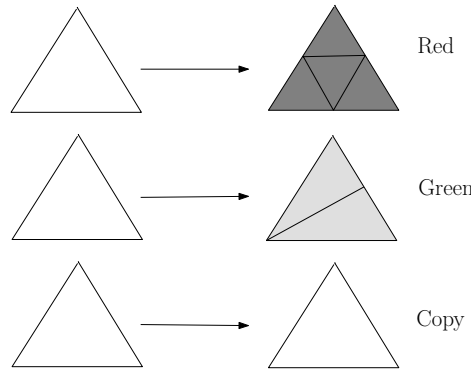


Figure 4.2: Red-Green-Copy refinement in 2D.

Then these quantities are used to judge the quality of the underlying discretization in the element T . In a next step a set T_i of elements which have to be refined are selected. Then a refinement rule can be applied to each element resulting in the generation of new elements on the next finer level. Each refinement rule is either of type *regular* or *red*, *irregular* and *copy*, see Figure 4.2. Here the refinement algorithm is responsible for generating an admissible mesh on each level, i.e. the intersection of two different elements is either empty, a node or an edge. In our simulations we used only regular and copy refinement rules. After choosing a T_i elements we are led to a robust and stable refinement strategy.

For the initial grid and successive refinements we used the programm package UG [12]. The refinement rules have been extended to 3D by Stefan Lang in the context of UG, see [59].

We start with a uniform initial triangulation of refinement level 0. Here the triangle T is marked for

1. refinement if $\phi_T > \phi_{\lambda(T)}$ and $\lambda(T) < i$ for $i = 0, \dots, \lambda_{max}$,
2. coarsening if $\phi_T < \phi_{\lambda(T)}$ and $\lambda(T) > i$ for $i = 0, \dots, \lambda_{max}$,

where ϕ_T is calculated according to Eq. (4.35).

If we consider the system (3.1)-(3.4) of four reaction-diffusion equations with approximate solutions \mathbf{u}_h , we compute ϕ_T according to Eq. (4.35) for each component separately, resulting in $\phi_{T,1}$, $\phi_{T,2}$, $\phi_{T,3}$ and $\phi_{T,4}$ for the four components respectively. Then we set

$$\phi_T := (\phi_{T,1} + \phi_{T,2} + \phi_{T,3} + \phi_{T,4})/4,$$

and we mark the elements as indicated above. We refine the mesh locally until a minimum of 4 grid points lie in the area of each channel in a cluster. This level we considered as the coarse level for our simulation. Here we have considered different cases according to the number of clusters. During the time step, the grid adaption process will be explained in Chapter 6.

Test Case 1: In this case we considered one cluster with 20 channels and the domain size is $[0,33000 \text{ nm}] \times [0,33000 \text{ nm}]$. For the initial triangulation a diameter of 700 nm for the triangle is considered. The adaption parameters are $\lambda_{max} = 6$ and $\phi_1 = 1 \cdot 10^{-5}$, $\phi_2 = 2 \cdot 10^{-5}$, $\phi_3 = 4 \cdot 10^{-5}$, $\phi_4 = 8 \cdot 10^{-5}$, $\phi_5 = 16 \cdot 10^{-5}$, $\phi_6 = 32 \cdot 10^{-5}$. Using these adaption parameters, we apply Z^2 -indicator and refine the mesh. In first level we identify the location of clusters. Then successively refine the mesh until each channel gets the minimum 4 grid points to lie in the area of each channel in a cluster. Then we stop the refine process and we fix the mesh for simulation. Initial mesh, first level mesh and the final mesh for this test case can be seen in Figure 4.3.

Test Case 2: In this case we considered 16 clusters with distance of 10 μm and each cluster consists of 20 channels. The domain size is $[0,24000 \text{ nm}] \times [0,24000 \text{ nm}]$. For the

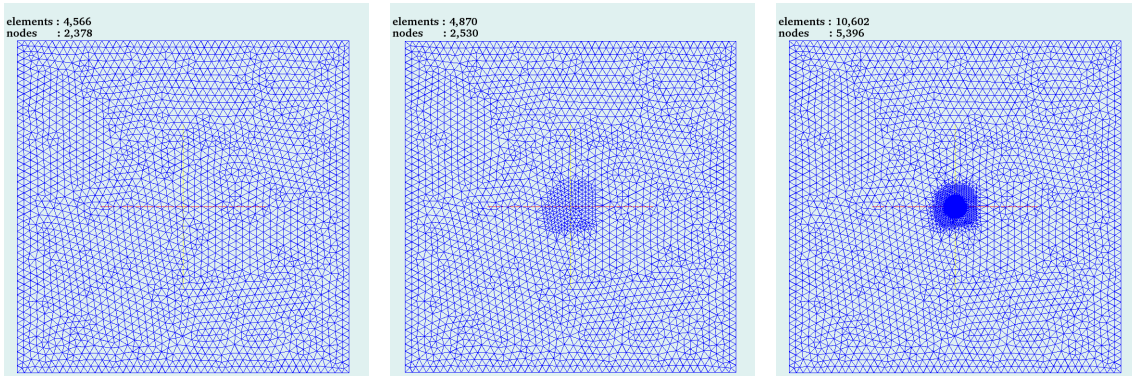


Figure 4.3: Grid refinement at initial level, level 1 and level 8 for **Test Case 1**.

initial triangulation a diameter of 700 nm for the triangle is considered. The adaption parameters are $\lambda_{max} = 6$ and $\phi_1 = 1 \cdot 10^{-5}$, $\phi_2 = 2 \cdot 10^{-5}$, $\phi_3 = 4 \cdot 10^{-5}$, $\phi_4 = 8 \cdot 10^{-5}$, $\phi_5 = 16 \cdot 10^{-5}$, $\phi_6 = 32 \cdot 10^{-5}$. Here also we refine the mesh as explained in Test case 1. Initial mesh, first level mesh and the final mesh for this test case can be seen in Figure 4.4.

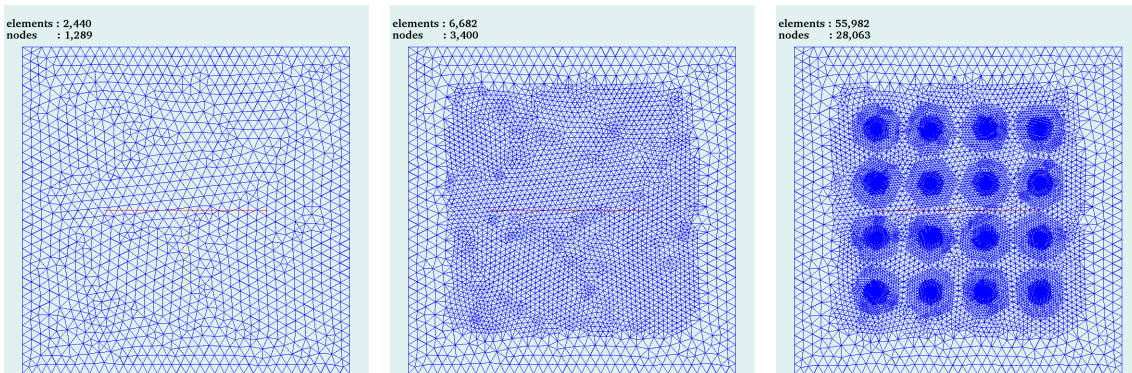


Figure 4.4: Grid refinement at initial level, level 1 and level 6 for **Test Case 2**.

Test Case 3: In this case we considered 100 clusters with a distance of $4 \mu\text{m}$ and each cluster consists of 20 channels. The domain size is $[0,48000 \text{ nm}] \times [0,48000 \text{ nm}]$. For the initial triangulation a diameter of 700 nm for the triangle is considered. The adaption parameters are $\lambda_{max} = 6$ and $\phi_1 = 1 \cdot 10^{-5}$, $\phi_2 = 2 \cdot 10^{-5}$, $\phi_3 = 4 \cdot 10^{-5}$, $\phi_4 = 8 \cdot 10^{-5}$, $\phi_5 = 16 \cdot 10^{-5}$, $\phi_6 = 32 \cdot 10^{-5}$. In this case also we proceed like in Test Case 1. Initial mesh, first level mesh, the final mesh for this test case can be seen in Figure 4.5.

Test Case 4: In this case we considered 1 cluster consists of 1 open channel. The

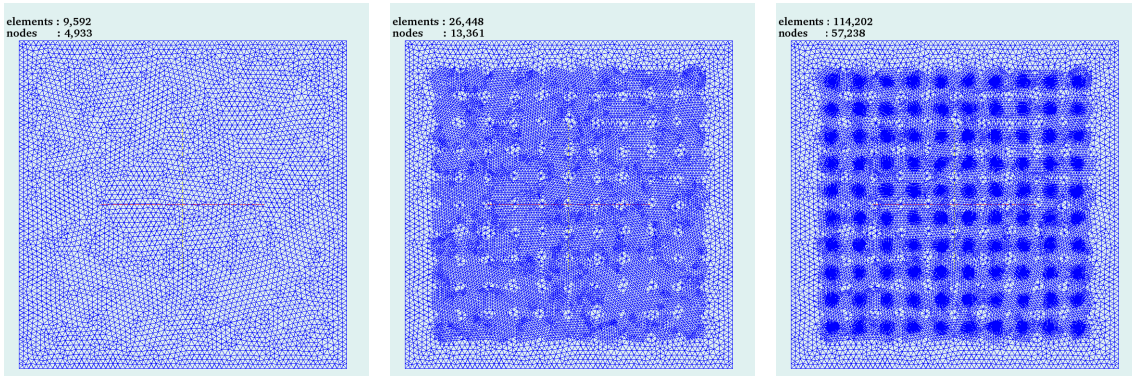


Figure 4.5: Grid refinement at initial level, level 1 and level 6 for **Test Case 3**.

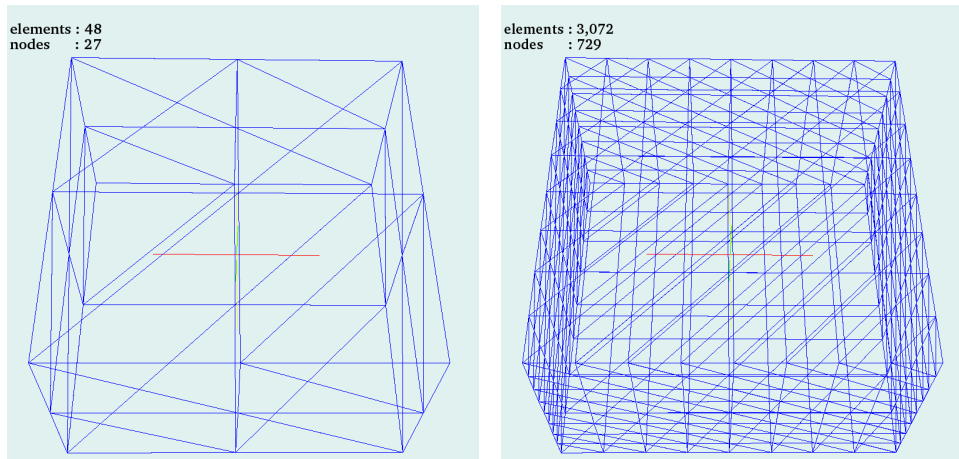


Figure 4.6: Grid refinement at initial grid and level 1 for 3D case.

domain size is $[0,8000 \text{ nm}] \times [0,8000 \text{ nm}] \times [0,5000 \text{ nm}]$. The adaption parameters are $\lambda_{max} = 12$ and $\phi_i = 9 \cdot 10^{-6}$, $i = 1, \dots, 12$. In this case also we proceed like in Test Case 1. Initial mesh, first level mesh, can be seen in Figure 4.6 and the level 9 mesh and level 12 mesh can be seen in Figure 4.7.

4.6 Solution of algebraic equations

In this section we concentrate on the solution of the algebraic equations arising within each time step of the fully implicit/fully coupled solution procedure. The nonlinear and linear solvers to be described in this section utilize a multigrid mesh structure to accelerate the solution process. This multigrid mesh structure is denoted by E_0, \dots, E_J . It is constructed

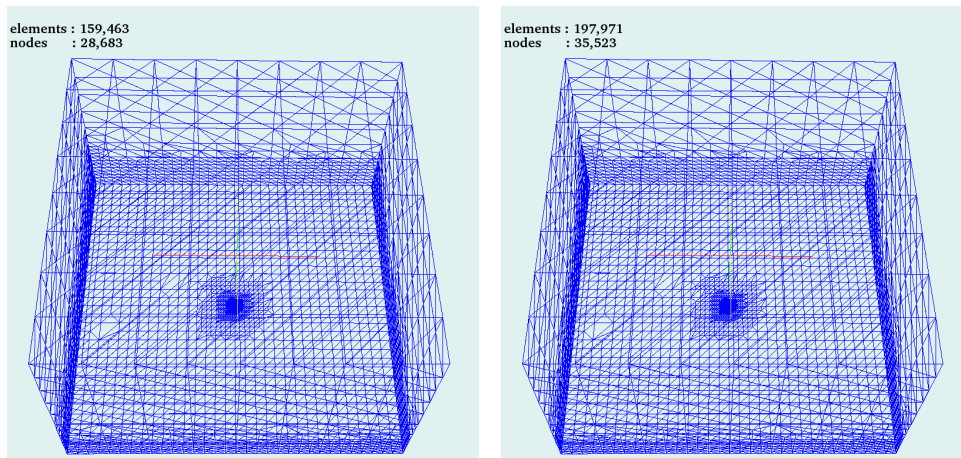


Figure 4.7: Grid refinement at level 9 and level 12 for 3D case.

from an initial mesh, which is generated by hand or an initial grid generator, by *regular* subdivision of each element. Initial mesh and finer meshes are generated in our calculations using the programming package UG (Unstructured Grid) [12]. For refinement rules in sequential and parallel cases we refer to Bank et al. [4], Bastian [8] and Lang [59].

4.6.1 Inexact Newton method

Each time step of the fully implicit scheme leads to a large set of nonlinear algebraic equations

$$\mathbf{G}(\mathbf{u}_h) = \mathbf{0},$$

to be solved. The vector \mathbf{u}_h contains the unknowns. Actually, those coefficients in \mathbf{u}_h corresponding to Dirichlet boundary conditions are not unknown and the number of nonlinear equations are reduced correspondingly, see for implementation issues in the programm package UG [12]. The linearization of the Jacobian J of \mathbf{G} at the linearization point \mathbf{u}_h , an iteration point of the method, is the matrix with entries that can be computed either analytically or by numerical differentiation

$$J(\mathbf{u}_h)_{ij} := \frac{\partial \mathbf{G}_i}{\partial \mathbf{u}_{hj}}(\mathbf{u}_h).$$

In algorithm superscript κ denotes the iteration number and let $\|\cdot\|_2$ be the Euclidean vector norm. Here is the inexact Newton algorithm, see Bastian [11], Braess [23].

- 1: input $\leftarrow \mathbf{G}, \mathbf{u}_h, TOL_{nl}$
- 2: $\kappa := 0; \mathbf{u}_h^0 := \mathbf{u}_h;$
- 3: **while** ($\|\mathbf{G}(\mathbf{u}_h^\kappa)\|_2 \geq TOL_{nl} \|\mathbf{G}(\mathbf{u}_h^0)\|_2$) **do**
- 4: choose $TOL_{lin}^\kappa \in]0, 1];$
- 5: find \mathbf{s}^κ such that
- 6: $\|\mathbf{G}(\mathbf{u}_h^\kappa) + J(\mathbf{u}_h^\kappa)\mathbf{s}^\kappa\|_2 \leq TOL_{lin}^\kappa \|\mathbf{G}(\mathbf{u}_h^\kappa)\|_2;$
- 7: choose $\lambda^\kappa \in]0, 1];$
- 8: $\mathbf{u}_h^{\kappa+1} := \mathbf{u}_h^\kappa + \lambda^\kappa \mathbf{s}^\kappa;$
- 9: $\kappa := \kappa + 1;$
- 10: **end while**

Algorithm 1: The above algorithm solves the nonlinear system $\mathbf{G}(\mathbf{u}_h) = \mathbf{0}$ to accuracy TOL_{nl} starting from the initial guess \mathbf{u}_h .

Since the Newton method converges only in a sufficiently close neighborhood of the solution, a damping strategy is needed to achieve global convergence. Step (7) in Algorithm 1 implements a simple line search strategy where the damping factor λ^κ is chosen as the largest value in the set $\{1, \frac{1}{2}, \frac{1}{4}, \dots\}$ such that

$$\|\mathbf{G}(\mathbf{u}_h^\kappa + \lambda^\kappa \mathbf{s}^\kappa)\|_2 \leq \left(1 - \frac{1}{4}\lambda^\kappa\right) \|\mathbf{G}(\mathbf{u}_h^\kappa)\|_2. \quad (4.36)$$

More details of this strategy one can find in to Braess [23].

4.6.2 Solution of linear equations

Next we treat the resolution of large and sparse systems of linear equations

$$A\mathbf{u}_h = \mathbf{b}. \quad (4.37)$$

This linear system arises from the linearization of the non linear system using Newton's method or in each stage of a linearly implicit time stepping method. For large number of unknowns iterative methods are appropriate choice. Starting with an initial guess \mathbf{u}_h^0 , iterative methods for the resolution of system (4.37) produce a sequence of iterates $\mathbf{u}_h^1, \mathbf{u}_h^2, \dots$ that converges to the exact solution \mathbf{u}_h . We used *Krylov subspace methods* in our numerical simulations. A good description of these algorithms is given in Barrett et al. [7]. The methods can be accelerated substantially by using a preconditioner, which is a basic iterative method or the multigrid method. For unsymmetric matrices A the minimization over

the Krylov subspace cannot be done as cheaply as in the symmetric case. Several methods are known, each sacrificing another property, see for more details Barrett et al. [7].

Here we give a short description about the standard multigrid algorithm. The multigrid mesh structure denoted by

$$E_0, E_1, \dots, E_L, \quad (4.38)$$

is constructed from an intentionally coarse mesh E_0 by *regular* subdivision of each element, where L is the maximal level. The set of vertices belonging to mesh E_l is written as V_l . The number of elements on level l is denoted by K_l and the number of vertices by N_l . The discretized equations on each mesh level are then given by

$$A_l \mathbf{u}_{hl} = \mathbf{b}_l, \quad l = 0, \dots, L. \quad (4.39)$$

The dimension of these systems is N_l . Furthermore we need *grid transfer operators* $\mathbf{R}_l, \mathbf{P}_l$ which are linear mappings of appropriate dimensions

$$\mathbf{R}_l : \mathbb{R}^{N_l} \rightarrow \mathbb{R}^{N_{l-1}}, \quad (\text{Restriction operator})$$

$$\mathbf{P}_l : \mathbb{R}^{N_{l-1}} \rightarrow \mathbb{R}^{N_l}. \quad (\text{Prolongation operator})$$

Here we give a basic multigrid algorithm, for the ideas behind this method see Hackbusch [43], Bastian [9].

if ($l = 0$) **then**

$$\mathbf{u}_{h_0} = A_0^{-1} \mathbf{b}_0;$$

else

Apply ν_1 iterations of \mathcal{S} to $A_l \mathbf{u}_{hl} = \mathbf{b}_l$;

$$\mathbf{d}_l = \mathbf{b}_l - A_l \mathbf{u}_{hl};$$

$$\mathbf{d}_{l-1} = \mathbf{R}_l \mathbf{d}_l;$$

$$\mathbf{s}_{l-1} = 0;$$

for ($g = 1, \dots, \gamma$) **do**

$$\text{mgc}(l-1, \mathbf{s}_{l-1}, \mathbf{d}_{l-1});$$

end for

$$\mathbf{s}_l = \mathbf{P}_l \mathbf{s}_{l-1};$$

$$\mathbf{u}_{hl} = \mathbf{u}_{hl} + \mathbf{s}_l;$$

Apply ν_2 iterations of \mathcal{S} to $A_l \mathbf{u}_{hl} = \mathbf{b}_l$;

end if

Algorithm 2: Standard multigrid method with finest level l applied to the current iterate \mathbf{u}_{hl} .

The parameters ν_1, ν_2 are the number of pre and post smoothing steps. Typically they are in the range 1, 2, 3. The parameter γ controls the cycle form. In our simulations we used only $\gamma = 1$, called a *V-cycle*.

The canonical way to define the prolongation operator (matrix) P_l is via finite element interpolation

$$(\mathbf{P}_l \mathbf{s}_{l-1})_i = \sum_{j=1}^{N_{l-1}} \mathbf{s}_{l-1,j} \varphi_{l-1,j}(\mathbf{x}_i), \quad (4.40)$$

where $\varphi_{l-1,j}$ is the finite element basis function corresponding to vertex j on level $l - 1$. Since the support of the basis functions is local the matrix \mathbf{P}_l is a very sparse rectangular matrix. The standard choice for the restriction operator \mathbf{R}_l is

$$\mathbf{R}_l = \mathbf{P}_l^T,$$

in the case of a finite element discretization. Good description can be found in Hackbusch [43], Wesseling [94], Shaidurov [83], Briggs [24], Bastian [11].

4.7 Domain decomposition methods

Domain decomposition methods allow for the effective implementation of numerical techniques for partial differential equations on parallel architectures. Any such method is based on the assumption that the given computational domain, say Ω , is partitioned into subdomains $\Omega_i, i = 1, \dots, \mathbf{p}$, that may or may not overlap. Next, the original problem can be reformulated upon each subdomain Ω_i yielding a family of subproblems of reduced size which are to coupled each others through the values of the unknown solution at subdomain interfaces. The interface coupling can be relaxed at the expense of introducing an iterative process among subdomains, yielding at each step independent sub problems upon subdomains. This domain decomposition may enter at the continuous level, where different physical models may be used in different regions. Or it may be introduced at the discretization level, where it may be convenient to employ different approximation methods in different regions. Or it may be introduced in the solution of the algebraic systems arising from the approximation of the partial differential equation. The parallel programming model in UG [12] is based on the third aspect of the domain decomposition approach.

The parallelization of all components of the adaptive multigrid method is based on a distribution of the data onto the set of processors. Here the multigrid covering the domain is split up into non overlapping regions. Overlapping storage of objects on processor

boundaries is used for an efficient implementation, see for more details Bastian [10, 9] and Bastian et al. [12]. Here the appropriate data partitioning is determined, this is called *load balancing*, by the help of the CHACO package [50]. The *load migration*, which is able to move the multigrid objects to the processors, is realized based on the parallel programming mode *DDD* (Dynamic Distributed Data), see for more details Birken and Bastian [20].

In our numerical simulations we used Recursive Coordinate Bisection (RCB) and Recursive Inertial Bisection (RIB) algorithms for domain decomposition. The RCB is a divide and conquer scheme. In each step of RCB the dual graph vertices are sorted according to their coordinate values. Then the vertices are bisected into two equal sets using the mean value. The sets are then further divided by the recursive application of the same splitting algorithm until the number of sets equals the number of processors. RIB is similar to RCB in that it divides the domain based on the location of the objects being partitioned by use of cutting planes. In RIB, the dual graph is first divided into two equal sets by a cutting plane orthogonal to the longest direction of the domain so that half the work load is in each of the sets. Again the sets are then further divided by recursive application of the same splitting algorithm until the number of sets equals the number of processors. This algorithm was first devised to cut into a number of sets which is a power of two, see for more details in CHACO users guide [50].

The most important part is the parallel grid manager. In the framework of UG [12] this is complemented by parallel linear algebra which allows to represent numerical algorithms easily on the distributed data with arbitrary load balancing of the grid on every level. A load balancing corresponds to a non overlapping domain decomposition

$$\bar{\Omega} = \bar{\Omega}_l^1 \cup \dots \cup \bar{\Omega}_l^P,$$

which may be independent on every grid level l . The overlapping decomposition

$$\mathcal{P}_l = \mathcal{P}_l^1 \cup \dots \cup \mathcal{P}_l^P \subset \mathbb{R}^d,$$

of the interpolation points on every grid level l such that $\mathcal{P}_l^i \subset \mathcal{P}_l \cap \bar{\Omega}_l^i$.

For a vector we distinguish between two different represented modes on the overlapping set of nodal points:

- consistent** The solution vector $\mathbf{u}_l \in \mathbb{R}^{N_l}$ and the correction $\mathbf{c}_l \in \mathbb{R}^{N_l}$ are represented consistently, i.e. $\mathbf{u}_l[P] = \mathbf{u}_l^i[P]$ for $P \in \mathcal{P}_l$.
- Inconsistent** the vector $\mathbf{y}_l \in \mathbb{R}^{N_l}$ represents inconsistently, i.e. $\mathbf{y}_l^i = \sum_{P \in \mathcal{P}_l} \mathbf{y}_l^i[P]$.

A matrix will be denoted consistent, if it transforms every consistent vector into a consistent vector. Every operation has to be checked for consistency. For more detail explanation on the parallel linear algebra one can refer to Wieners [95], Bastian [11]. Here we will give short overview of parallel iteration step, here we denote the matrix $A_l \in \mathbb{R}^{N_l \times N_l}$, the right hand side vector $\mathbf{b}_l \in \mathbb{R}^{N_l}$ and the defect $\mathbf{d}_l \in \mathbb{R}^{N_l}$.

- | | | |
|--------------------------------|--|-------------------------|
| compute the defect | $\mathbf{d}_l^i = \mathbf{b}_l^i - A_l^i \mathbf{u}_l^i$ | (without communication) |
| compute the local correction | $\mathbf{s}_l^i = S_l^i \mathbf{d}_l^i$ | (without communication) |
| make the correction consistent | $\mathbf{c}_l^i[P] = \sum_{P \in \mathcal{P}_l} \mathbf{s}_l^i[P]$ | (with communication) |
| add the correction | $\mathbf{u}_l^i := \mathbf{u}_l^i + \mathbf{c}_l^i$ | (without communication) |

For the formulation and efficient implementation of multigrid methods on locally refined grids refer to Bastian [9, 10, 8], Bastian et al. [12] and Lang et al. [60].

Chapter 5

Numerical Results of Heat and Mass Transfer in Fluidized Beds

In this chapter we present the simulation results for heat and mass transfer in fluidized beds which is explained in Chapter 2. In the first section we give the simulation results with a uniform liquid distribution. Here we study the numerical behavior of heat and mass transfer with different liquid film thicknesses, influence of the ratio of heat transfer coefficient, influence of the air inlet temperature, influence of the air mass flow rate, influence of the liquid mass flow rate and influence of the diameter of particles. In next subsection we present the two dimensional numerical results with a number of nozzle net depositions. At the end we present the numerically convergent results with semi implicit and implicit method as well as using different linear solvers. In the second section we present the numerical results in three space dimensions and comparison to experimental results. In the final section we give the numerical results based on domain decomposition methods.

5.1 Numerical results in 1D and 2D

5.1.1 One-dimensional simulation results with uniform liquid distribution

For the unsteady one-dimensional problem, computational results of the balance quantities air humidity, air temperature, degree of wetting, liquid film temperature and particle temperature, assuming uniform liquid distribution, are presented. The sprayed liquid is distributed over the domain, with a spraying rate $\dot{m}_L = 5.9 \text{ kg/h}$. Here we consider that the sprayed liquid to be water. For simulations, we assume that the wall temperature is constant. All parameters used for the simulation are listed in Table A.1. We can

observe from Figure 5.1 that the air humidity increases almost linearly in the axial direction with the distance from the gas distributor. Thereby the steady-state is reached after 200 seconds. When the liquid distribution starts over the domain, the air humidity increases suddenly at the outlet, and then increases slowly until it reaches the stationary solution. Air temperature decreases from the gas distributor plate to the top of the fluidized bed, see Figure 5.1.

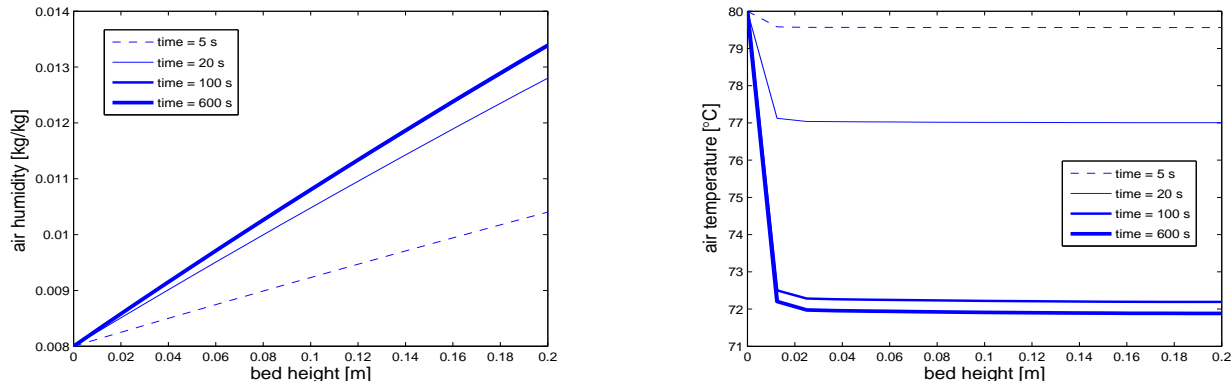


Figure 5.1: Simulations of air humidity and air temperature for different time levels depending on the bed height.

The air temperature is at its maximum at the bottom surface because of the inflow of the air. After reaching the stationary solution, the simulated outlet air temperature coincides with the outlet air temperature determined from the total balance, see Eq. (2.16). The average values are determined using the L^1 norm, which is defined as follows,

$$a_{avg} = (|a_1| + \dots + |a_n|)/\text{length}(a).$$

The particle temperature decreases slower than the air temperature, because of the particle heat capacity, and during stationary operation it lies somewhat under the air temperature. The particle temperature is locus dependent, due to the high air-particle and air-liquid film heat transfers. The difference between the maximum and minimum values of the particle temperature is very small, see Figure 5.4. At the initial time $t = 0$, the average degree of wetting is 10^{-8} . When the simulation starts, it increases sharply at the outlet and thereafter increases steadily towards the final value, which we can observe in Figure 5.3.

The instantaneous mass transfer conditions during the start-up causes also an increase of the degree of wetting. Despite the constant evaporation flow, this temporal change is triggered by the change of the liquid film temperature, which at first jumps from the

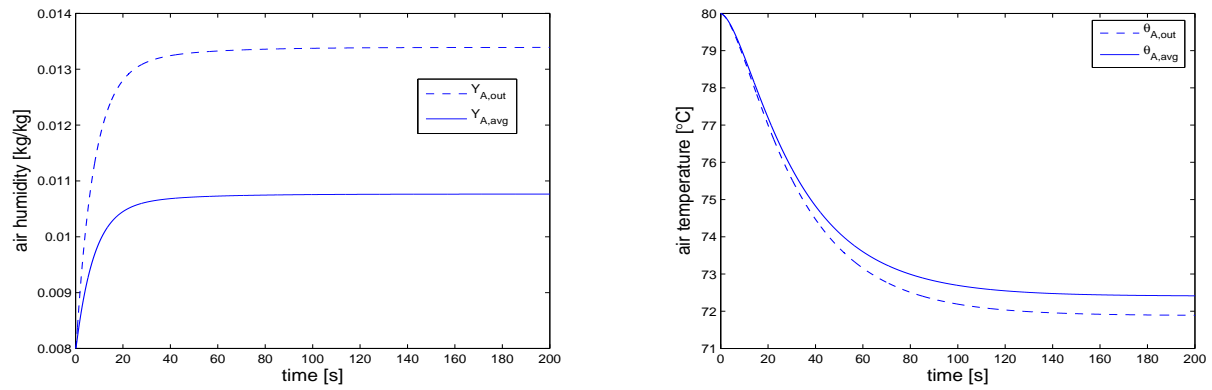


Figure 5.2: Simulations of outlet and average values of air humidity and air temperature.

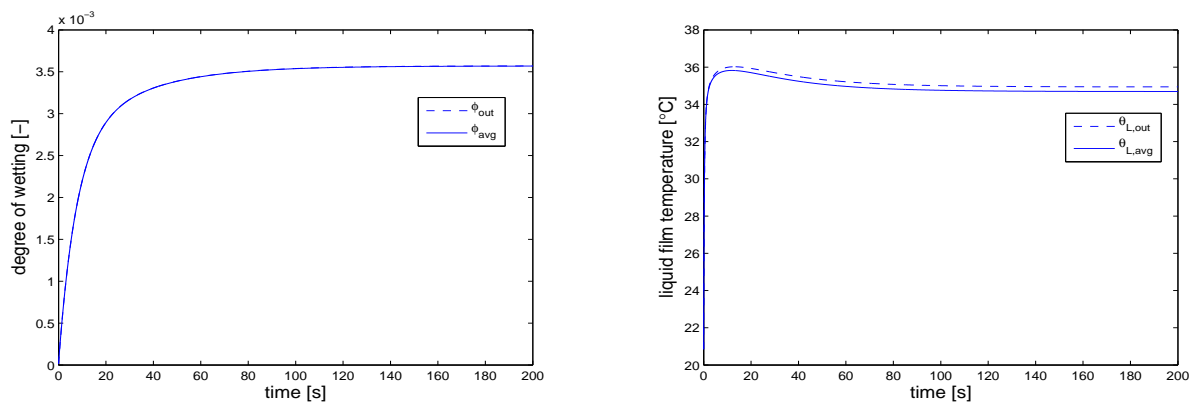


Figure 5.3: Simulations of outlet and average values of degree of wetting and liquid film temperature.

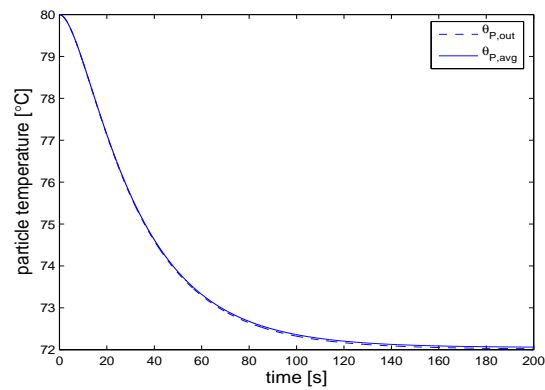


Figure 5.4: Simulations of outlet and average values of particle temperature.

temperature of the sprayed liquid ($20\text{ }^{\circ}\text{C}$) up to $35\text{ }^{\circ}\text{C}$ and then moves toward the steady state final value. Note that the liquid film temperature is non-monotone due to the fact that an equilibrium is reached between heat supplied by the gas and heat loss due to evaporation which sets in later. Initially the heat flow from the gas and the particles dominates the liquid film enthalpy. Thus the liquid film temperature increases rapidly. Thereby the evaporation becomes more pronounced. But initially it is not strong enough to prevent overshooting of the equilibrium temperature of the liquid film.

Influence of the liquid film thickness

The surface of the solid particles, the mass transfer surface between the liquid on the particles and the air as well as the density of the liquid are time-independent. Thus, according to Eq. (2.8) only the liquid film thickness F onto the particles has an influence on the maximal liquid mass per volume element. In Figures 5.5, 5.6, and 5.7 the temporal progressions of the balance equations are shown for the first 200 seconds under variation of the liquid film thickness. We can observe from the figures that for different liquid film thicknesses all the balance quantities reach the same steady-state, but differ in transition time. This transition time for the water evaporation is shorter at smaller liquid film thicknesses, because the mass of water is lower. From the physical point of view, the actual liquid film thickness depends on the surface tension between the solid and the injected liquid.

Influence of the ratio of heat transfer coefficient

The stationary solution of the balance equations is computed under the assumption that the gas to air mass-transfer coefficient is depending only on the saturation loading and/or the saturation vapor pressure at the gas-liquid phase boundary. To compute this quantity, the temperature at the phase boundary is required. This corresponds to a uniform liquid film temperature according to our model. This temperature is the result of a liquid energy balance. If the particle temperature and film temperature are uniform, a jump in temperature will occur at the contact area between particle and film, though this is physically impossible. Nevertheless, in order to utilize this model the thermal conduction between particle and film is approximated by the heat transfer coefficient between particle and film.

The ratio of the heat transfer coefficient f between air and the particles as well as the liquid film is defined in Eq. (2.2). For different liquid film thicknesses F and different heat transfer coefficients, the outlet values of air humidity, air temperature, degree of

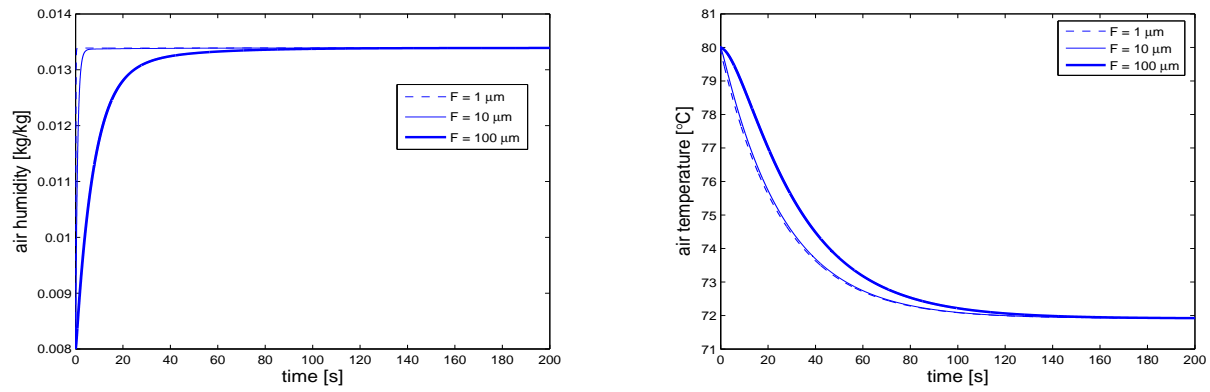


Figure 5.5: Simulation of air humidity and air temperature for different liquid film thicknesses at the outlet.

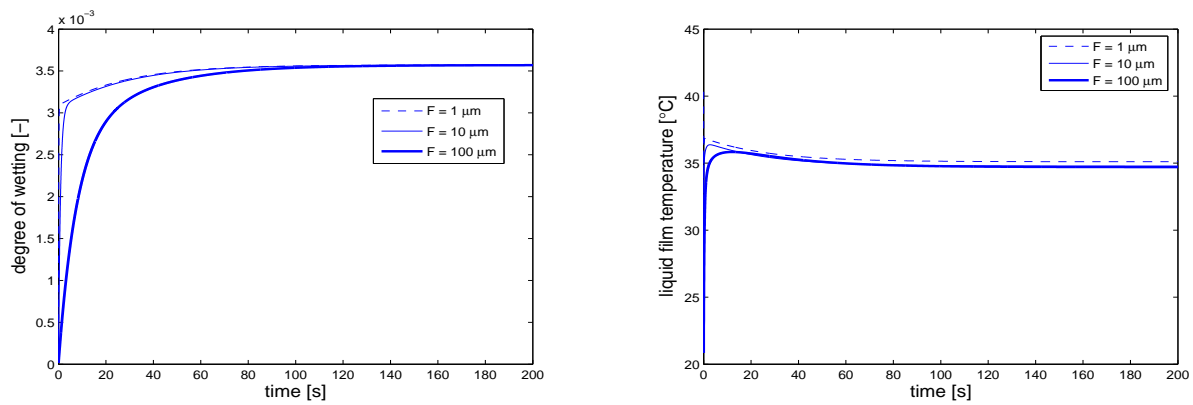


Figure 5.6: Simulation of degree of wetting and liquid film temperature for different liquid film thicknesses at the outlet.

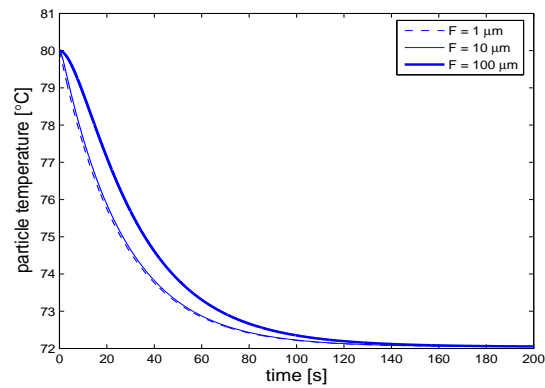


Figure 5.7: Simulation of particle temperature for different liquid film thicknesses at the outlet.

wetting, liquid film temperature and particle temperature are shown in Figures 5.8, 5.9, and 5.10. We can observe from the results that at liquid film thickness $F = 1 \mu m$ the outlet air humidity is nearly identical for different heat transfer coefficients, because f has no effect on the evaporation flow. Analogously at the outlet, air temperature and particle temperature are also identical for different heat transfer coefficients. In the case of increase of f at the liquid film thickness $F = 100 \mu m$, a faster decrease of the outlet air temperature occurred towards the final value, and the particle temperature also decreases faster, which is shown in Figures 5.11, and 5.13. If we observe the degree of wetting at liquid film thickness $F = 1 \mu m$ and $F = 100 \mu m$, increase in the heat transfer coefficient results in a decrease of the degree of wetting at the outlet values, see in Figures 5.9 and 5.12. In the case of a better heat transfer between the particles and the liquid film, a higher liquid film temperature occurs with a smaller degree of wetting when the evaporation flow is constant independently of the thickness of the liquid film. In other words, if the liquid film temperature increases the saturation humidity of the gas and thus the driving force for the liquid evaporation increases.

The calculated size of the wetted surface depends on the calculation of the liquid film temperature and thus on the size of the heat transfer between particle and liquid film described by the factor f . The process can be described with the help of the Mollier diagram. An increase in f leads to a higher liquid film temperature. A slight increase of this temperature is due to the small upward gradient of the saturation line with a significant larger saturation loading of the gas in direct proximity of the phase boundary. This increment of the saturation loading entails an increased driving force for the liquid evaporation. As more liquid is evaporated the mass transfer surface will be smaller. Since the effects of the driving force and the mass transfer surface are moving in opposite directions, the evaporated liquid mass flow remains almost constant. Thus, an incorrect balance of the liquid enthalpy and the liquid mass in the bed will have no measurable effect on temperature and humidity. Evaluation of the quality of the calculated mass transfer surface can be made only by evaluating the conversion of a gas component deviating from the water vapor. The kinetics for the description of this conversion should be limited only by the particle surface and the gas mass transfer. Sulfur dioxide absorption in the liquid-sprayed fluidized bed may possibly be estimated by the size of the wetted surface controls, see [53]. An investigation of the model on the basis of experimental data shows that for a certain bed material, e.g. glass spheres of a certain particle diameter, an f value exists which describes all experiments. The experiments with small values for f are easily reproducible - a fact

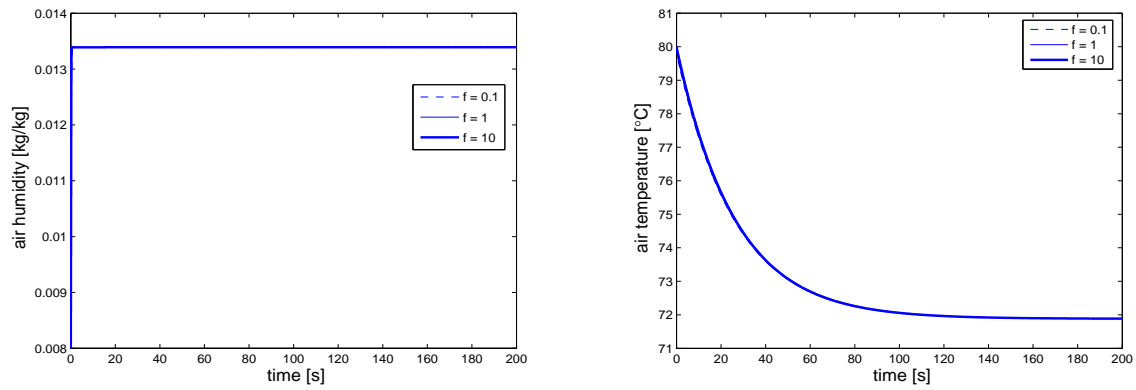


Figure 5.8: Simulations of the outlet values of air humidity and air temperature for different heat transfer ratios at liquid film thickness $F = 1 \mu\text{m}$.

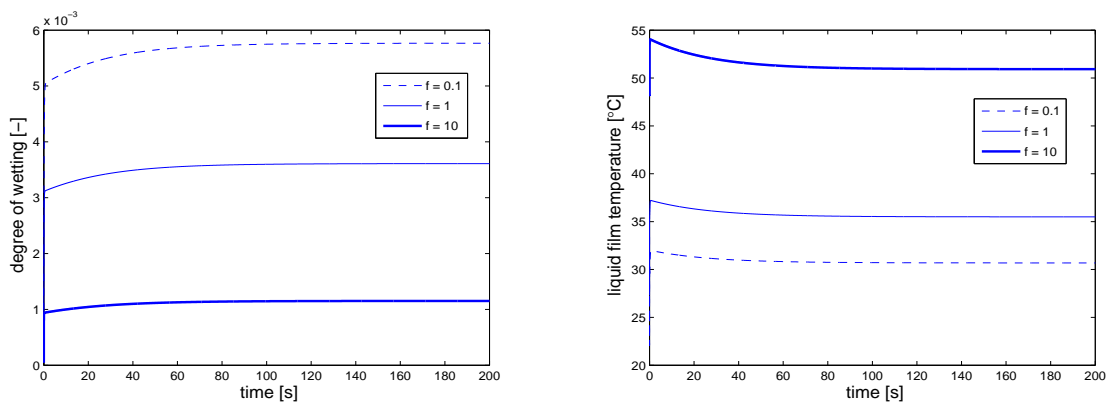


Figure 5.9: Simulations of the outlet values of degree of wetting and liquid film temperature for different heat transfer ratios at liquid film thickness $F = 1 \mu\text{m}$.

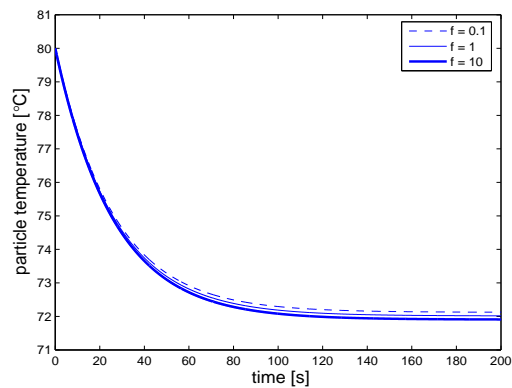


Figure 5.10: Simulations of the outlet values of particle temperature for different heat transfer ratios at liquid film thickness $F = 1 \mu\text{m}$.

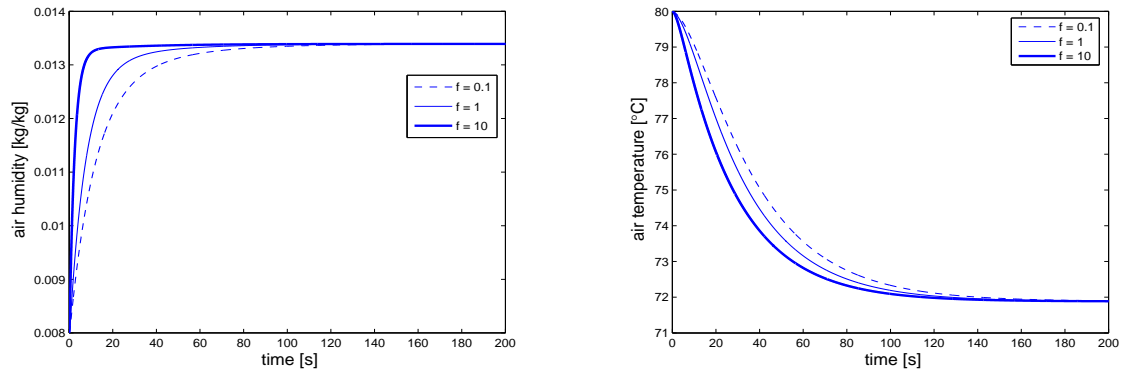


Figure 5.11: Simulation of the outlet values of air humidity and air temperature for different heat transfer ratios at liquid film thickness $F = 100 \mu m$.

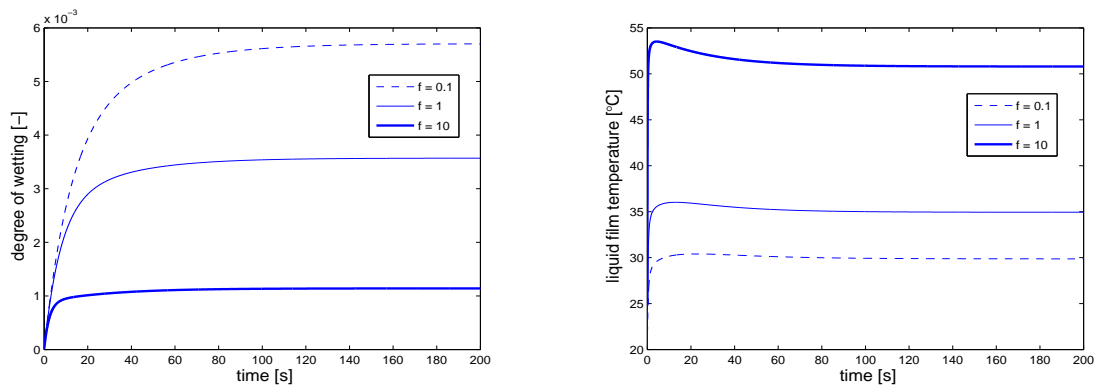


Figure 5.12: Simulation of the outlet values of degree of wetting and liquid film temperature for different heat transfer ratios at liquid film thickness $F = 100 \mu m$.

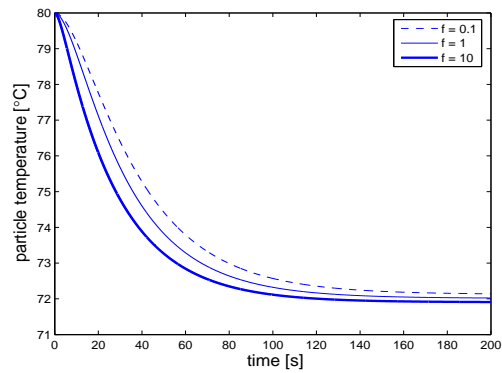


Figure 5.13: Simulation of the outlet values of particle temperature for different heat transfer ratios at liquid film thickness $F = 100 \mu m$.

which can be proven with the help of a high contact resistance.

Influence of the air inlet temperature

For different air inlet temperatures, all balance quantities are depicted in Figures 5.14, 5.15, and 5.16. We can observe from the results that a higher air inlet temperature results in a higher air temperature at the outlet. This leads to a more intense heating up of the liquid film. With the increase of the liquid film temperature, the saturation humidity of the air on the boundary surface increases and consequently increases the term $(Y_{sat} - Y_A)$.

Influence of the air mass flow rate

The effect of different air mass flow rates on the time-dependent balance quantities are shown in Figures 5.17, 5.18, and 5.19. We can observe from Figure 5.17 that the air temperature is different for different air mass flow rates at the outlet, according to the total balance equation, which is shown in Eq. (2.16). In the case of a smaller dry air mass flow, the water balance of the air around the fluidized bed yields a higher outlet air humidity, and thus there is a big difference between the air inlet and outlet temperatures. The steady final values of the liquid film temperature deviate slightly from one another. The saturation humidity Y_{sat} occurring as a function of the liquid film temperature is the same in all calculations.

Influence of the liquid mass flow rate

An increase in the liquid injection rate results in a larger degree of wetting. If the effective wetted surface becomes larger, the air temperature at the outlet, the particle temperature as well as the liquid film temperature decrease as a result of the enhanced evaporation flow. This is shown in Figures 5.20, 5.21, and 5.22. The evaporation flow increases, which leads to a larger steady final value of air humidity at the outlet of the fluidized bed, see Figure 5.20.

Influence of the diameter of particles

An increase in the diameter of the bed particles results in a larger Re -number at constant gas mass flow and bed mass. The larger Re -number leads to an increased Sh -number and a decreased mass transfer coefficient gas-particle $\beta = \frac{D_{AL}(\theta_A)Sh(\theta_A)}{d_p}$. Reduced Re -, Ar - and Sh -numbers do not provide a large gas mass transfer coefficient, as this will decrease

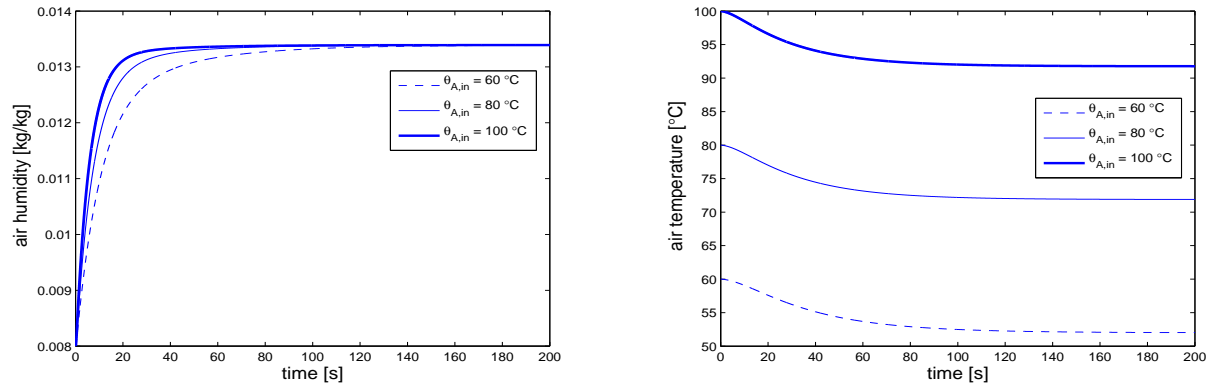


Figure 5.14: Simulation of air humidity and air temperature for different inlet air temperatures at the outlet.

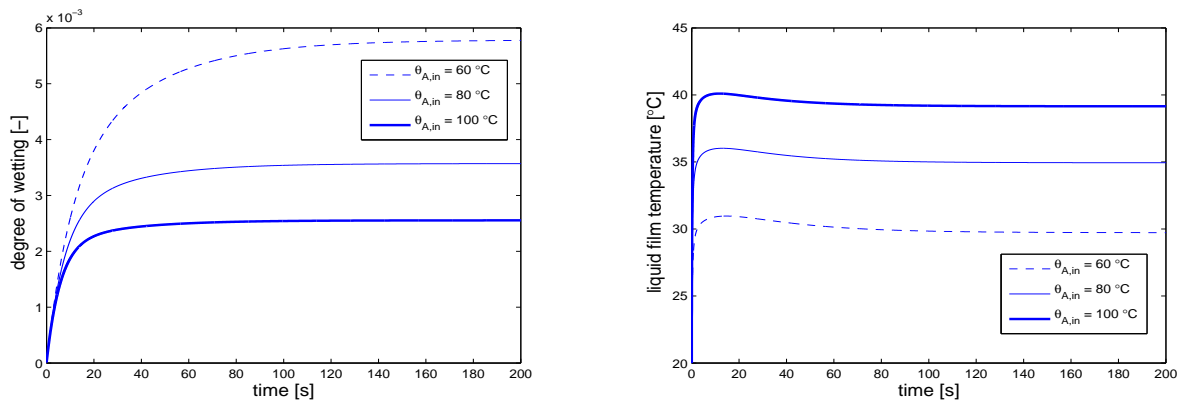


Figure 5.15: Simulation of degree of wetting and liquid film thickness for different inlet air temperatures at the outlet.

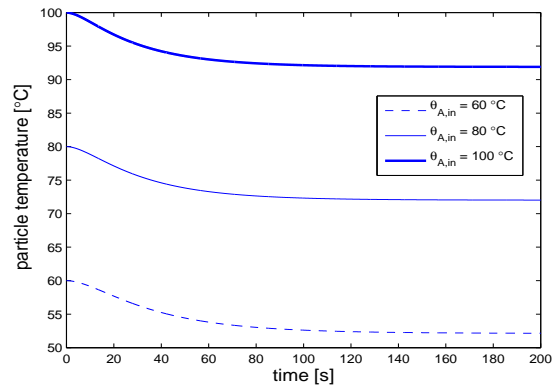


Figure 5.16: Simulation of particle temperature for different inlet air temperatures at the outlet.

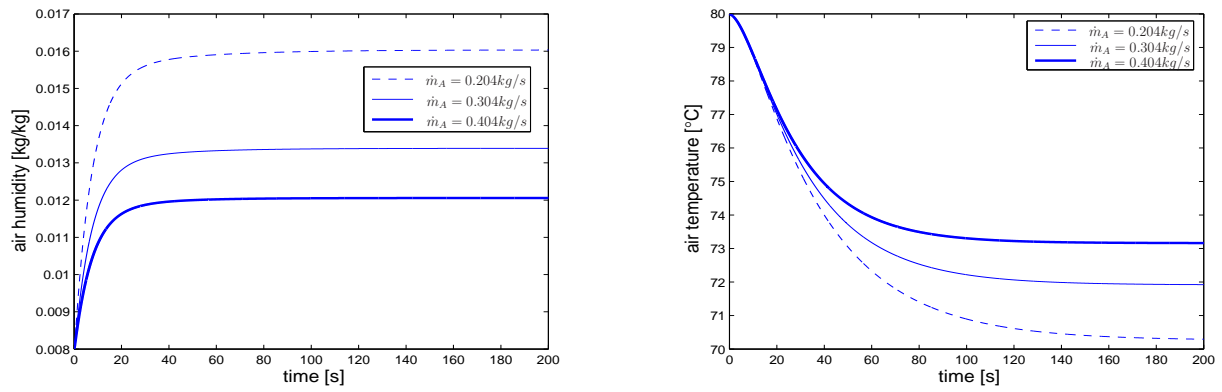


Figure 5.17: Simulations of air humidity and air temperature for different air mass flow rates at the outlet.

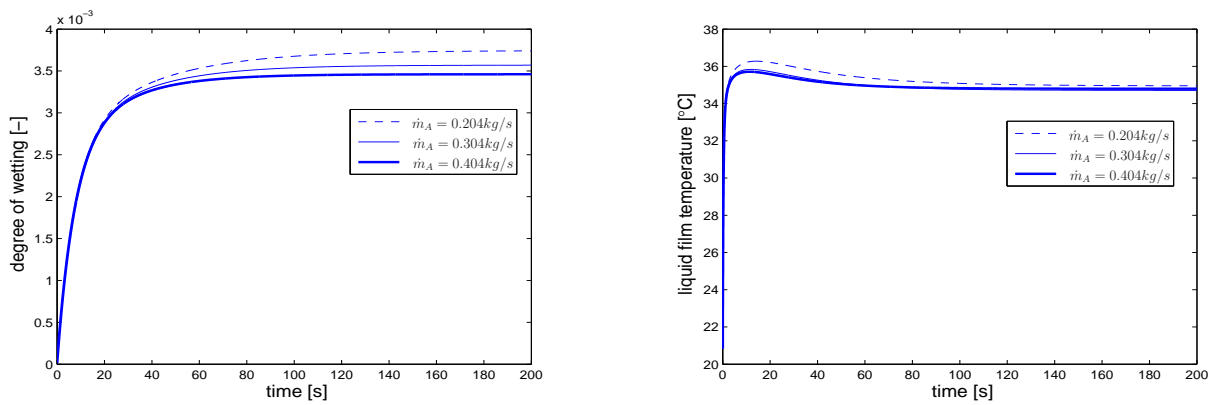


Figure 5.18: Simulations of degree of wetting and liquid film temperature for different air mass flow rates at the outlet.

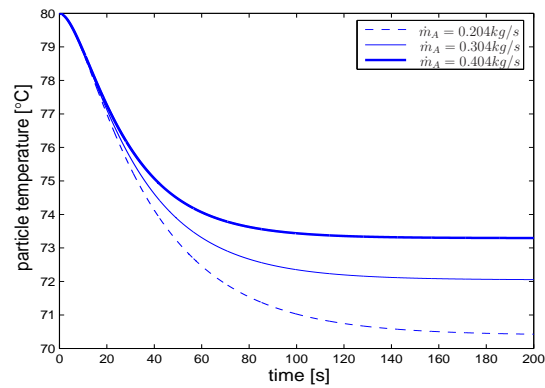


Figure 5.19: Simulations of particle temperature for different air mass flow rates at the outlet.

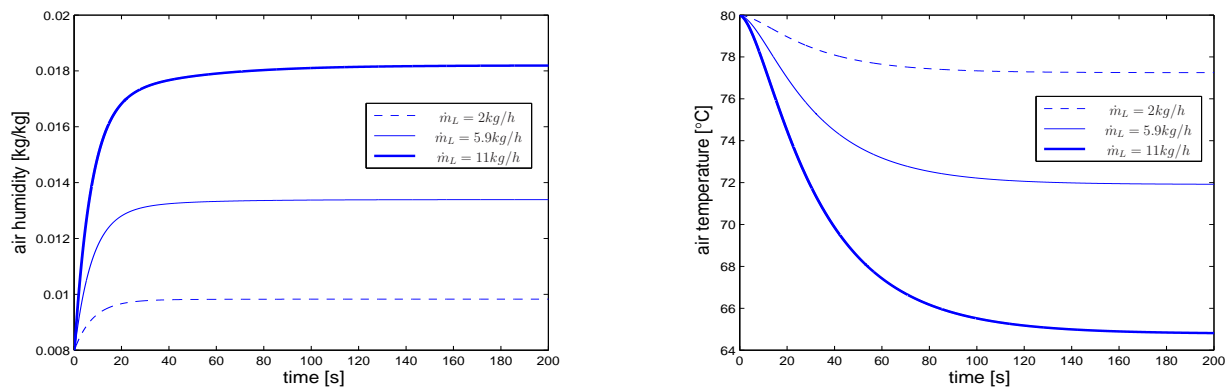


Figure 5.20: Simulations of air humidity and air temperature for different liquid mass flow rates at the outlet.

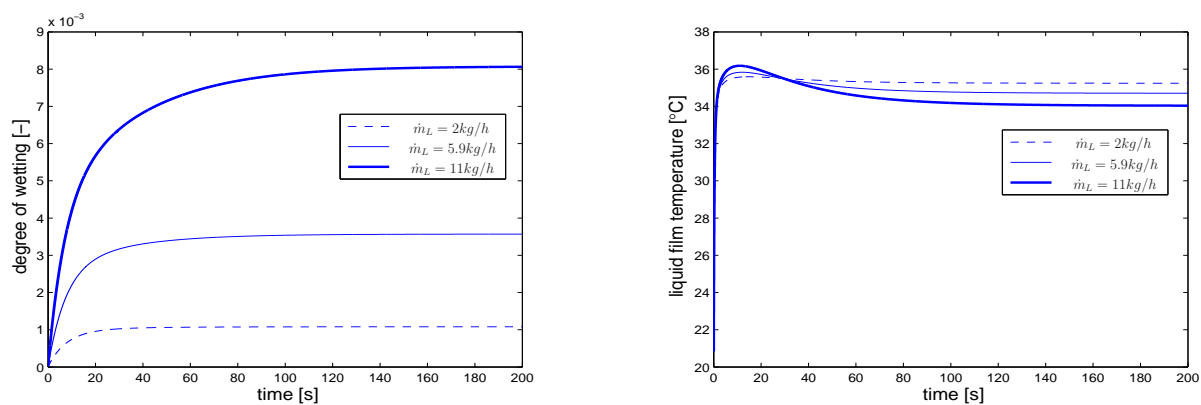


Figure 5.21: Simulations of degree of wetting and liquid film temperature for different liquid mass flow rates at the outlet.

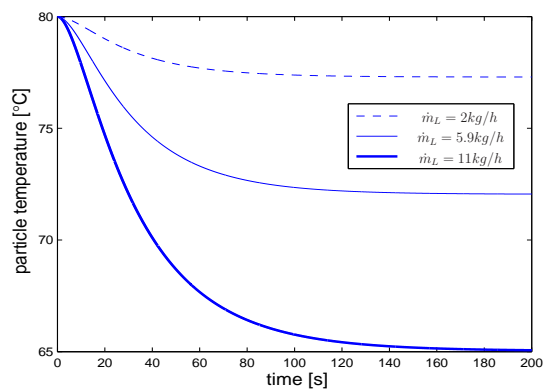


Figure 5.22: Simulations of particle temperature for different liquid mass flow rates at the outlet.

due to diameter growth and thus lead to larger Sh -numbers. On the other hand a small particle diameter corresponds to an increase of the mass transfer surface, which leads to a higher degree of wetting. The simulated balance quantities at different particle diameters are illustrated in Figures 5.23, 5.24, and 5.25. We can observe from the results, that the air humidity reaches the steady final value faster for smaller particles. The final temperatures of air, particles and liquid film are higher.

5.1.2 Two-dimensional simulation results with non-uniform liquid distribution

In this section we present the time dependent numerical solution of the air humidity, air temperature, degree of wetting, liquid film temperature and particle temperature, taking into account a liquid spray nozzle injection which was described in Chapter 2.3. And we give the numerical solution at the particular time $t = 200$ s. We describe non-ideal particle mixing and thus non-uniform liquid distribution by assuming axial and radial particle dispersion coefficients, see Chapter 2.

We considered computational domain $\Omega = [0, 0] \times [0.4, 0.2]$, i.e. the diameter of the apparatus is 0.4 m and the height of the fluidized bed is 0.2 m, as shown in Figure 5.28. We assume that the spraying has occurred at the top surface of the apparatus and the position of the nozzle is fixed at (0.2, 0.2) and with a spraying angle of 60° . For simulations we used the initial parameters as listed in Table A.1, whereby a constant wall temperature is assumed. For the simulations the deposition efficiency is kept constant with $\phi_{dep} = 20$ %. According to Eq. (2.23), the representation of the mass flow of the liquid in Figure 5.28 shows a complete deposition of the liquid droplets onto the particles after some centimeters near the nozzle region. This means that the evaporation also takes place primarily in the nozzle region.

We observed from the simulation results that the air humidity increases almost linearly in the axial direction with the distance from the distributed plate. When spraying of the liquid starts, the air humidity increases suddenly, later it increases slowly until it reaches the stationary solution. After an initial time we can see high air humidity near the nozzle. As time goes on it distributes uniformly at the top surface, see left hand side Figure 5.26. Air temperature decreases from the distributor plate to the top. After reaching the stationary solution, it coincides with outlet air temperature determined from the total balance, see right hand side Figure 5.26. The particle temperature decreases slower than the air temperature because of the particle heat capacity. During stationary operation it

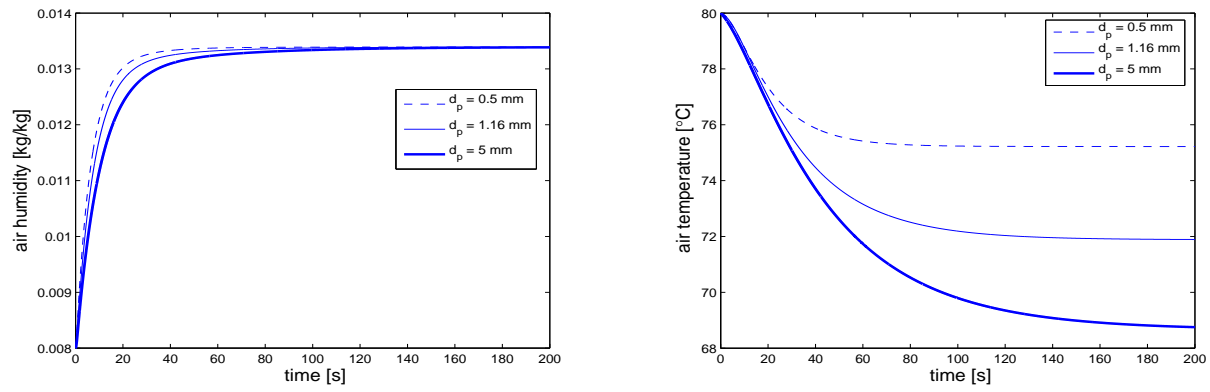


Figure 5.23: Simulations of air humidity and air temperature for different particle diameters at the outlet.

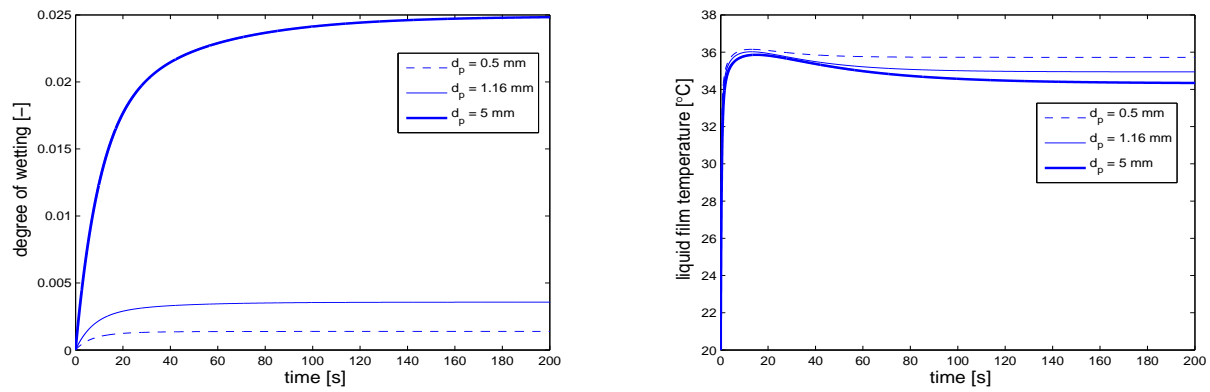


Figure 5.24: Simulations of the degree of wetting and liquid film temperature for different particle diameters at the outlet.

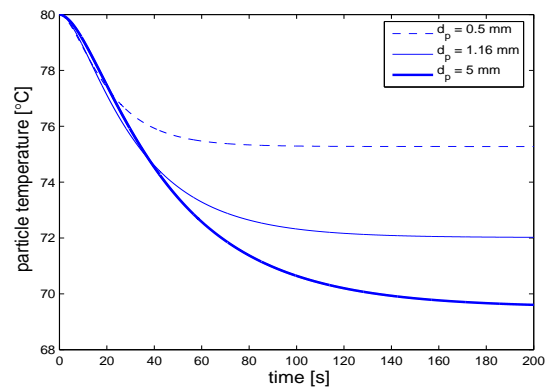
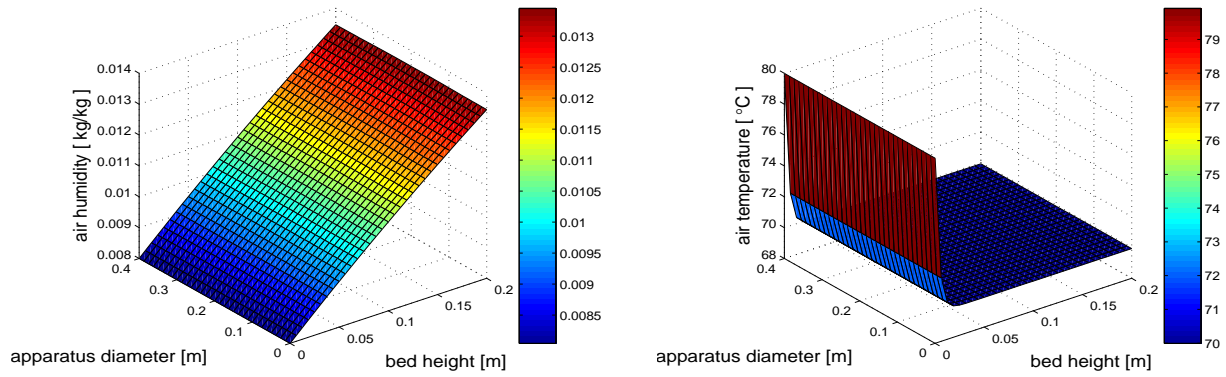
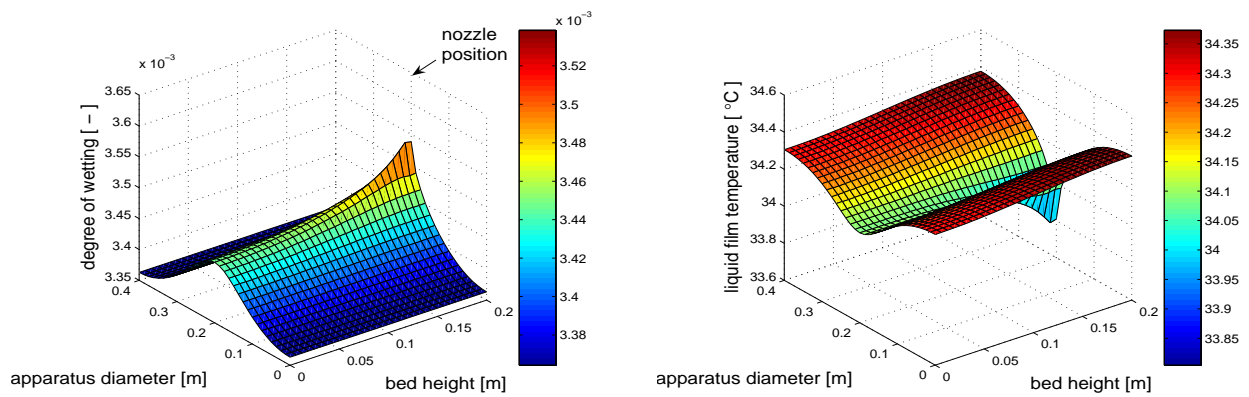
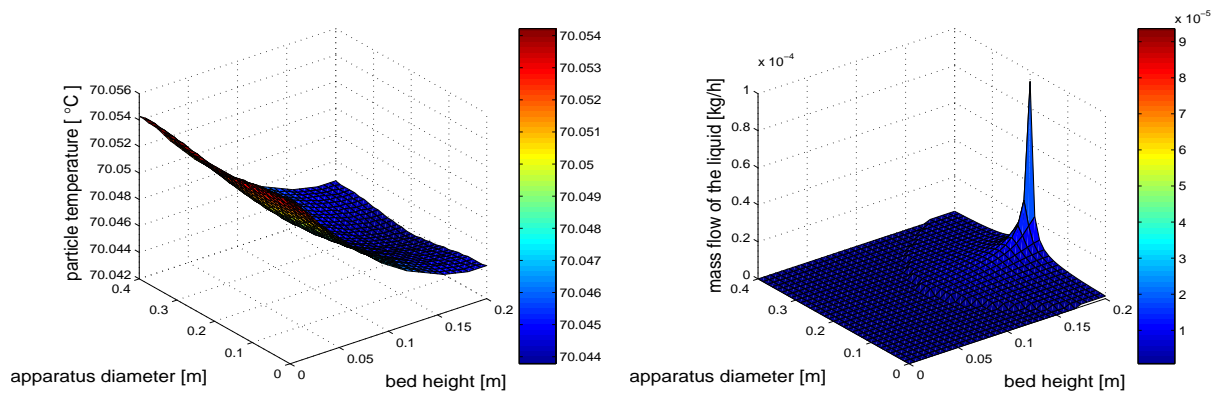


Figure 5.25: Simulation of particle temperature for different particle diameters at the outlet.

Figure 5.26: Simulation of air humidity and air temperature at time $t = 200s$.Figure 5.27: Simulation of degree of wetting and liquid film temperature at time $t = 200s$.Figure 5.28: Simulation of particle temperature at time $t = 200s$ and mass flow of the liquid from spray nozzle with spraying rate $\dot{m}_L = 6kg/h$.

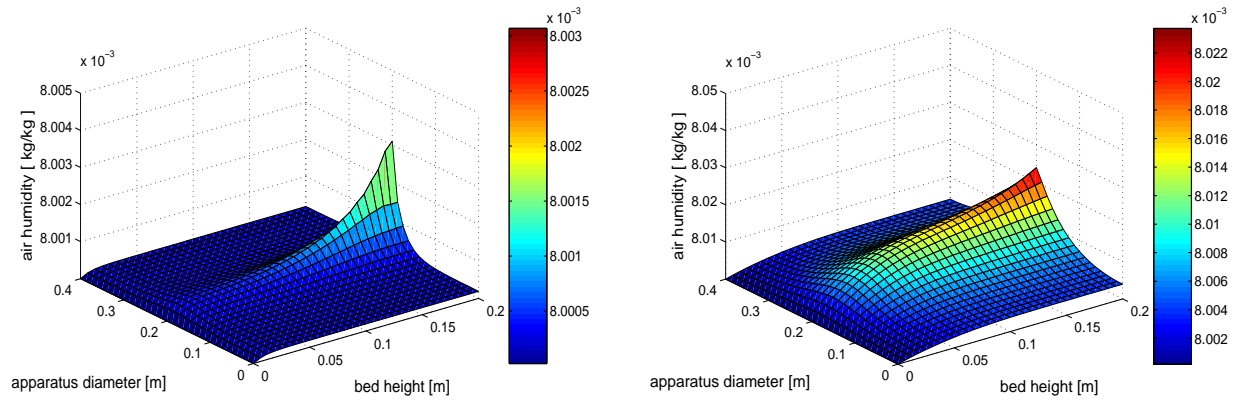


Figure 5.29: Simulation of air humidity at time $t = 0.01s$ and $t = 0.1s$.

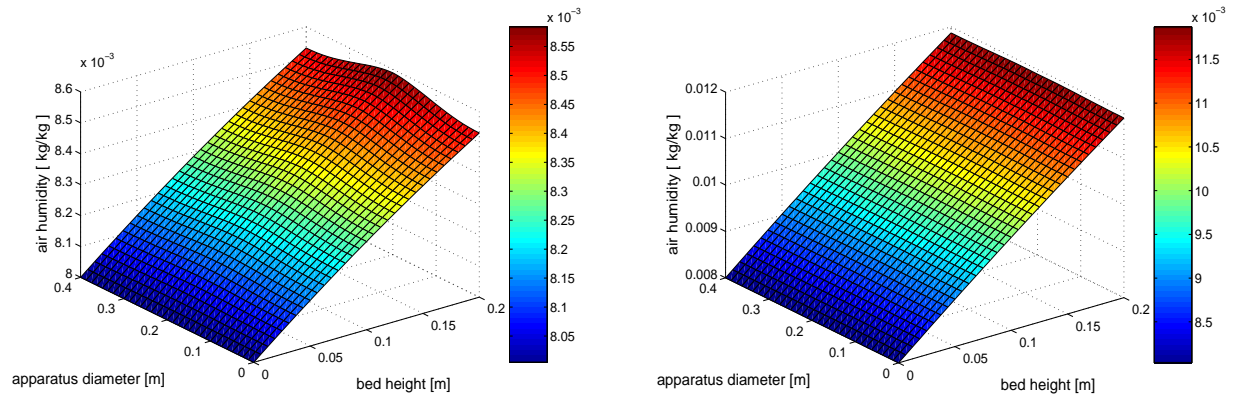


Figure 5.30: Simulation of air humidity at time $t = 1s$ and $t = 10s$.

lies somewhat under the air temperature. The particle temperature is practically locus dependent due to the high air-particle and air-liquid film heat transfers, see left hand side Figure 5.28. The wetting efficiency is high near the nozzle and it decreases from the top to bottom. At initial time $t = 0$ the average of degree of wetting is 10^{-8} . When the simulation starts it increases sharply to 0.30% near the nozzle and thereafter increases steadily towards the final value 0.35%. This is due to the fact that the temperature of the liquid is $20\text{ }^{\circ}\text{C}$ and so the liquid film temperature sinks where energy is absorbed from the particles and increases where the energy is emitted, see left hand side Figure 5.27. Here we did not show the patterns of the parameters locally in 2D computations because their behavior is similar to the 1D pattern. Here we cannot show the time dependent behavior of all the parameters, we show only the behavior of air humidity at different time levels, see Figures 5.29, 5.30.

Influence of the position of liquid injection

To predict the influence of the local position of the liquid injection, we carried out simulations with two nozzles and four nozzles from top to bottom.

The liquid spraying was evenly distributed through two nozzles, with $\dot{m}_L = 6 \text{ kg/h}$ in both cases and other parameters are listed in Table A.1. Two nozzles were arranged, each with a spraying angle 60° and the nozzle positions are:

- * nozzle 1 is fixed at $(x, y) = (0.15, 0.2)$, spraying direction from top to bottom,
- * nozzle 2 is fixed at $(x, y) = (0.25, 0.2)$, spraying direction from top to bottom.

Figures 5.31, 5.32, and 5.33 represent the simulation results of the balance variable distributions Y_A , θ_A , ϕ , θ_L , θ_P and Y_{sat} at time $t = 200 \text{ s}$. The mass flow of the liquid distribution per volume element is defined in Eq. (2.23). Due to the higher total liquid injection rate in comparison to the previous case with only one nozzle, the air humidity in the fluidized bed as well as the saturation humidity is higher and all temperatures are lower. We can observe from the Eq. (2.9) that when the liquid spraying is started from the nozzles the degree of wetting is increased suddenly. We can observe from the surface plots that in case of the degree of wetting, more liquid is deposited near the nozzles and the distribution is decreases with further distance from the nozzles. When the degree of wetting increases, the evaporation flow term, $\beta\rho_A A_P \phi(Y_{sat} - Y_A)$ increases, so from Eq. (2.5), the air humidity distribution increases near the nozzle. Air temperature and particle temperature decrease near the nozzle. Again, we can see a constant air temperature in axial direction in the fluidized bed and the air temperature decreases strongly with the distance from the gas distributor plate.

Next, the liquid spraying was evenly distributed through four nozzles, with $\dot{m}_L = 6 \text{ kg/h}$ in all cases and other parameters as listed in Table A.1. Four nozzles were arranged, each with a spraying angle of 60° . The nozzle positions are:

- * nozzle 1 is fixed at $(x, y) = (0.08, 0.2)$, spraying direction from top to bottom,
- * nozzle 2 is fixed at $(x, y) = (0.16, 0.2)$, spraying direction from top to bottom.
- * nozzle 3 is fixed at $(x, y) = (0.24, 0.2)$, spraying direction from top to bottom,
- * nozzle 4 is fixed at $(x, y) = (0.32, 0.2)$, spraying direction from top to bottom.

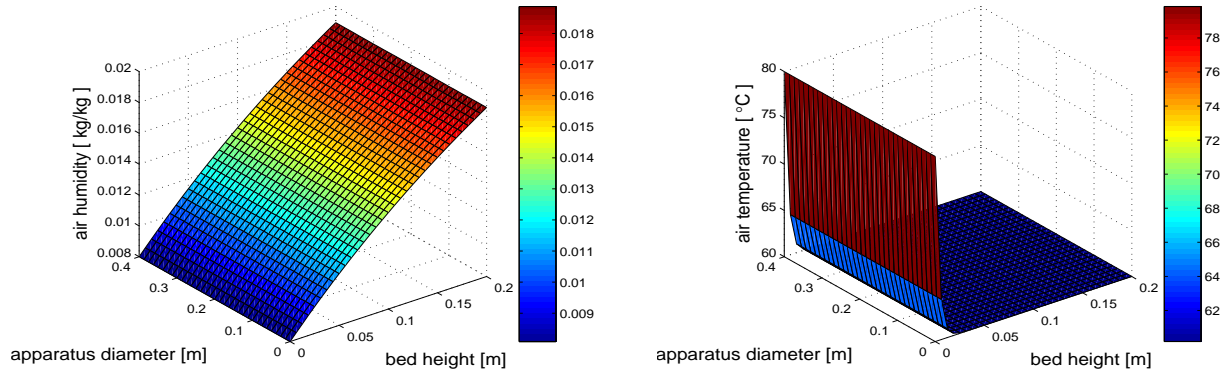


Figure 5.31: Simulation of air humidity and air temperature at time $t = 200s$ with two sprayed nozzles.

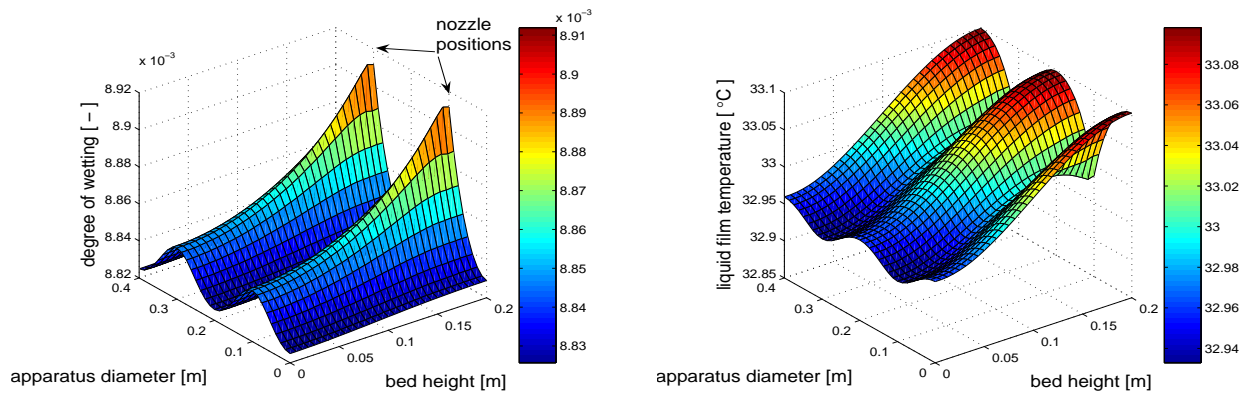


Figure 5.32: Simulation of degree of wetting and liquid film temperature at time $t = 200s$ with two sprayed nozzles.

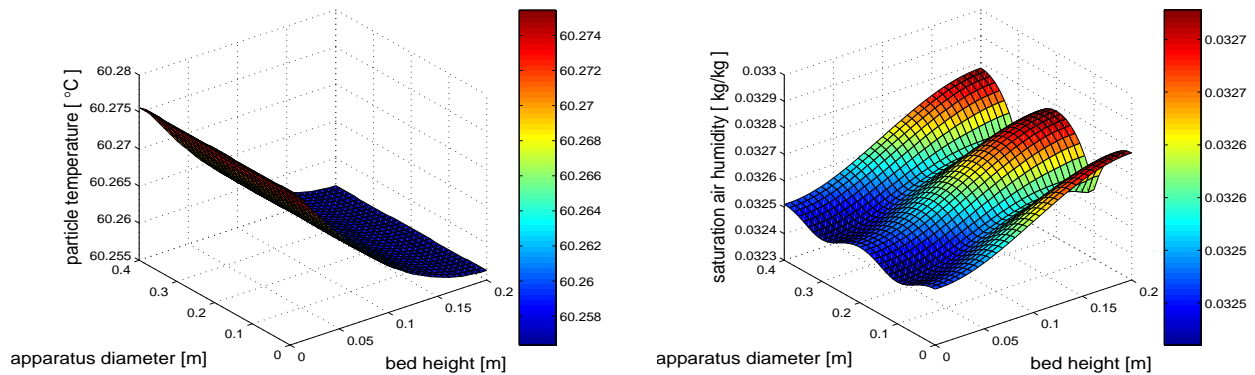


Figure 5.33: Simulation of particle temperature and saturation air humidity at time $t = 200s$ with two sprayed nozzles.

Figures 5.34, 5.35, and 5.36 represent the simulation results of the balance variable distributions Y_A , θ_A , ϕ , θ_L , θ_P and Y_{sat} at time $t = 400s$. We can observe that the increased injection rate results in higher humidities close to the nozzles and the air humidity distribution versus the bed height is more digressive than linear. The particle temperature distribution reaches almost the point of saturation, characterized by low differences between liquid film temperature, which are related to the point of saturation, and the particle temperature. The increase of the degree of wetting is more intensive compared to the injection with two nozzles, which may lead in practice to agglomeration effects, see Nagaiah et al. [71] for some more results.

Numerical tests

The left hand side of Figure 5.37 shows the comparison between the implicit and semi-implicit schemes. In the implicit case we used a Newton solver describer in Subsection 4.6.1 for linearizing the algebraic nonlinear equations arising from the time discretization. For solving the linear system we used the BiCGSTAB method [92] with preconditioner. Here we used the multigrid solver [43] as preconditioner. In the multigrid solver, the BiCGSTAB method is used as a base solver and the ILU method is used for the pre and post smoothing steps. In the case of the semi implicit method instead of taking the Newton solver for linearizing the nonlinear system, we just use the old solution in the Jacobian matrix, then we get a linear system. For solving the linear system we used the BiCGSTAB method with the multigrid solver as preconditioner, as explained above. Here we obtained faster convergence with the implicit Euler scheme compared to the semi implicit scheme, see Figure 5.37. Here the L^2 -norm of the air humidity is plotted up to a simulation time of 20 seconds only for a better view of the transition period. It is observed that both solutions converge almost after 100 seconds of simulation time. The main difference is that the implicit method converges very fast but takes more CPU time compared to the semi implicit method. This is because for linearizing the nonlinear system the Newton method takes some extra CPU time. The right hand side of Figure 5.37 shows the L^2 -norm of the air humidity for the BiCGSTAB solver, the BiCGSTAB solver with ILU preconditioner and the BiCGSTAB solver with multigrid solver as preconditioner as explained as above. These three curves are the same in terms of the convergence but only differ in computational time. For this problem the CPU times are 3h 05m 56s, 1h 45m 43s, 1h 56m 53s respectively needed to obtain them, also see Nagaiah et al. [70] for some numerical results.

In Table 5.1 we explain the experimental order of convergence of the air humidity. The

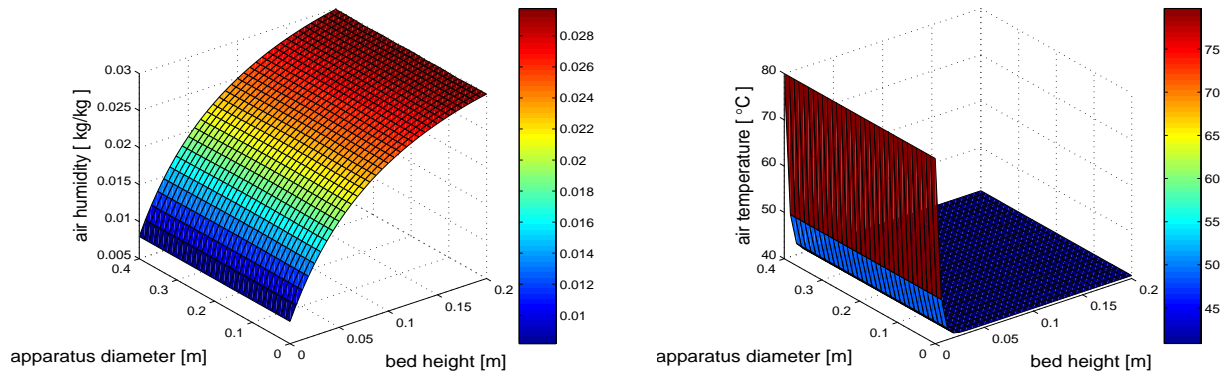


Figure 5.34: Simulation of air humidity and air temperature at air temperature at time $t = 200s$ with four sprayed nozzles.

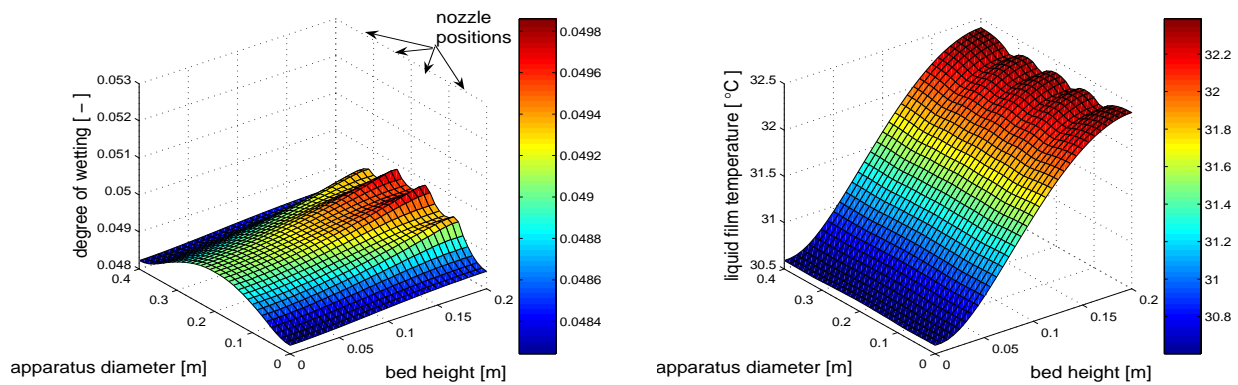


Figure 5.35: Simulation of degree of wetting and liquid film temperature at time $t = 200s$ with four sprayed nozzles.

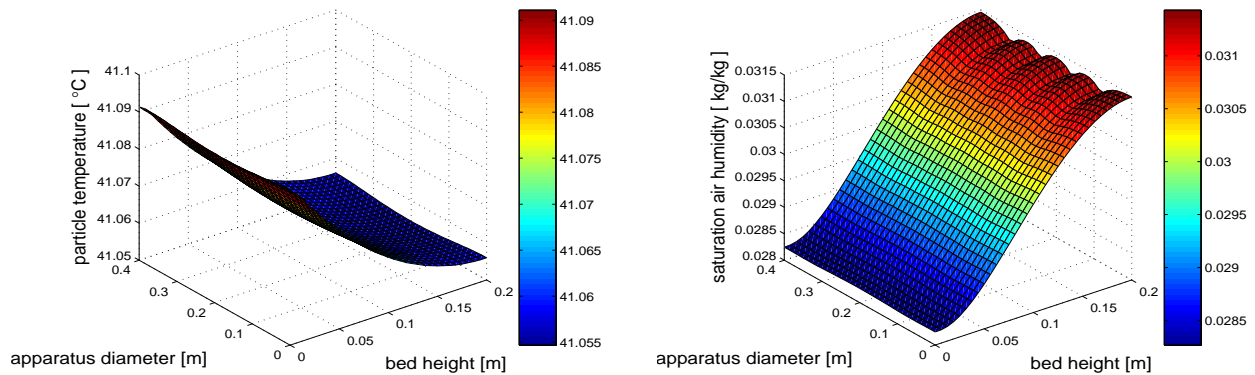


Figure 5.36: Simulation of particle temperature and saturation of air humidity at time $t = 200s$ with four sprayed nozzles.

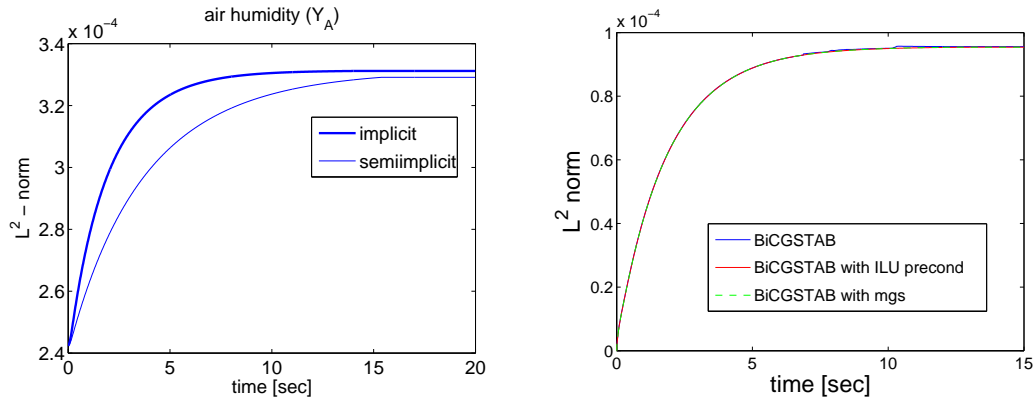


Figure 5.37: Simulation of air humidity and air temperature at air temperature at time $t = 200s$ with four sprayed nozzles

mesh size	$\ u_{h/2} - u_h\ _{L^2}$	EOC	$\ u_{h/2} - u_h\ _{L^\infty}$	EOC
17×17	-	-	-	-
33×33	0.0276871146	0.8854188211	0.02468235	0.9521690121
65×65	0.0511464296	0.9764371353	0.04775490	0.9781248865
129×129	0.1006357271	-	0.09407254	-

Table 5.1: Experimental order of convergence of the air humidity at the stationary solution.

first column indicates the mesh size and the second column shows the L^2 error of the sizes h and $h/2$. The last column represents the experimental order of convergence (EOC), which is defined with respect to the L^p -norm as

$$\text{EOC} = \ln \left[\frac{\|u_{h/2} - u_h\|_{L^p}}{\|u_{h/4} - u_{h/2}\|_{L^p}} \right] / \ln(2), \quad \text{for } (1 \leq p \leq \infty). \quad (5.1)$$

We can observe that the EOC is achieved successfully almost 1 in the case of bilinear finite elements. This result demonstrate that the method has convergence order that agree with theoretical considerations.

5.2 Numerical results in 3D

The three space dimensional transient numerical simulations of balance equations are presented in this section. The fluidized bed zone is represented by a cylinder in the numerical simulation. The initial mesh and successive uniform mesh refinements are done using the programm package UG (Unstructured Grid), see [12]. The hexahedron element, which has

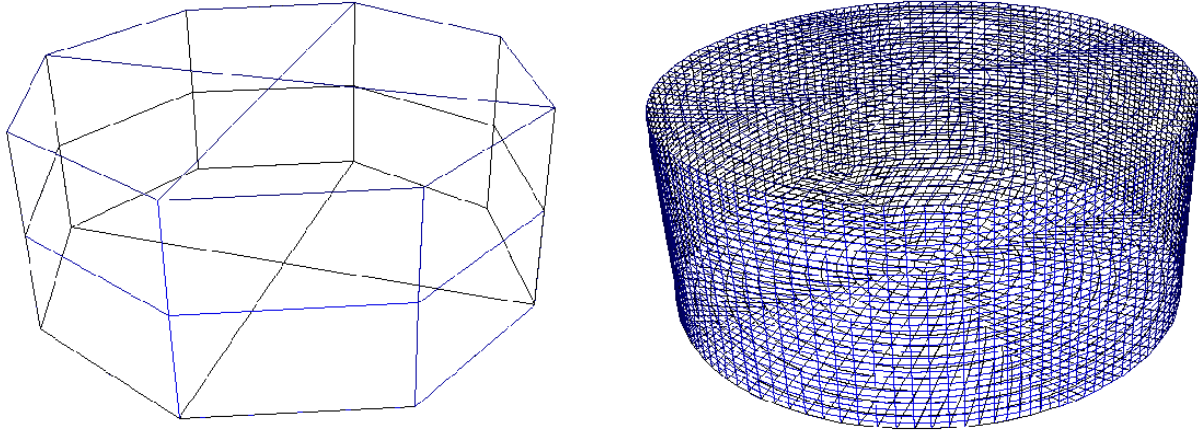


Figure 5.38: Initial mesh and uniform refinement mesh at level 4

8 nodes and 6 surfaces, is used in the mesh generation. The initial mesh (level 0) consists of 8 hexahedron elements, and the fine mesh (level 4) consists of 32,768 elements, that are obtained by using uniform mesh refinement, see Figures 5.38.

The standard Galerkin method is used for space discretization and implicit Euler method is considered for time discretization for this problem. Here, the semi implicit approach is considered in solving the nonlinear algebraic equations, which is explained in Section 4.6. The multigrid method is used for solving the algebraic equations, which are arising after time discretization. The BiCGSTAB method is considered as base solver and ILU method is used as pre- and post- conditioner in multigrid solver.

The simulation results of the balance quantities air humidity, air temperature, degree of wetting, liquid film temperature, particle temperature and saturated air humidity are presented. In the numerical simulations, the sprayed liquid is considered as water. To compare our numerical results with experiments, we have considered 2 sets of experimental data for the numerical simulations. The first set of data taken from the Ph. D thesis of Heinrich [47] and second set of data taken from the Ph. D thesis of Henneberg [51]

5.2.1 Experiment-1

The simulation parameters are shown in Table A.2 for the first test case. In this case one spray nozzle is considered in the simulation, which is fixed at $(0.2, 0.2, 0.2)$ on a cylindrical domain with spray angle $\theta = 45^\circ$. The model equations and our programm allow the

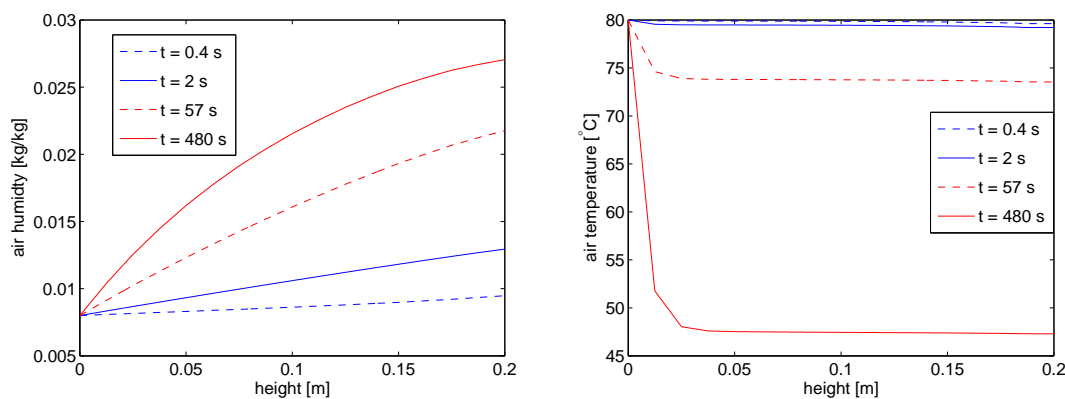


Figure 5.39: Simulation of the air humidity and air temperature for different time steps depending on the distance from the distributor plate to the nozzle center at the top of the fluidized bed.

flexibility to arrange more than one nozzle positions arbitrarily. In the numerical simulation the Eq. (2.29) is used to calculate the spatial profile of the spray nozzle. Here, the bed mass $m_P = 20$ kg and mass flow of the nozzle $\dot{m}_L = 13.79$ kg/h are considered. Initial and boundary values, which are used in simulations, are shown in Table A.2.

In left hand side of Figure 5.39, observe that the air humidity increases almost linearly in the axial direction with the distance from the distributor plate to the nozzle mouth. When liquid is sprayed onto solid particles, the degree of wetting starts increase. Then, the evaporation term $\beta A \phi (Y_{sat} - Y_A)$ also increases. So the distribution of air humidity increases in axial direction due to the increase of evaporation term. And also observe that the air humidity distribution increases rapidly near the nozzle region in the initial period, for example until time 57 s. The reason is that, in this area the liquid on the particle surface evaporates through its contact with air. Therefore, the air humidity is larger in this region. This region can be identified as the mass transfer region. Then the air humidity increases slowly until it reaches the stationary solution. At initial time the air humidity is 0.008 kg/kg and it reaches to 0.0276 kg/kg at the steady state solution, seen clearly in the spatial profile of the air humidity in the top Figure 5.40. Here, the solution is plotted for the stationary solution, at time $t = 600$ s.

The air temperature decreases strongly in the vicinity of the nozzle region, seen clearly in the right hand side of Figure 5.39. This is mainly due to the absorbed energy by colder particles in this region. It is observed that it takes around 600 s to reach the stationary solution. The time dependent spatial profile of the air temperature is depicted

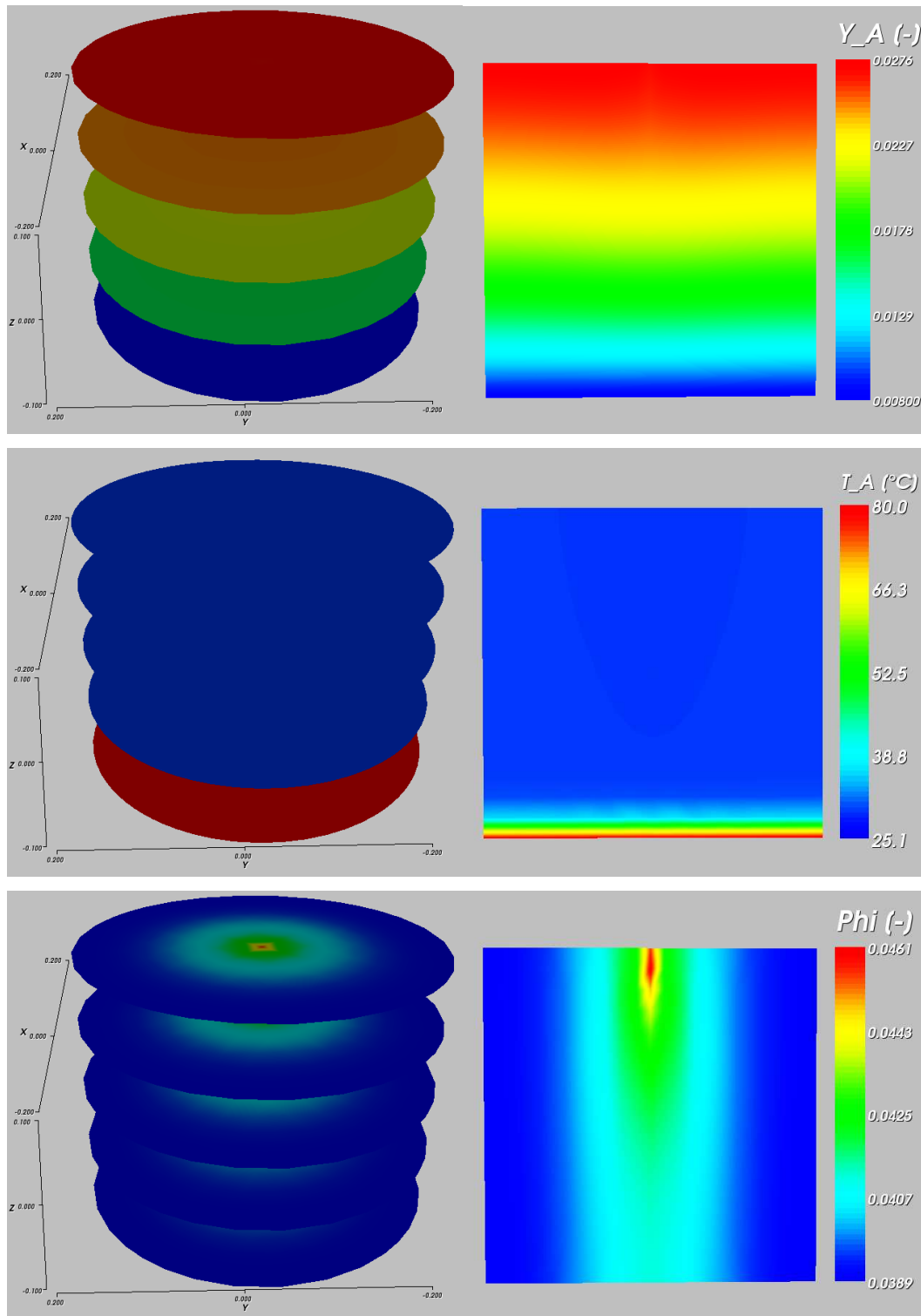


Figure 5.40: Steady state spatial profile of the simulation results of water sprayed fluidized bed at time $t = 600$ s with parameters as shown in Table A.2, **top**: air humidity, **middle**: air temperature, **bottom**: degree of wetting.

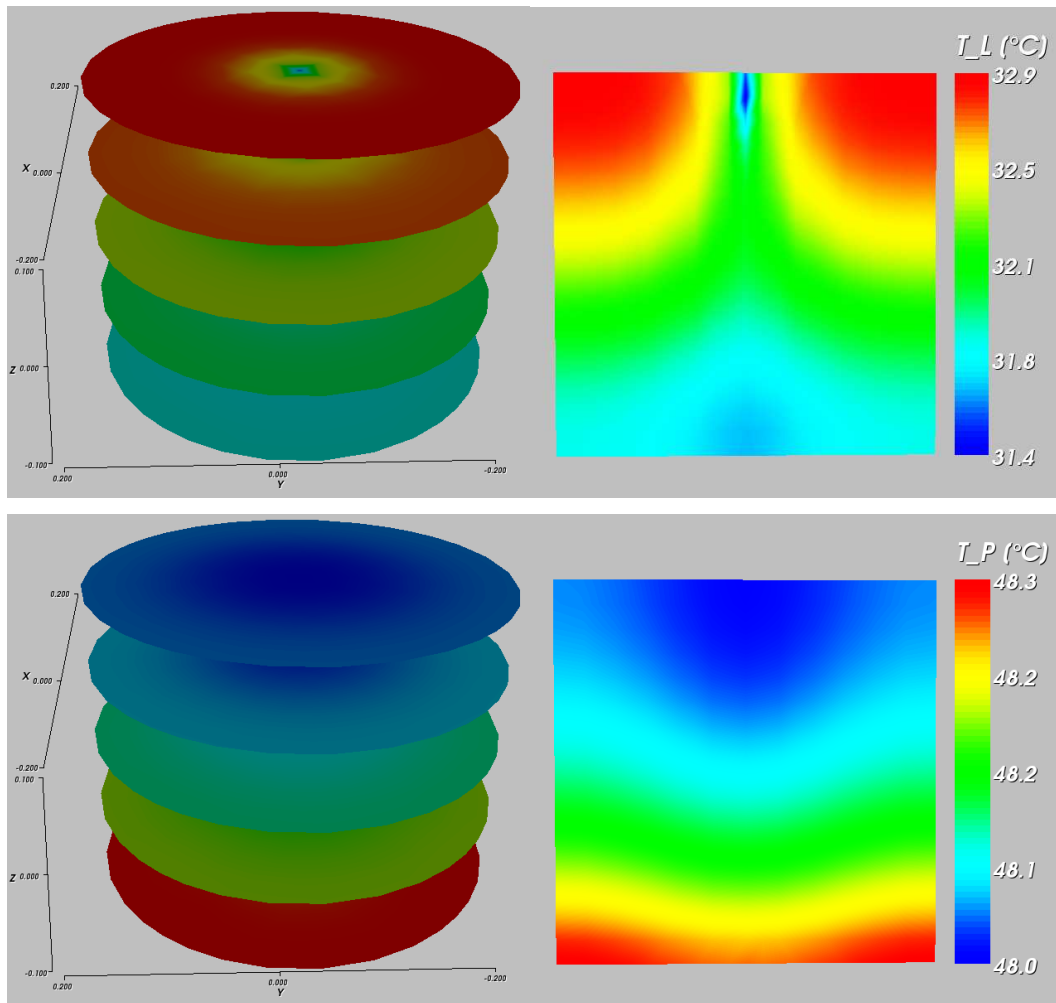


Figure 5.41: Steady state spatial profile of the simulation results of water sprayed fluidized bed at time $t = 600$ s with parameters as shown in Table A.2, **top**: liquid film temperature, **bottom**: particle temperature.

in Figure 5.42 at different time steps. The spatial profile of the air temperature at steady state is shown in the middle of Figure 5.40. These results are compared with experimental results of Heinrich [47]. The coming paragraphs demonstrate the comparison between numerical and experimental results. Here, we have observed that the numerical results are in a good qualitative agreement with the experimental results.

The particle temperature decreases slower than the air temperature due to the particle heat capacity. At the stationary state, the particle temperature lies slightly under the air temperature. It is practically locus independent due to the high air-particle and air-liquid

film heat transfers. The degree of wetting is high near the nozzle and it decreases from the top to bottom. At initial time $t = 0$ s the value of degree of wetting is set to 10^{-8} in our numerical simulations. When the simulation starts it increases sharply near the nozzle and thereafter increases steadily towards the steady state solution, see clearly the steady state solution at $t = 600$ s in bottom of Figure 5.40. This is due to the fact that the temperature of the liquid is 20 °C and so the liquid film temperature sinks where energy is absorbed from the particles and increases where the energy is emitted. The difference between the maximum and minimum values of particle temperature is small. The steady state solution of the liquid film temperature and particle temperature are depicted in Figure 5.41. Here we do not present the pattern of the parameters locally in the 3D computations because their behavior is similar to the 1D pattern.

Comparison with experimental results

The numerical results are validated with the experimental results. The experimental results are measured in a water sprayed fluidized bed plant. For more details regarding experimental set up, we refer to Heinrich [47]. For the comparison, the experimental result of a three dimensional distribution of air temperature over the bed height is depicted in Figure 5.43. Here, the experiments were carried out with considering the direct evaporation in the spray liquid. The direct evaporation means that some liquid evaporates in the sprayed liquid, which is sprayed through the nozzle, due to hot air. One can clearly see the temperature is lower in the spray nozzle zone as compared to other regions. In experimental results, the initial temperature is 80 °C and at the steady state the minimum value reaches 25 °C. The corresponding numerical results can be seen in the middle Figure 5.40. In numerical results, the initial temperature is 80 °C and at the steady state the minimum value reaches 25.1 °C. The numerical results are simulated without considering direct evaporation. Despite this it can be observed that the numerical results are in good agreement with the experimental results qualitatively as described.

5.2.2 Experiment-2

In this subsection, the numerical results are presented using the second set of parameters, which are shown in Table A.3. Here, the domain $(1.5m, 1.5m, 0.6m)$ is considered in numerical simulations. This cross section area is 14 times bigger than the previous plant considered in our first simulation. In this case also one spray nozzle is considered, which is fixed at $(0.75, 0.75, 0.6)$ on the cylindrical domain with spray angle $\theta = 40^\circ$. Here, the bed

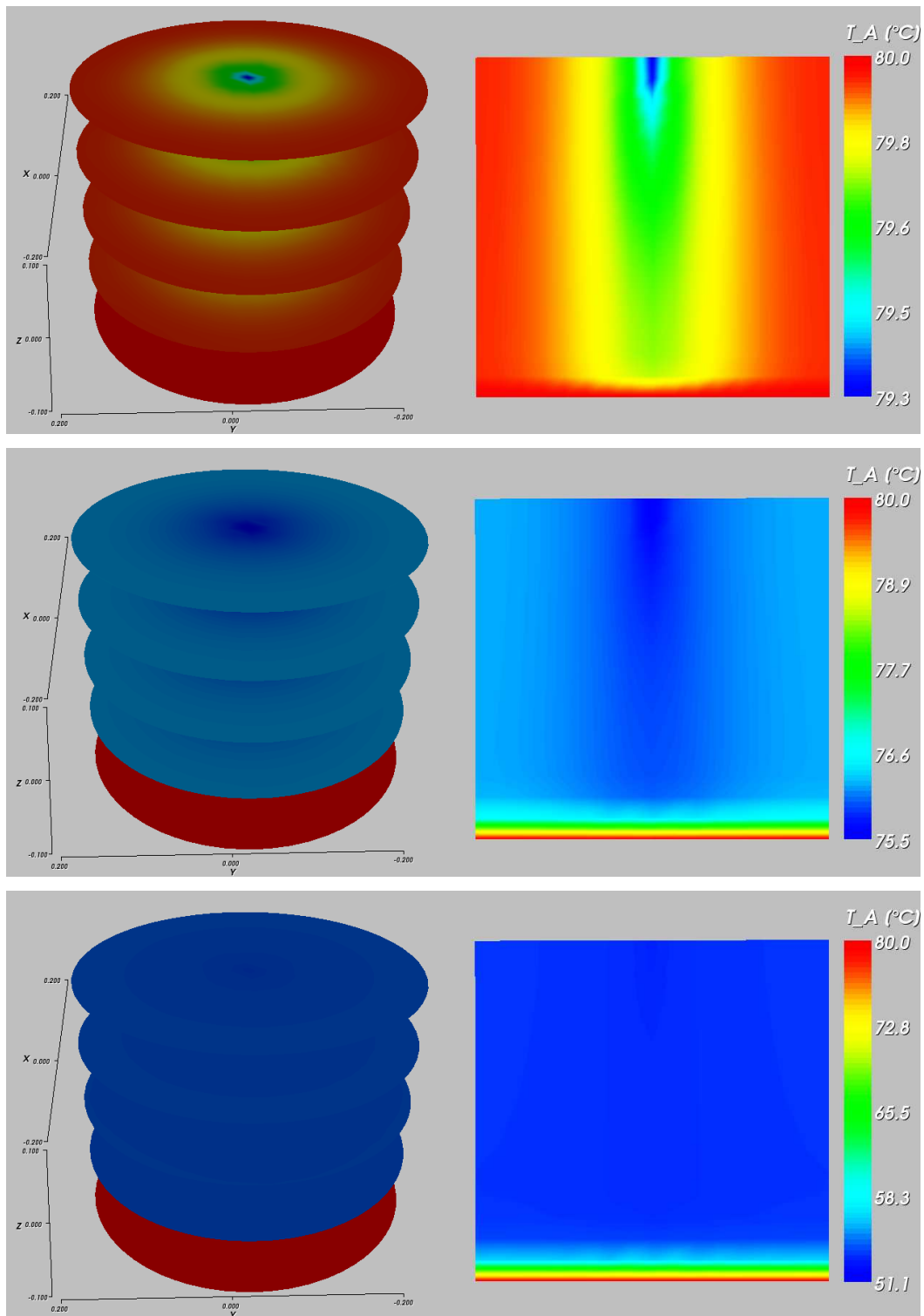


Figure 5.42: Steady state spatial profile of the simulation results of air temperature distribution of water sprayed fluidized bed with parameters as shown in Table A.2 at time $t = 1$ s, $t = 10$ s $t = 100$ s.

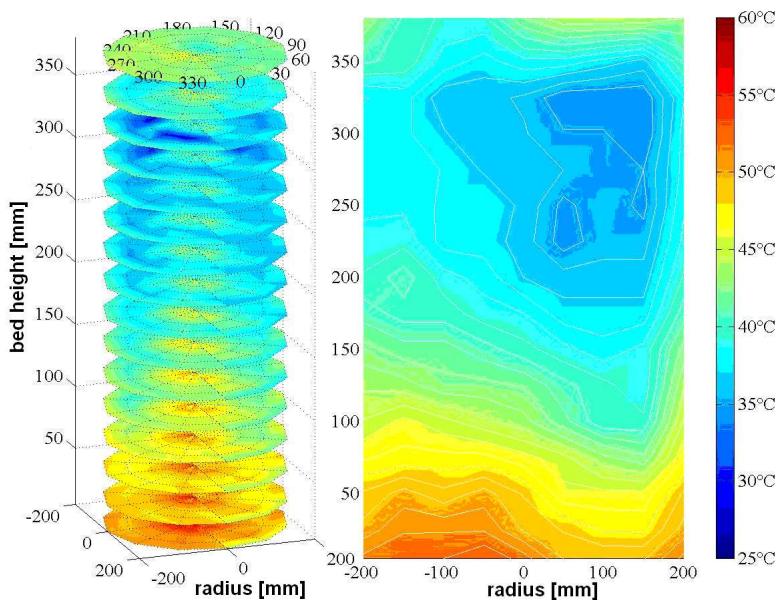


Figure 5.43: Experimental result of air temperature distribution of water sprayed fluidized bed with parameters as shown in Table A.2 (from Heinrich [47]).

mass $m_p = 370 \text{ kg}$ and mass flow of the nozzle $\dot{m}_L = 90 \text{ kg/h}$ are considered. The initial and boundary values are shown in Table A.3. This set of data is taken from the Ph. D thesis of Henneberg [51].

In this case we also observed the same behavior of balance variables in the simulation as explained for previous case. At the initial time the air humidity is 0.005 kg/kg . At the steady state solution it reaches 0.0225 kg/kg . In this simulation the air temperature decreases from 60°C to 26.4°C at the steady state solution, which can be seen in Figure 5.44. Also observed that, the same behavior for the degree of wetting, liquid film thickness and particle temperature as explained in the previous simulation result, see Figure 5.45.

Comparison with experimental results

The numerical results mentioned above are validated with experimental results. The experimental results are measured in a water sprayed fluidized bed plant. For more details regarding the experimental set up, we refer to Henneberg [51]. For the comparison, the experimental result of three dimensional distribution of air temperature over the bed height is depicted in Figure 5.46. Here, the experiments were carried out with consideration of the direct evaporation in the spray liquid. One can clearly see that, the temperature is lower

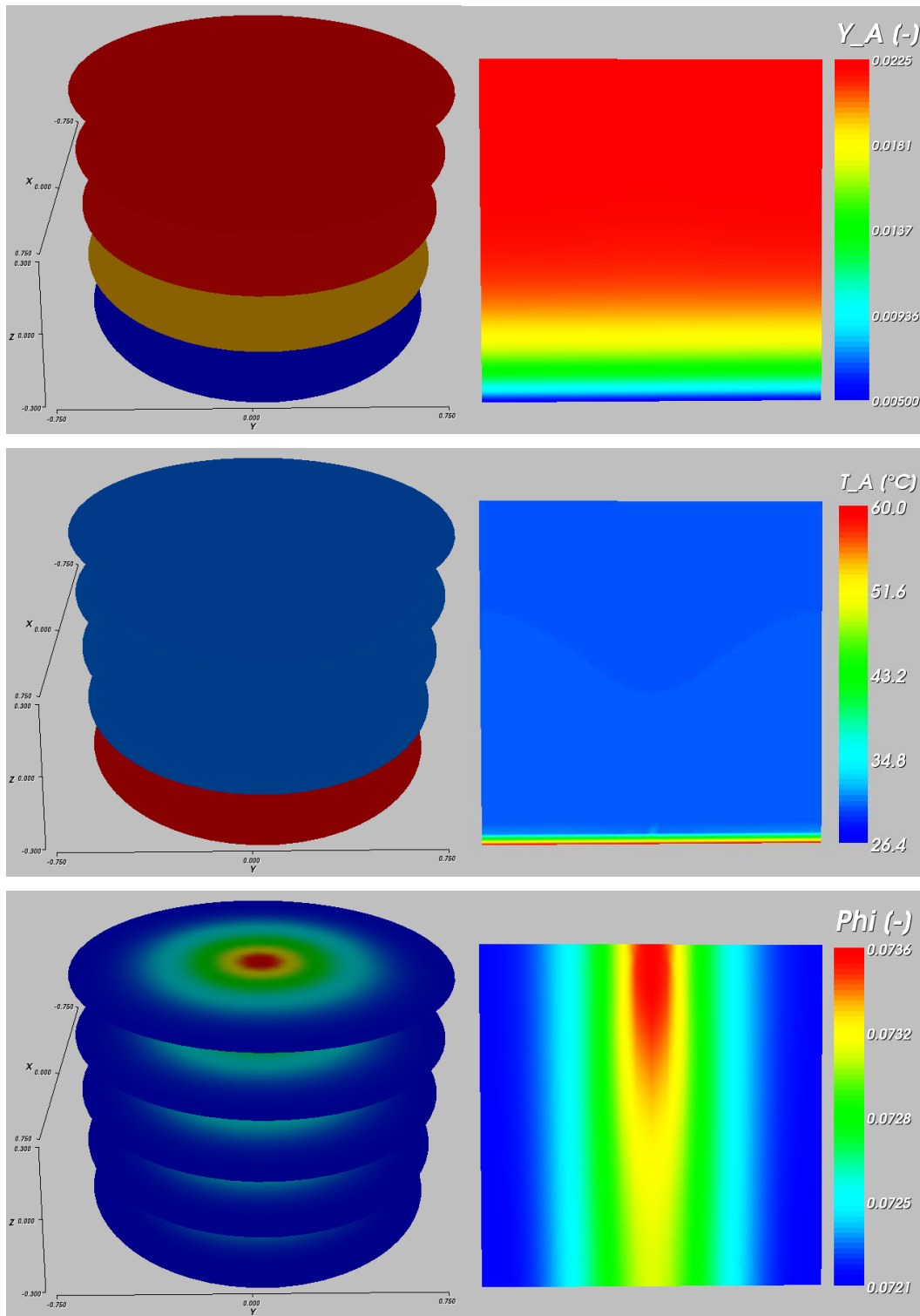


Figure 5.44: Simulation result of water sprayed fluidized bed at time $t = 600$ s with parameters as shown in Table A.3, **top**: air humidity, **middle**: air temperature, **bottom**: degree of wetting.

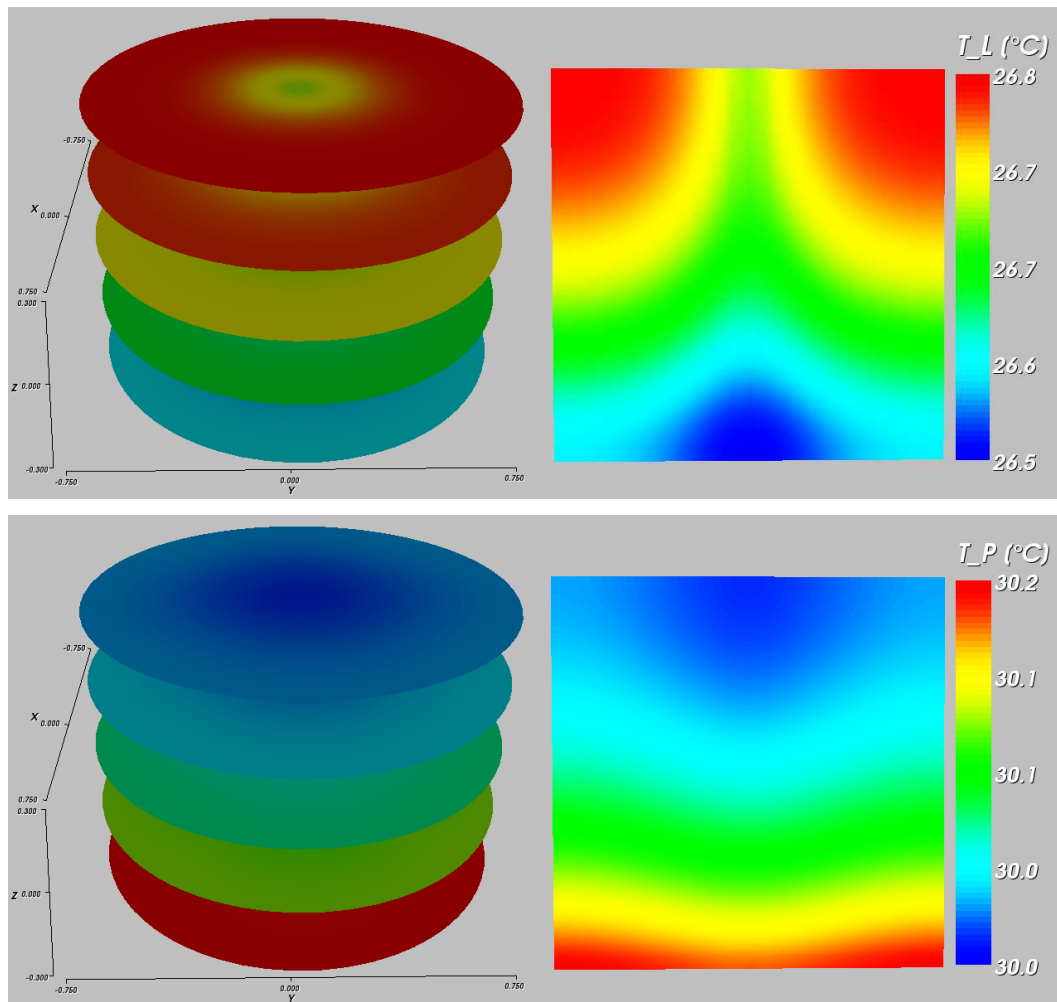


Figure 5.45: Steady state spatial profile of the simulation results of water sprayed fluidized bed at time $t = 600$ s with parameters as shown in Table A.3, **top**: liquid film temperature, **bottom** particle temperature.

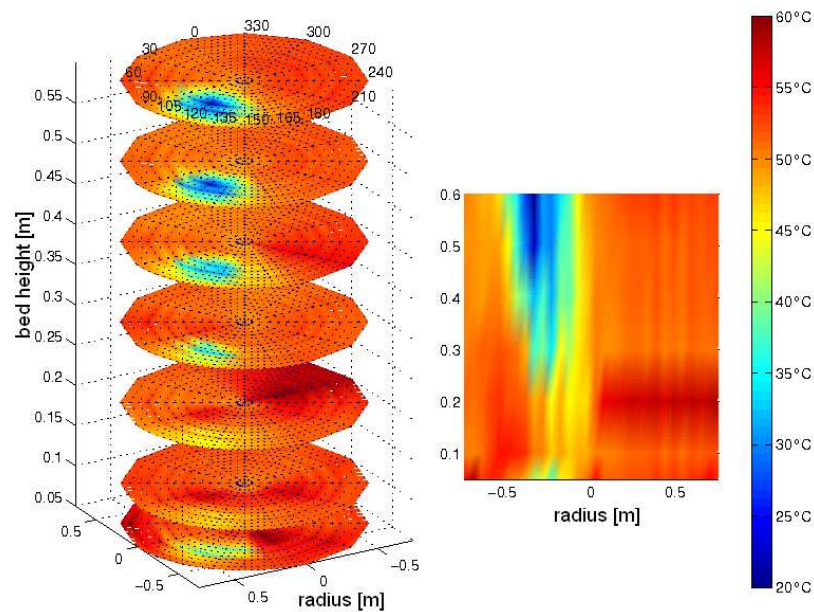


Figure 5.46: Experimental result of air temperature distribution of water sprayed fluidized bed with parameters as shown in Table A.3(from Henneberg [51]).

in the spray nozzle zone as compared to other regions. In the experimental results, the initial temperature is 60°C and at the steady state the minimum value reaches to 20°C . In numerical results, the initial temperature is 60°C and at the steady state the minimum value reaches to 26.4°C . The numerical results are simulated without considering direct evaporation. Despite this it can be observed that the numerical results are in good agreement with the experimental results qualitatively as described. The corresponding numerical results can be seen in the middle of Figure 5.40.

5.3 Parallel numerical results

In this section we presents the numerical results based on the domain decomposition methods, which are explained in Section 4.7. If we want to consider many nozzle net depositions then this computation involve thousands of unknowns in 3 dimensions. If we include many unknowns in 3D, in sequential computations it takes several days to obtain the results. To reduce this overload on sequential machines we prefer to use parallel architectures to obtain the faster results. In this section in all simulations we considered the spatial discretization based on the standard Galerkin method and time discretization by the backward Euler

method. For solving the system we used the semi implicit method. In this case we used the BiCGSTAB method as linear solver with the multigrid method as preconditioner. For the pre and post smoothing steps the ILU method with a damping factor $\omega = 0.95$ is used and the BiCGSTAB method is used as the base solver in the multigrid solver. These calculations were carried out on a HP-UX B.11.11 U 9000/800 machine with 2GB RAM for each processor that is connected to a 64 node cluster with 3GFOLPS processor speed at the Institute for Analysis and Numerics, Magdeburg.

In the parallel numerical simulation the most important part is the parallel grid manager and parallel implementation of the linear solver. For the domain decomposition we used the graph partitioning package CHACO [50]. Load balancing has been achieved as follows: the meshes of level-0 and level-1 have been kept on one processor and the mesh on level-2 as well as higher mesh levels have been distributed to all processors. For the domain decomposition we used the RCB (Recursive Coordinate Bisection) and RIB (Recursive Inertial Bisection) methods, as explained in Section 4.7.

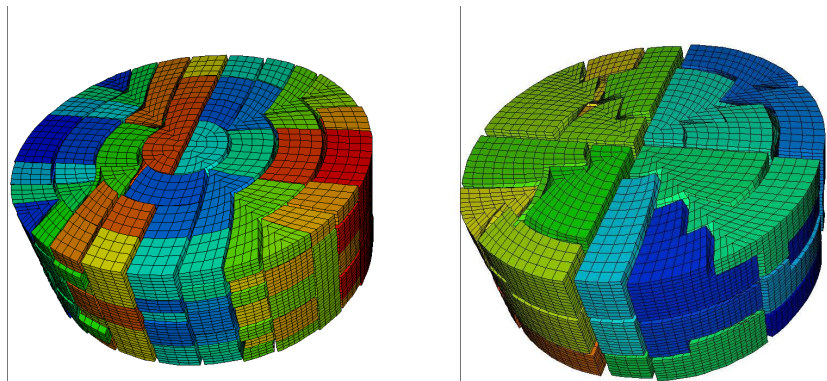


Figure 5.47: Domain decomposition of RCB and RIB algorithms using 48 processors

The Figure 5.47 shows the difference between the domain decomposition of RCB and RIB algorithms using 48 processors at mesh level 4. In the following tables the first column shows the number of processors used, the second column denotes the number of unknowns, the third column represents the number of time steps used in the simulation. Here we fixed the number of time steps and the time step size for all computations to show the efficiency between different processors. The fourth column shows the number of multigrid cycles performed in each time step, the sixth column shows the CPU time for one multigrid cycle, the seventh column shows the total CPU time for computation to require the number of time steps. The last column gives the efficiency of the computation

no. of procs	unknowns	time steps	mgs	1 mgs[s]	cpu time	efficiency
1	71,874	101	2	3.80	34m 58s	-
16	71,874	101	2.5	0.268	2m 47s	0.7852
32	71,874	101	2.49	0.143	1m 31s	0.7205
48	71,874	101	2.47	0.124	1m 14s	0.5907
58	71,874	101	2.45	0.1085	1m 5.5s	0.5525

Table 5.2: Efficiency test at mesh level 4 using RCB algorithm

no. of procs	unknowns	time steps	mgs	1 mgs[s]	cpu time	efficiency
1	71,874	101	2	3.80	34m 58s	-
16	71,874	101	3	0.2333	2m 45s	0.7947
32	71,874	101	3	0.1167	1m 30s	0.7285
48	71,874	101	3	0.08333	1m 5s	0.6724
58	71,874	101	3	0.0719	57s	0.6346

Table 5.3: Efficiency test at mesh level 4 using RIB algorithm

using several processors. This efficiency can be calculated using the formula $\frac{1}{p} \frac{T(1)}{T(p)}$, where $T(1)$ and $T(p)$ are total CPU time for 1 processor and p processors respectively. If we look at the Table 5.2, the efficiency reduces with increasing number of processors. Here we obtained poor performance using the RCB algorithm. We observed poor performance also using RIB algorithm. But we observed that using the RIB algorithm we obtained a better load balancing compared to the RCB algorithm, see Table 5.3 and Table 5.2. In the RCB algorithm the number of multigrid cycles are less as compared to the RIB algorithm but it takes more CPU time compared to the RIB algorithm.

The Figure 5.48 show the domain decomposition of the cylinder at mesh level 4, 5 and 6 using RIB algorithm with 58 processors. Mesh level 4 consists of 32,786 elements and mesh level 5 consists of 262,144 elements while mesh level 6 consists of 2,097,152 elements.

no. of procs	unknowns	time steps	mgs	1 mgs[s]	cpu time [s]	efficiency
1	549,250	11	3	31.80	2400	-
16	549,250	11	3.09	2.0930	173	0.8671
32	549,250	11	3.09	1.1300	91.43	0.8203
48	549,250	11	3.09	0.8451	62.38	0.8015
58	549,250	11	3.09	0.7310	52.44	0.7891

Table 5.4: Efficiency test at level 5 using RIB algorithm

no. of procs	unknowns	time steps	mgs	1 mgs[s]	cpu time
32	4,293,378	11	4	9.40	15m
48	4,293,378	11	3.363	7.196	9m 52s
58	4,293,378	11	3.181	6.263	8m 12s

Table 5.5: Efficiency test at level 6 using RIB algorithm

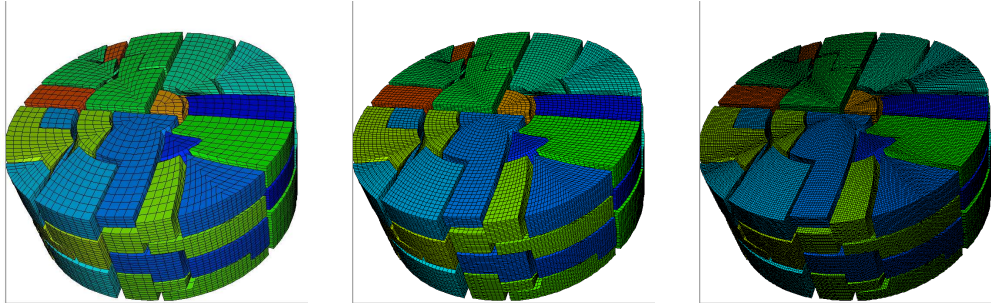


Figure 5.48: Domain decomposition of mesh with RIB algorithm using 48 processors at level 4, 5 and 6

The efficiency test for these mesh levels 4, 5 and 6 is shown in Table 5.3, Table 5.4 and Table 5.5 respectively. If we observe at mesh level 5 using 16 processors the efficiency is 0.8671 and with 58 processors 0.7891, while at mesh level 4 using 16 processors 0.7947 and with 58 processors is 0.6346. Here the efficiency is increased with increasing of unknowns.

Chapter 6

Numerical Results for the Intracellular Ca^{2+} Dynamics

This chapter is devoted to the numerical solution of intracellular Ca^{2+} dynamics, which was already explained in Chapter 3. This chapter is organized as follows: The first section describes the numerical results in two space dimensions and also explains different test cases based on the behavior of a channel opening and closing in clusters. In the first test case, we consider the opening of one channel in one cluster for a while deterministically and presents the convergence results using different mesh levels. Comparison of numerical results with the ROS3P method and the W-method are discussed. In the second test case, stochastic behavior for the channel transition in one cluster is considered and the numerical results of Ca^{2+} concentrations are presented with stochastic channel transition. In the third test case, the numerical results for opening of channels deterministically and stochastically in several clusters are presented. In the following section, numerical results in three space dimensions are presented and the convergence of solutions with opening of one channel in one cluster at different mesh levels are discussed. Finally, we present parallel numerical results using domain decomposition methods.

6.1 Numerical results in 2D

In the numerical simulations, different length scales of domain size for different numbers of cluster arrangements are considered. These length scales were explained Section 4.5 on the grid adaptivity. In these numerical simulations, free calcium in the cytosol, dye, mobile and stationary buffers in the cytosol are considered. In the simulations the ER calcium concentration is set to a constant value, say $700 \mu\text{M}$. In this section all numerical

simulations are performed on a rectangular geometry as computational domain and the refinement is considered by using triangular elements. The parameters used in the 2D numerical simulations are listed in Table B.1. In the numerical simulations, we use piecewise linear finite elements for the space discretization and linearly implicit Runge-Kutta methods for the time discretization. These were explained in Sections 4.3 and 4.4. For solving the resulting linear systems, the BiCGSTAB method [92] with SSOR preconditioner is used. All numerical computations were performed by using an Opteron Linux machine with 2GB RAM, 2GHz processor, gcc-3.3.5 compiler and the programm package UG [12].

6.1.1 Opening of one channel deterministically in one cluster

In this subsection, one cluster with 20 channels and a domain size of $[0,18000 \text{ nm}] \times [0,18000 \text{ nm}]$ are considered for numerical computations. For the initial triangulation a diameter of approximately 700 nm is considered for the triangles.

Tests of convergence with refinements

Here, the convergence results of one cluster which opens one channel deterministically for a while are presented. In this case the coarse mesh (level 0) consists of 4566 triangular elements and the fine mesh (level 8) consists of 10,602 elements, as was shown in Figure 4.3. An overview of the number of nodes and elements for the adaptive grid at different mesh levels at time $t = 0 \text{ s}$ is depicted in Table 6.1.

levels	nodes	elements	min area of elements (nm^2)	nodes in cluster area
0	2,378	4,566	89810.0	-
8	5,396	10,602	3.78471	135
9	5,817	11,444	0.946177	534
10	6,235	12,280	0.236544	952
11	6,664	13,138	0.0591361	1,381

Table 6.1: The number of nodes and elements at different adaptive mesh levels.

First let us consider the numerical simulation of one channel is opened for a while. The numerical results are tested with different mesh levels. The average value of cytosol calcium concentration at different grid levels is shown in Figure 6.1. The average value of the solution is calculated by using the following formula

$$\bar{f} = \frac{1}{|\Omega|} \int_{\Omega} f(x) dx, \quad (6.1)$$

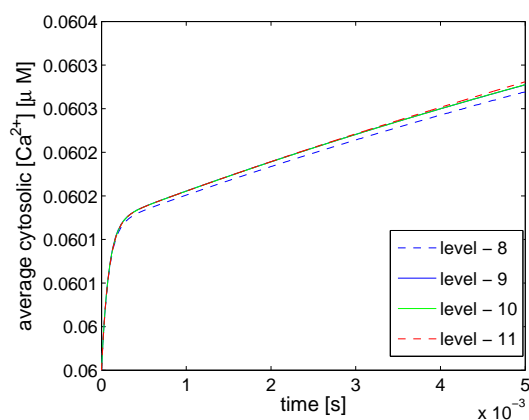


Figure 6.1: The average cytosolic Ca^{2+} concentration at different mesh levels.

where $f(x)$ is the numerical solution over the domain. From Figure 6.1 it is observed that the numerical results are converging with finer meshes. It is also observed that the number of elements are not enough to converge to the solution at level 8. Therefore, level 9 is considered as the minimum level for convergence of the solutions of this problem.

The maximum cytosolic Ca^{2+} concentration for reaching the stationary solution versus time at mesh level 9 is shown on left hand side Figure 6.2. Here it is observed that the maximum concentration reaches the stationary solution in the order of milliseconds. The right hand side Figure 6.2 shows the average cytosolic Ca^{2+} concentration at mesh level 9 and one can observe that the average value reaches the stationary solution in the order of seconds.

In left hand side of Figure 6.3, the maximum cytosolic Ca^{2+} concentration, which occurs at the point (16504.4, 16498.4), over the simulation time for different levels is presented. It can be observed that the solution converges for finer meshes. One can also see in right hand side of Figure 6.3 the convergence of the maximum Ca^{2+} concentration at the stationary solution for different nodal points.

The left hand side Figure 6.4 shows for a channel opening the spatial profile of free cytosolic Ca^{2+} at different mesh levels over the distance of length $1 \mu\text{m}$ from both sides of the cluster origin in the steady state solution. Here the numerical convergence of the solutions is tested at different mesh levels, i.e. increasing the number of mesh points in the area of the channel. We observe that they are in good agreement between all levels. At the peak point it is observed that the curve at mesh level 8 is lower than the curves of other mesh levels. The zoom of the peak point is depicted in right hand side Figure 6.4 to

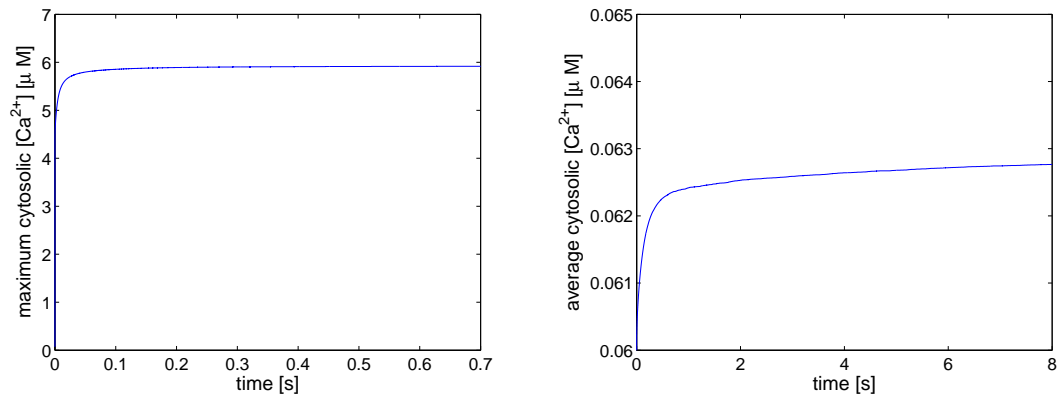


Figure 6.2: The solution of cytosolic Ca^{2+} concentration versus time at mesh level 9; **left**: maximum, **right**: average cytosolic Ca^{2+} concentration.

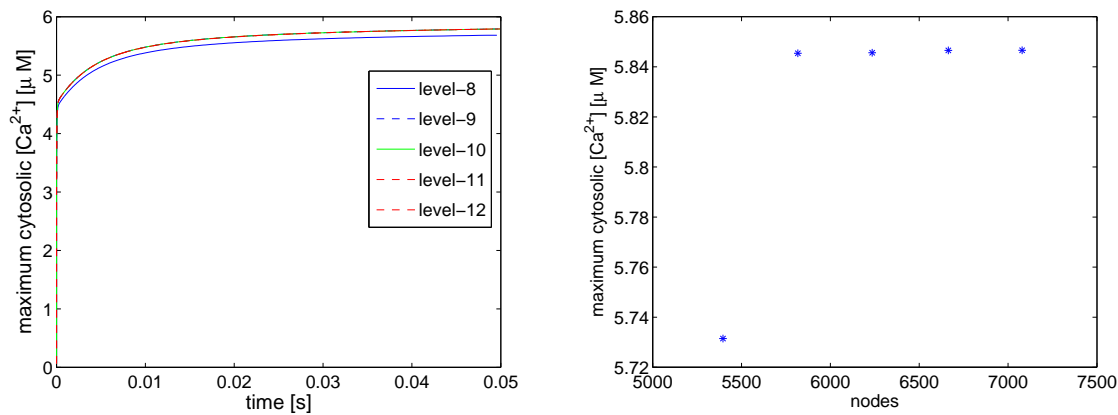


Figure 6.3: **left**: Solution of cytosolic Ca^{2+} concentration at point (16504.4, 16498.4) (close to the point where $[\text{Ca}^{2+}]$ reaches a maximum value) versus time at different levels; **right**: maximum concentration for the stationary state at different numbers of nodal points.

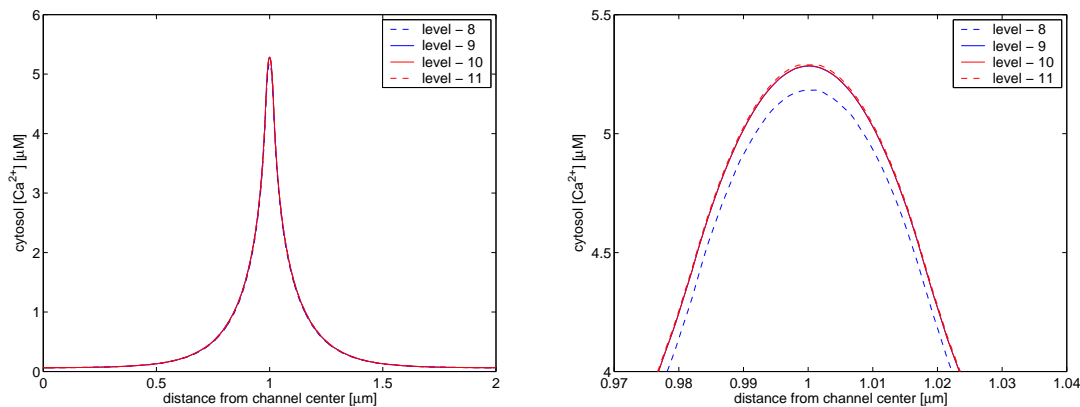


Figure 6.4: **left:** Spatial profile of free cytosolic Ca^{2+} at different mesh levels of channel opening over the distance of length $1 \mu\text{m}$ from the channel center at stationary solution, **right:** zoom at the peak point.

see the difference clearly. As explained in the previous case, the number of mesh points is not enough when the channel stays opened for a while. The obvious remedy is spatial grid adaption during the intermediate time steps.

Study of time step control at opening and closing events

Here we explain the necessity of adaptive time step control at channel transitions. The surface plots of the cytosol Ca^{2+} concentration are depicted in Figure 6.5 for one open channel at different times. A very high transient solution at the middle of the cluster and a smooth solution outside of cluster have been observed. This high concentration restricts us to take small time steps in the numerical computations. Next, the spatial profiles of free cytosolic Ca^{2+} with a distance of $1 \mu\text{M}$ from both sides of the channel center at different time levels are depicted in left hand side of Figure 6.6.

It can be observed that, the calcium concentration in the vicinity of a few nano meters from the channel center rises within microseconds upon opening of a channel. Here, the channel is opened until time 0.001 s and then closes afterwards. In this case one needs a efficient time stepping method and a good adaptive time step control to follow these rapid changings in time. For this purpose linearly implicit Runge-Kutta methods for time integration are a suitable choice. Adaptive time step control theory has been well established for these methods, see Hairer and Wanner [45], Lang and Verwer [58] and Schmitt and Weiner [82]. The right hand side of Figure 6.6 shows the spatial profile of the closing channel over time levels where the opened channel is closed at certain time.

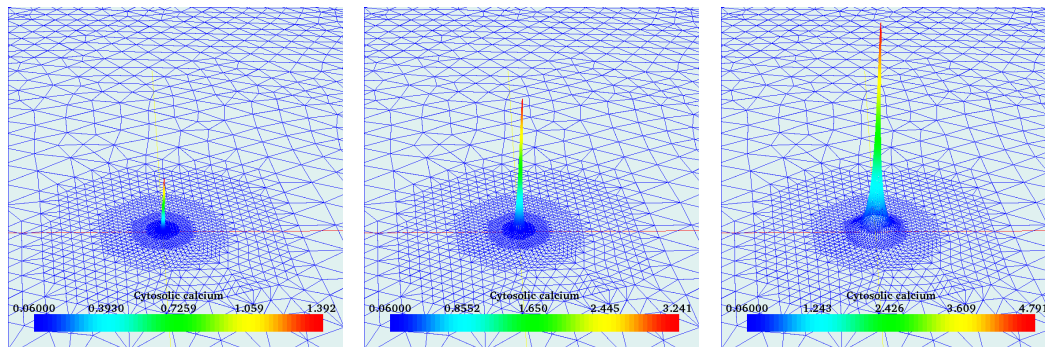


Figure 6.5: The surface plot of the cytosolic calcium concentration for a opening of channel at times $1.0 \cdot 10^{-06}$, $1.1 \cdot 10^{-05}$ and $9.970 \cdot 10^{-04}$.

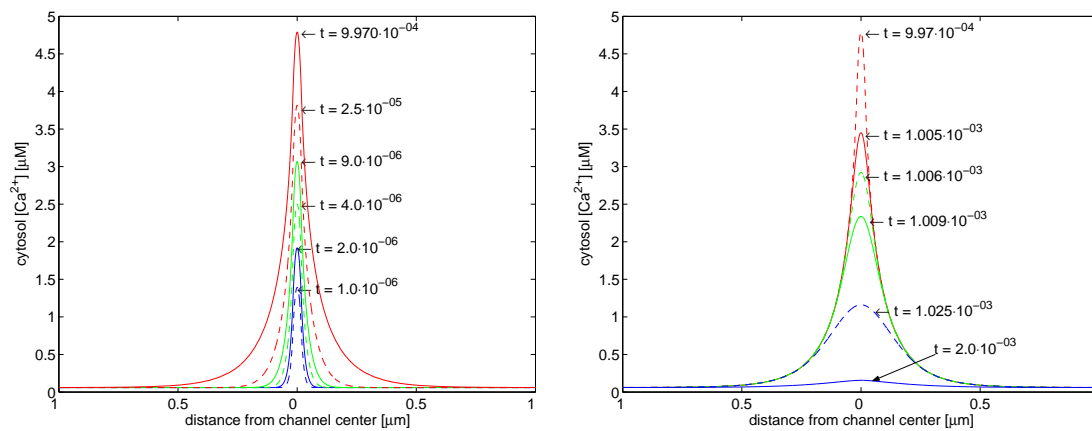


Figure 6.6: Spatial profile of the free cytosolic Ca^{2+} at different time levels (sec) of channel opening at the **left** and closing at the **right**.

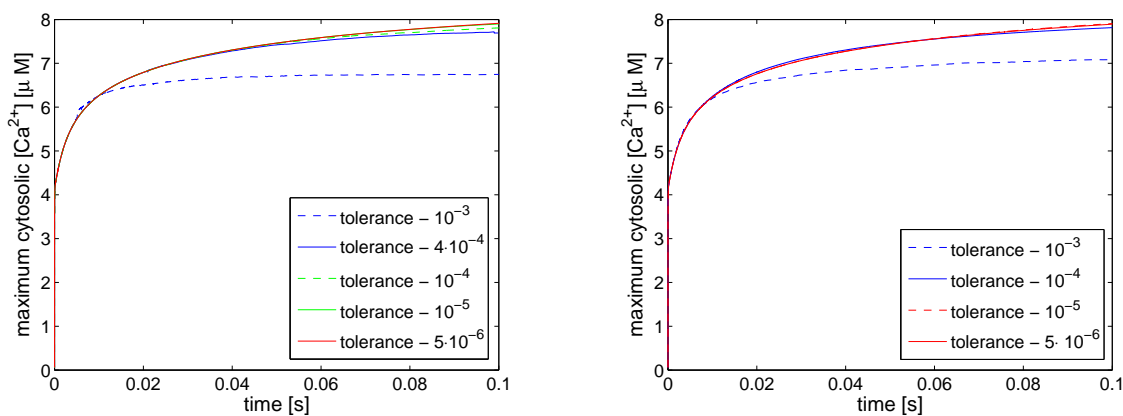


Figure 6.7: Maximum cytosolic Ca^{2+} concentration versus time until $t = 0.1\text{s}$ at different tolerances; **left**: with ROS3P method, **right**: with W-method

Comparison of ROS3P and W-methods

Here we show the numerical results based on ROS3P and W-methods. The left hand side Figure 6.7 represents the comparison of the solutions by using the ROS3P method and the W-method. For this simulation the initial time step 10^{-9} and different tolerances are considered. Furthermore, the ER Ca^{2+} concentration is considered instead of the dye buffer for these computations. In the simulations, a tolerance for the linear solver based on the time step tolerance is considered, say $TOL = TOL_t/\tau$ where TOL is the tolerance for linear solver, TOL_t is the tolerance for time step control and τ is the time step size. The final simulation time is $t = 0.1\text{ s}$ and all the computations are performed at mesh level 9. The convergence is presented of solutions at different tolerances with the ROS3P method, in left hand side of Figure 6.7 and with the W-method in right hand side of Figure 6.7. One can observe that $4 \cdot 10^{-4}$ and 10^{-4} are the maximum tolerances for the ROS3P method and the W-method respectively to give numerically convergent solutions. At these tolerances it is observed that the solutions of both methods converge efficiently. The W-method uses large time steps as compared to the ROS3P method. One can see the number of rejected time steps versus time in the left hand side of Figure 6.8 and the number of accepted time steps versus time in right hand side of Figure 6.8 for both methods. We give in Table 6.2 the number of accepted and rejected time steps as well as the CPU time.

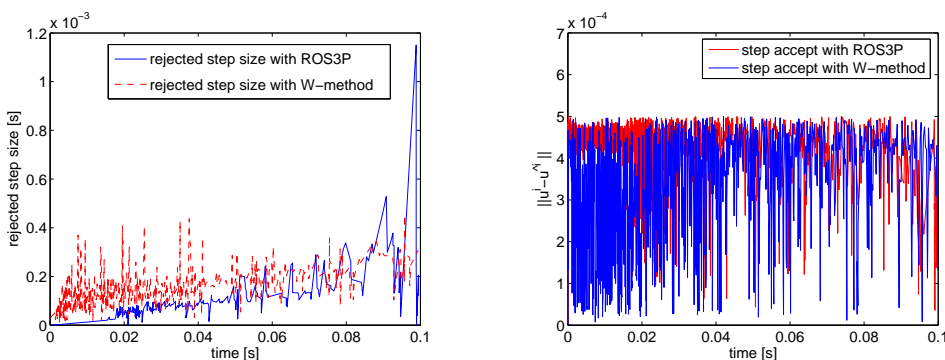


Figure 6.8: Comparison between the ROS3P method and W-method; **left**: rejected step size, **right**: error between two solutions when step accepted.

	ROS3P method	W-method
unknowns	21,368	21,368
no. of steps accepted	5883	898
no. of steps rejected	343	592
CPU time	2h 27m 32s	1h 27m 49s

Table 6.2: Comparison between the ROS3P method and W-method where the simulation time is $t = 0.1$ s and one channel opened.

Coupling of step size control and mesh adaption

Generally, the channel opening occurs in the order of microseconds. It stays open for some order of milliseconds while calcium diffuses slowly to neighboring channels. When it diffuses to the neighboring channels that region also needs a finer mesh. In this case spatial grid refinement is necessary during the intermediate time steps to reduce the overall computational cost. In this study we considered two types of adaptive grid refinement procedures. In the first type of procedure, the spatial error estimator is called whenever a time step is rejected. In the second type of procedure, this error estimator is called for all time steps.

Let us consider the first type of procedure. Here, one important point to notice is that during the simulations the original structure of the grid is fixed. It is created before the simulation starts. Moreover, we allow only refinement when the time-step error estimator is called for the first time during the intermediate time steps. Then coarsening is allowed only on refined elements which were created after the simulation started. We consider an initial level 9 grid for our simulation and keep it fixed. In this computation the simulation

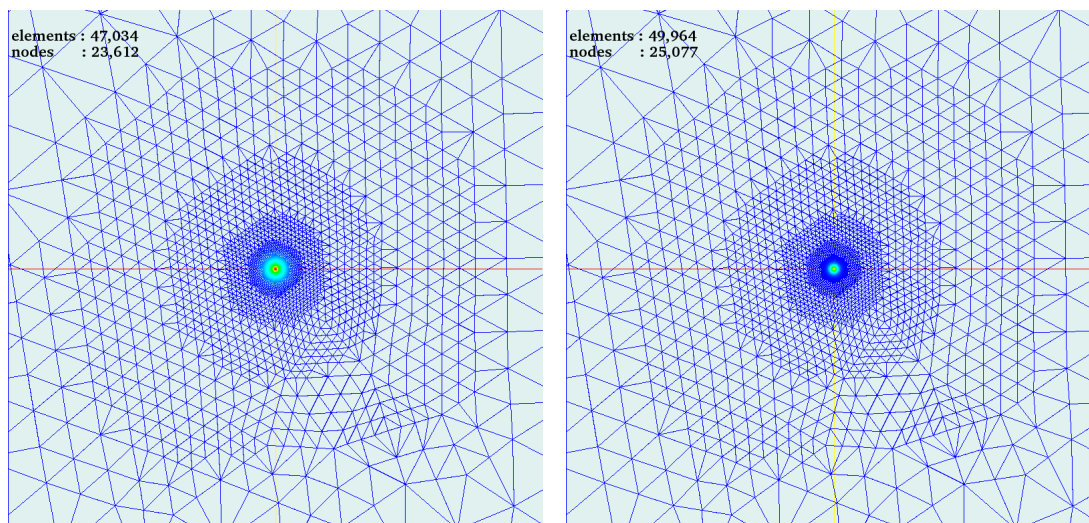


Figure 6.9: Local zoom of the spatial grids of the adaptive mesh at time $t = 0.002$ s; **left**: where the refinement is done with rejected time steps, **right** where the refinement is done with every time step.

time is set to 0.002 s. Initially, the number of nodes and elements of this mesh at level 9 are 5,817 and 11,444. However, at the end of the simulation, the number of nodes and elements of the final mesh at level 15 have reached to 25,077 and 49,964 respectively. Here the level 9 grid is always a subgrid of the level 15 grid. The local zoom of the level 15 grid structure at time $t = 0.002$ s is shown in the left hand side of Figure 6.9.

Next, we considered the second type of adaptive grid refinement procedure. In this case, initially the level 9 grid again has 5817 nodes and 11,444 elements and at the end of the simulation, the number of nodes and elements reached to 23,612 and 47,034 respectively. The zoom of final adaptive mesh at level 15 at time $t = 0.002$ s is shown in the right hand side of Figure 6.9. The results with both refinement processes are presented in Figure 6.10. For this calculation, the tolerance for step size control was $5 \cdot 10^{-5}$ while the tolerance for the spatial error was $3 \cdot 10^{-3}$. The initial time step started with $1 \cdot 10^{-8}$. From the results, in the top two plots it is observed that good convergence of the solutions is achieved by using adaptive grid refinement during the intermediate time steps with both procedures for this problem. In the middle two plots of Figure 6.10 we observe that initially many elements are refined with the second type of refinement procedure as compared to the first type of procedure. We can see clearly the difference between spatial error estimator in the bottom plots of Figure 6.10. It is also observed that the spatial refinement process restricted to

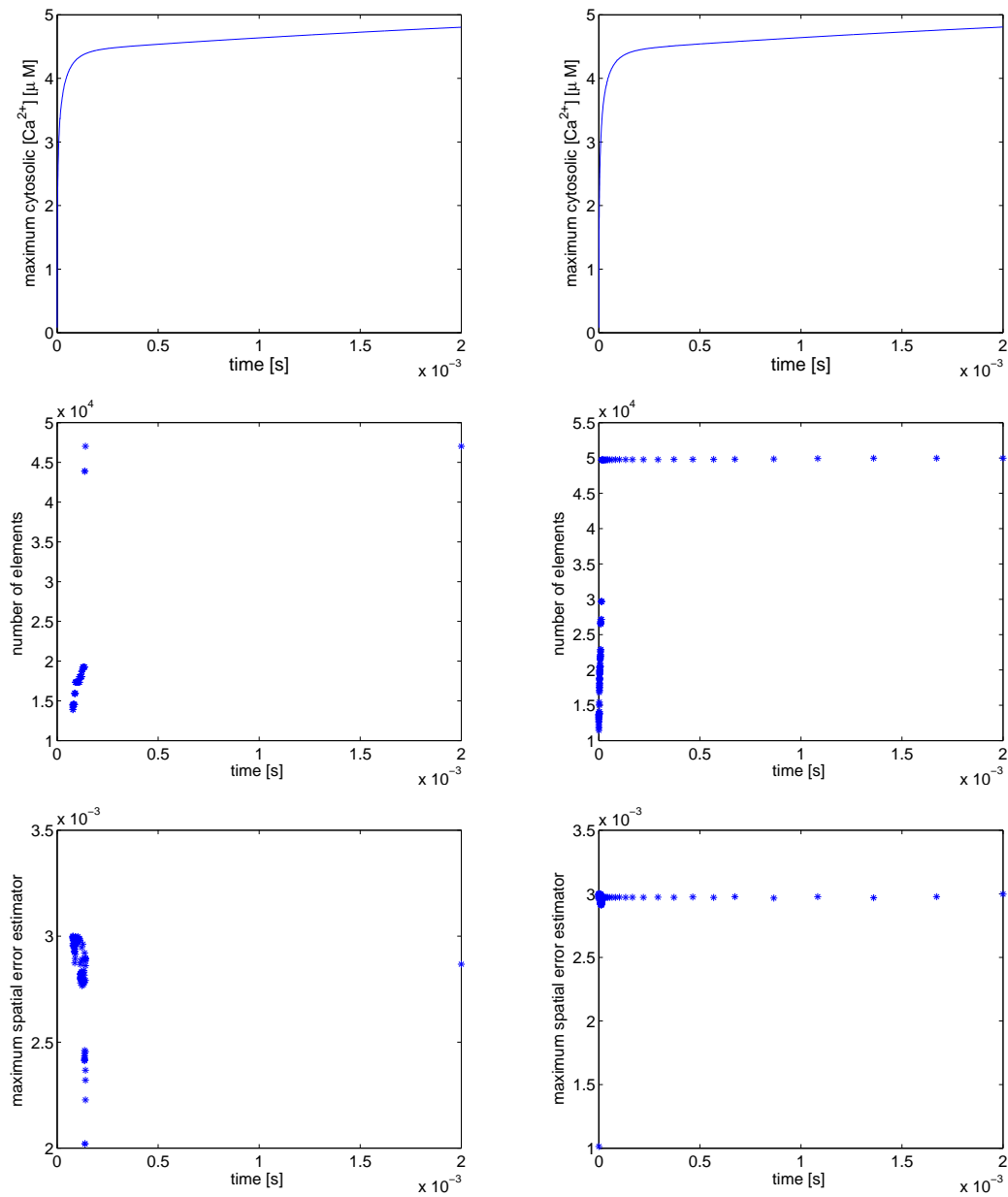


Figure 6.10: Comparison between two refinement processes; **left**: results where the refinement is considered whenever time step rejected, **right**: where the refinement is considered at each time step.

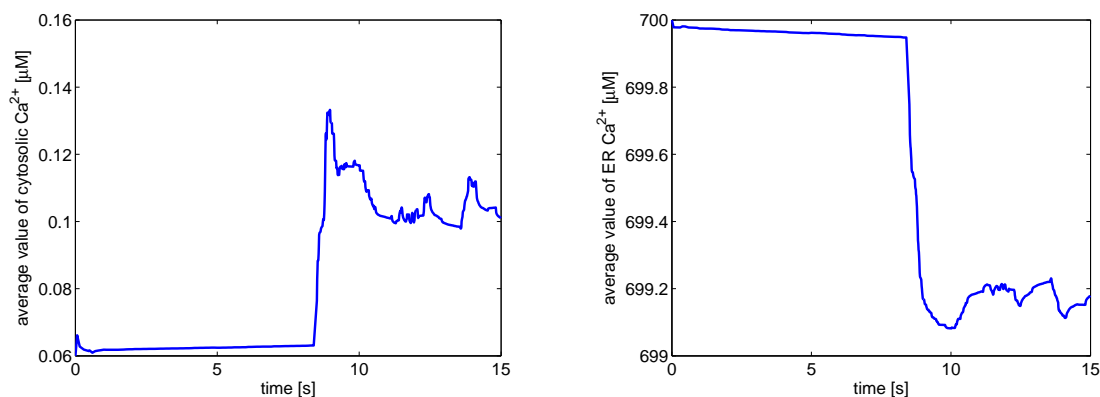


Figure 6.11: The average solution of cytosolic calcium concentration at **left** and ER calcium concentration at **right** for opening of channels stochastically in a one cluster over a time.

rejected time steps takes less CPU time as compared to the refinement at every time step. The CPU times are 34m 58s and 54m 38s, respectively to compute the solution until the simulation time $t = 0.002$ s, also see Nagaiah et al. [68] for some more results.

6.1.2 Numerical results of the stochastic channel transition in one cluster

In this subsection, the numerical solutions of calcium concentrations with stochastic channel transition in one cluster are presented. In the simulations free Ca^{2+} in the cytosol, free Ca^{2+} in the ER, the mobile and the stationary buffers are considered. In our numerical simulations deterministic and stochastic equations are coupled and require two different time steps. For solving the deterministic equations the linearly implicit methods have been used, which are suitable to use adaptive time step control. The stochastic solver is based on the Gillespie method, which is explained in Section 3.3. The Gillespie method also uses adaptive time steps, in the sense that its time step follows the evolution of transition probability. As explained in Section 3.3, the usual Gillespie method has one drawback, since this method solves stochastic processes where the propensities are constant during the subsequent transitions. However, the propensities of the problem considered in this work depend on the calcium concentrations which may change rapidly due to channel opening and closing. For this purpose we adopted the hybrid algorithm which was proposed by Alfonsi et al. [2]. This hybrid algorithm solves the coupled deterministic and stochastic equations where the propensities depend on the channel opening and closing. The step by step process of the hybrid algorithm can be found as Algorithm 3 in Appendix B.

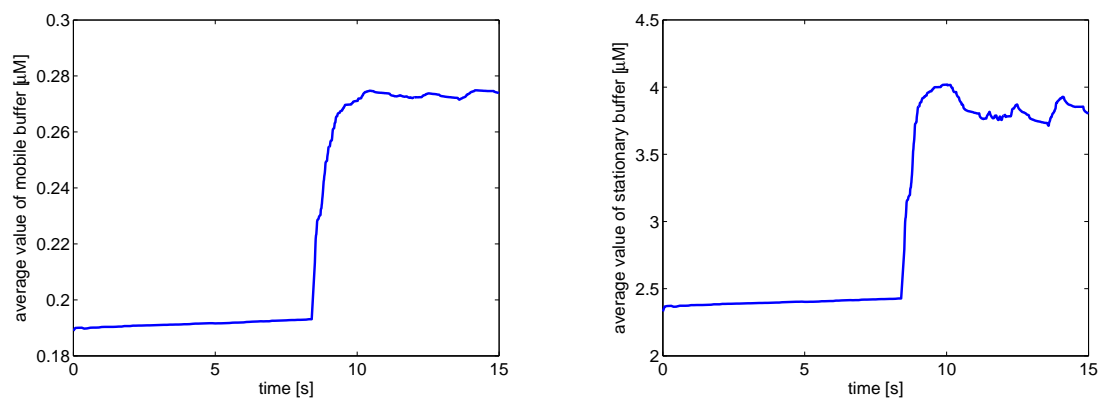


Figure 6.12: The average solution of cytosolic mobile buffer concentration at left and stationary buffer concentration at right for opening of channels stochastically in a one cluster over a time.

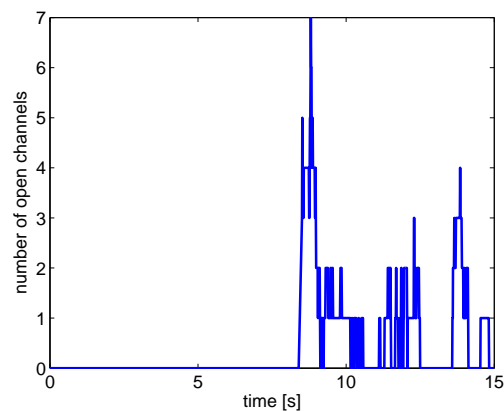


Figure 6.13: The number of open channels stochastically in one cluster versus the time.

First, let us consider the simulation of one cluster which consists of 20 channels. The concentration changes occur rapidly when a channel opens and closes. It stays constant when all channels are closed, as is shown in Figures 6.11 and 6.12. The left hand side Figure 6.11 shows that the Ca^{2+} concentration is constant until time $t = 8.4$ s because no channel is opened during this time and after this time the Ca^{2+} concentration changes due to channels opening and closings.

In Figure 6.13, where the number of open channels versus time is plotted. Typically, the time step reduces to 10^{-8} s during the channel opening and it returns to 10^{-3} s when all channels are closed. During this fast changing the adaptive time step control plays an important role to maintain the accuracy of the solution.

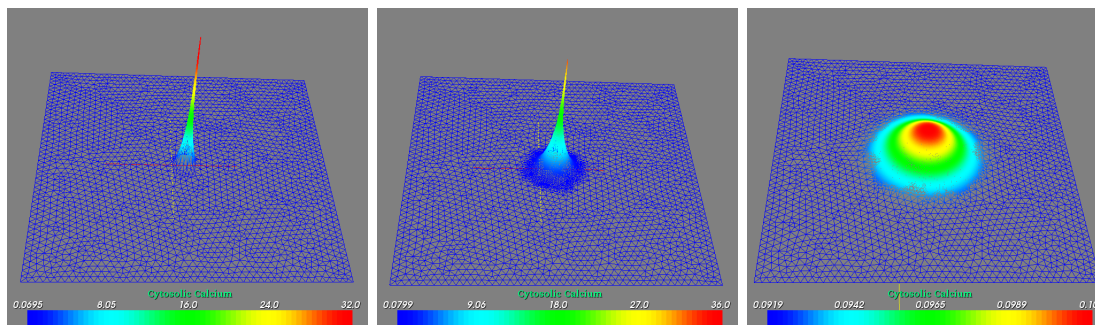


Figure 6.14: The spatial profile of the cytosolic calcium concentration at time $t = 6.504$ s, 6.68 s, 8.92 s in one cluster arrangement at mesh level 7.

Next, grid adaptivity is incorporated during the intermediate time steps at different mesh levels. Here the finest adaptive mesh is fixed which is created at the initial time. The basic idea for keeping initial adaptive grid fixed is to maintain the original structure of the clusters or to maintain the minimum 4 grid points lie in the area of each channel during the intermediate time steps. Then we apply the spatial grid adaptivity during the time steps. The refined and coarsened mesh grids are always subgrid of initial mesh grid. Due to our previous studies, the grid adaptivity is based on the time step rejections. When the time step step is rejected the Z^2 error estimator is called for possible refinement and coarsening of the spatial grid.

Lets say we have incorporated grid adaptivity during the intermediate time steps at mesh level 7. Here we intended to have less mesh points as compared to the previous test cases. It will refine the elements during the simulation where the solution is less regular. Here the channel opening is considered in the stochastic case. Initially mesh level 7 contains 2737 nodes and 5284 elements. At time $t = 6.504$ s it contains 3216 nodes and 6242 elements when one channel is opened and at time $t = 8.92$ s it reaches to 18493 nodes and 36796 elements when all channels are closed, see Figure 6.14. From the above results one can see the importance of the grid adaptivity in the stochastic regime.

6.1.3 Numerical results with many clusters

Results with deterministic opening of channels

In this subsection we considered the arrangement of 16 and 100 clusters for numerical simulation. In these numerical simulations free calcium in the cytosol, free calcium in the ER, the mobile and the stationary buffers in the cytosol are computed. To generate the grid, initially we took a uniform grid refinement and then applied the Z^2 error indicator

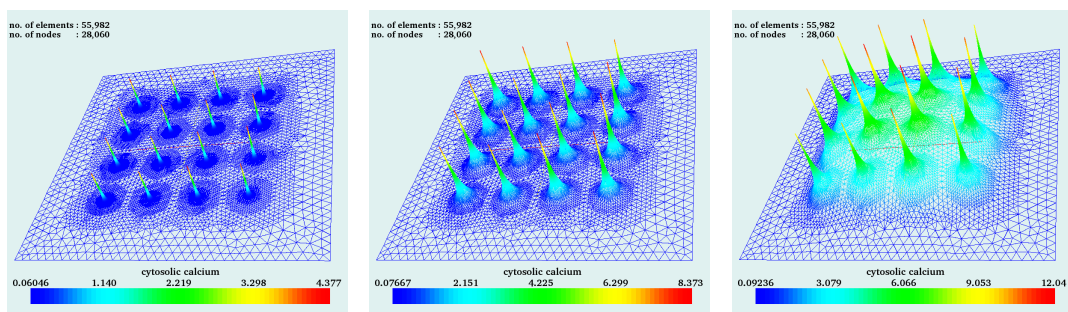


Figure 6.15: Surface plot of the cytosolic calcium concentration for opening of one channel in each cluster of a 16 cluster arrangement at times $9.0 \cdot 10^{-05} s$, $0.1 s$ and $2 s$.

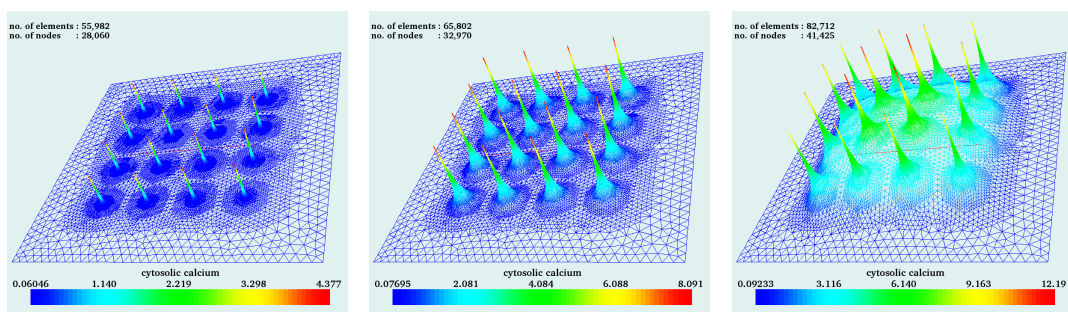


Figure 6.16: Surface plot of the cytosolic calcium concentration for opening of one channel in each cluster of a 16 cluster arrangement using temporal adaptive grid refinement at times $9.0 \cdot 10^{-05} s$, $0.1 s$ and $2 s$.

using the strongly localized initial function to get a fine mesh in the area of channels and clusters, see Figure 4.4 and Figure 4.5 for 16 and 100 cluster arrangements respectively. The simulations start with constant initial values for calcium concentrations and buffers over the domain.

First, in the numerical simulation let us consider the opening of channels in a deterministic way i.e. one channel is opened in each cluster for a while. The plots of the cytosolic Ca^{2+} concentration for 16 clusters at times $9.0 \cdot 10^{-05} s$, $0.1 s$ and $2 s$ are shown in Figure 6.15. The cytosolic Ca^{2+} concentration for 16 clusters with grid adaption during the intermediate time steps is shown in Figure 6.16. In both cases the simulations start with 55,982 elements and 28,060 nodes. In the case of grid adaptivity during the intermediate time steps, at stationary solution, lets say time $t = 2 s$, the mesh consists of 41,425 nodes and 82,712 elements. Here the maximum cytosolic Ca^{2+} concentrations are $12.04 \mu\text{M}$ and $12.19 \mu\text{M}$ with fixed adaptive grid and temporal adaptive grid during the intermediate

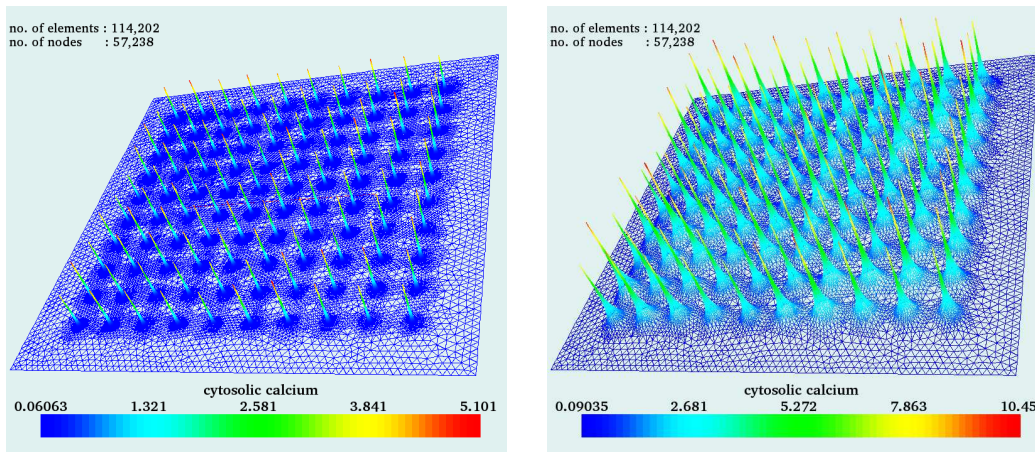


Figure 6.17: The surface plot of the cytosolic calcium concentration for opening of one channel in each cluster of 100 cluster arrangement at times $9.4 \cdot 10^{-05}$ s and 0.4 s.

time steps respectively.

Likewise, the surface plots of the cytosolic Ca^{2+} concentration for opening of one channel in 100 cluster arrangement at different time steps are depicted in Figure 6.17. Here the most important issue is that to get an accurate and efficient solution at higher numbers of cluster arrangements. In these cases one has to consider many mesh points to get accurate solution. Therefore, the computational cost increases rapidly with the increase of cluster arrangements.

Results with stochastic opening of channels

In fact, the opening and closing of channels follows a random process, as explained in the previous subsection. This stochastic process leads to many challenging mathematical and computational problems with higher numbers of cluster arrangements.

First let us consider the arrangement of 16 clusters with stochastic channel transitions. In this arrangement the initial adaptive grid is fixed during the intermediate time steps. It is created before the simulation starts. The number of open clusters and channels versus the simulation time until 15.3 s are depicted in Figure 6.19. The corresponding average value of the cytosolic calcium concentration and the ER calcium concentration are presented in the Figure 6.18. In this simulation the time steps typically vary between 10^{-3} s and 10^{-8} s. We observe is that when opening of one channel occurs in any cluster then the scheme rejects some steps and the step size goes down to 10^{-8} s. When all channels are closed in all clusters then the time step returns to the order of micro seconds.

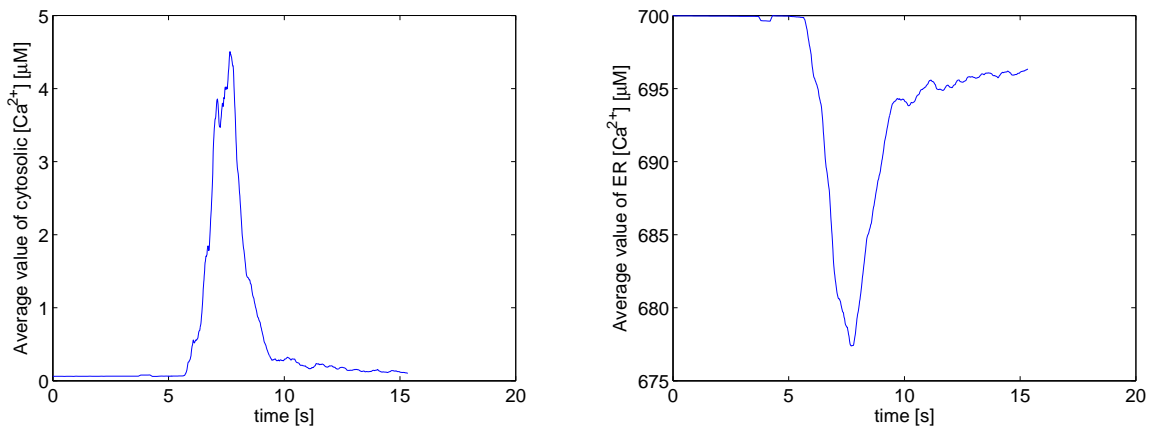


Figure 6.18: The average value of cytosolic Ca^{2+} concentration and ER Ca^{2+} concentration plotted over the time.

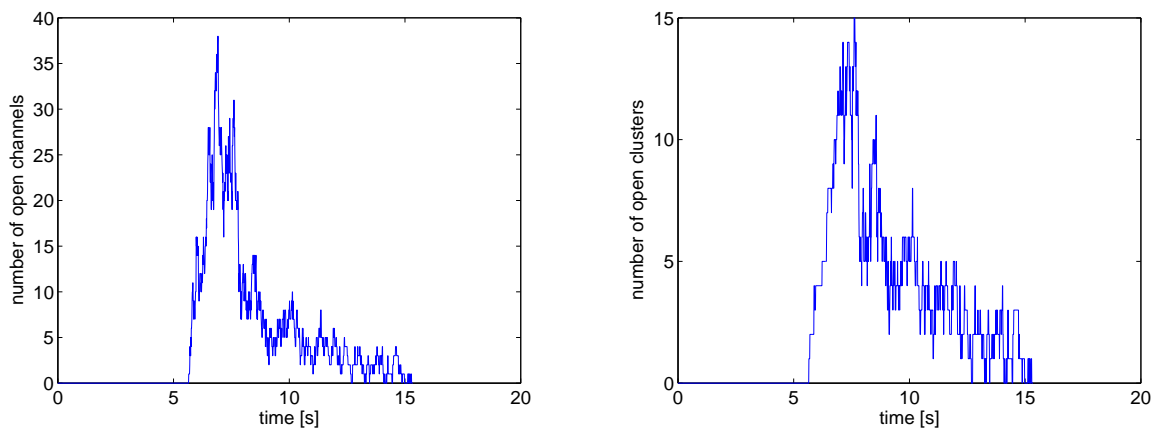


Figure 6.19: The number of open clusters at the **left**: and channels at the **right**: are depicted in the arrangement of 16 clusters over the time.

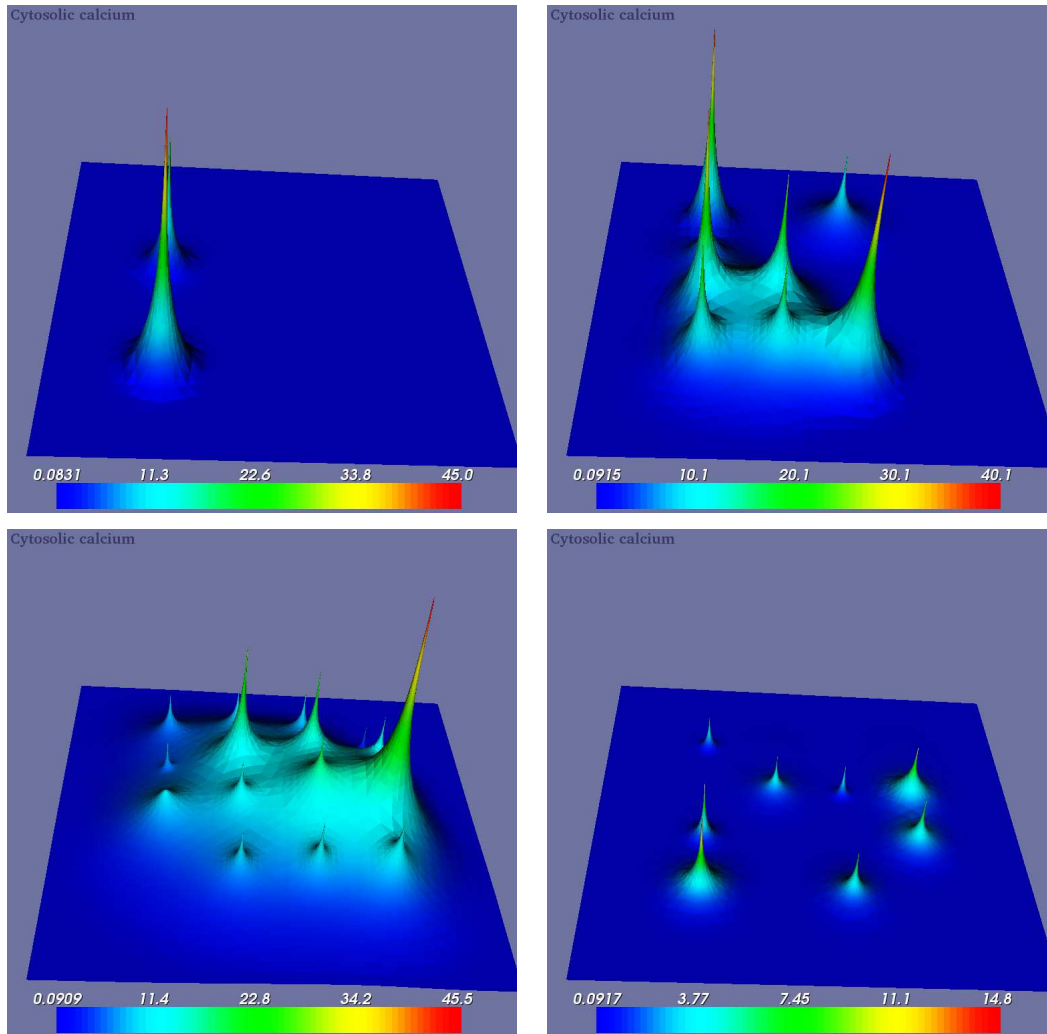


Figure 6.20: Solution of cytosolic concentration of 16 clusters with stochastic channel transition of channels at different time steps $t = 5.509870$ s, 6.239932 s, 6.999976 s and 9.809869 s.

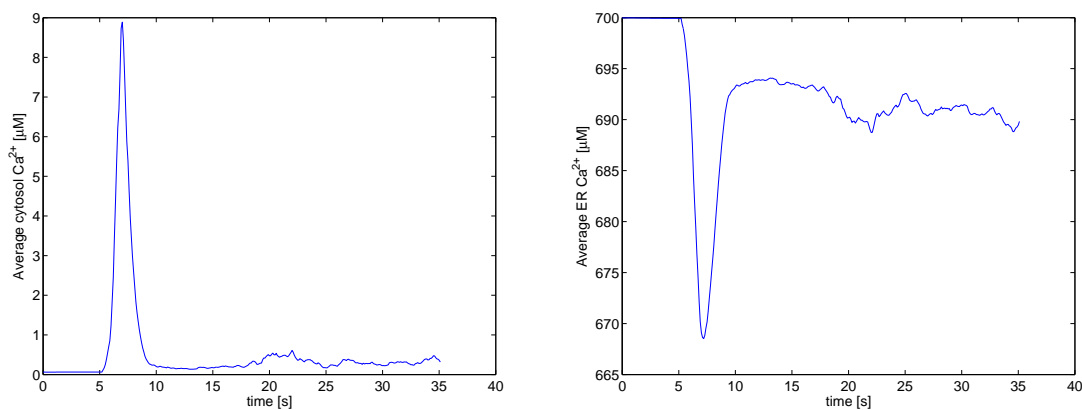


Figure 6.21: The average cytosol free calcium at **left**: and ER free calcium at **right**: over the time in 100 cluster arrangement.

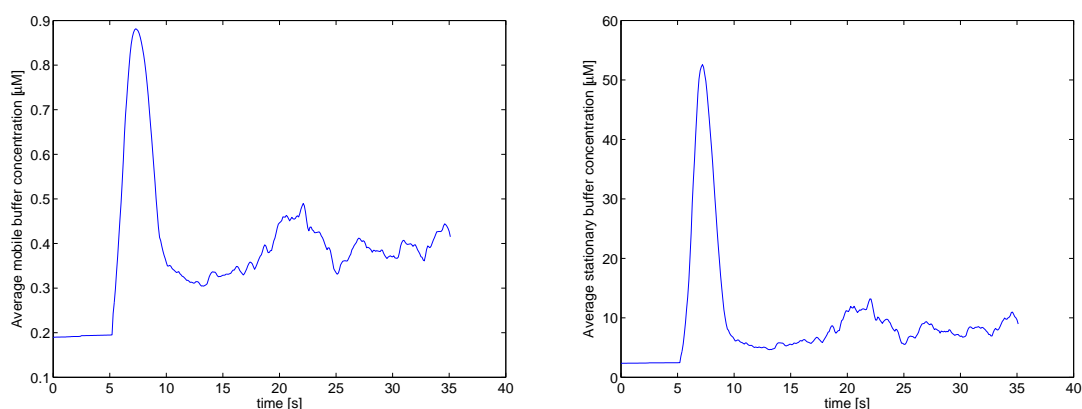


Figure 6.22: The average solution of mobile and stationary concentrations in cytosol over the time in 100 cluster arrangement.

The surface plot of the cytosolic calcium concentration is depicted at different times in Figure 6.20. Here one can see clearly the opening and closing of channels stochastically in many clusters.

Now let us consider the 100 cluster arrangement for numerical simulations. The average solution of the cytosol free calcium and the ER free calcium concentrations over the time is plotted in Figure 6.21. Also can see the mobile and the stationary buffer concentrations in the Figure 6.22. Here we can observe that around the time $t = 6.5$ s many clusters are open and these open clusters form a wave.

The corresponding number of open channels and open clusters can be seen clearly in Figure 6.23. The first opening of the channel event occurs approximately after time

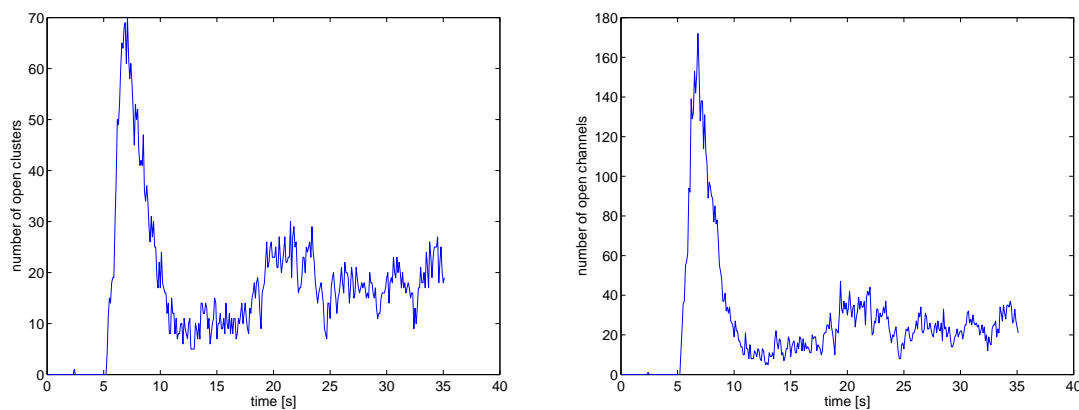


Figure 6.23: The number of open clusters and open channels in 100 cluster arrangement; **left**: open clusters, **right**: open channels.

$t = 5.3$ s. This leads to the increased binding probability at the activating sites. Thus many channels are open in many clusters during the time approximately 6.5 s. Activation is a fast process provided that the Ca^{2+} concentration is high. The average Ca^{2+} concentration is very high at this time, see clearly in Figure 6.21.

The time dependent behavior of the cytosolic Ca^{2+} concentration is depicted in Figures 6.24 and 6.25. Here, one can clearly see the spatial profile of the Ca^{2+} concentration at different time steps.

6.2 Numerical results in 3D

This section is concerned with the 3D numerical results of deterministic and the hybrid solutions. For the numerical simulation, the Ca^{2+} concentration, the dye, the mobile and the stationary buffers in the cytosol are considered as variables and the ER Ca^{2+} concentration is set to be constant during the simulation. The system of reaction-diffusion equations (3.7–3.10) has been solved in cube with Robin type boundary condition (3.11) over the time interval $[0, T]$. In the simulations, the geometry as shown in Figure 3.4 is used. The implementation of the IP_3R channels in the domain requires multiple length scales, i.e. very fine mesh in the vicinity of the channel area and a very coarse mesh in other parts. This has been achieved using *a posteriori* error estimators, as explained in Section 4.5. For the mesh generation a special function is used which has a spatially localized active source term in the vicinity of channel subdomain at the ER membrane. The initial mesh and adaptive refined meshes are created by using the programm package

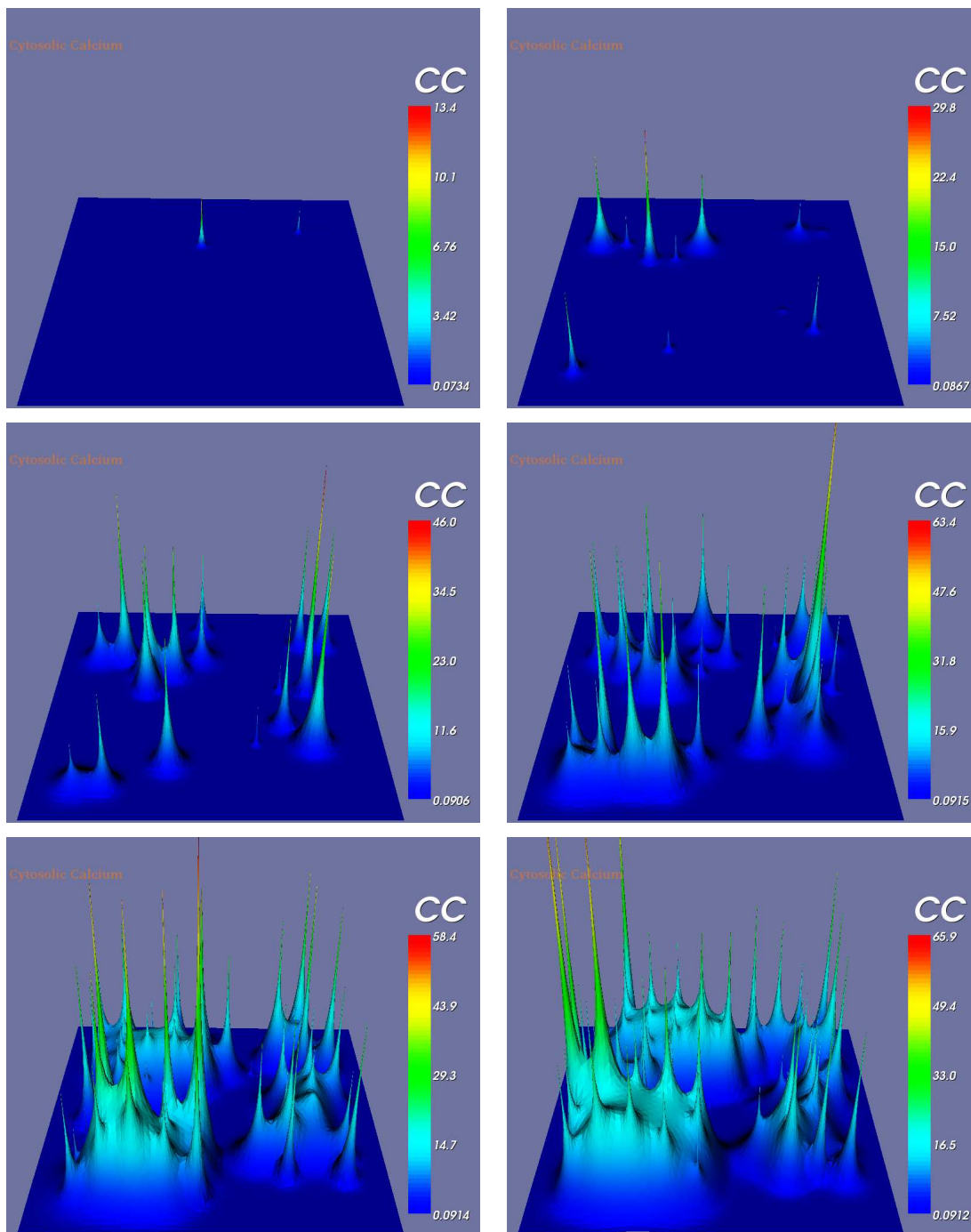


Figure 6.24: Solution of cytosolic concentration of 100 clusters with stochastic channel transition of channels at different time steps $t = 5.139571$ s, 5.439938 s, 5.689859 s, 6.089905 s, 6.339917 s and 6.489918 s.

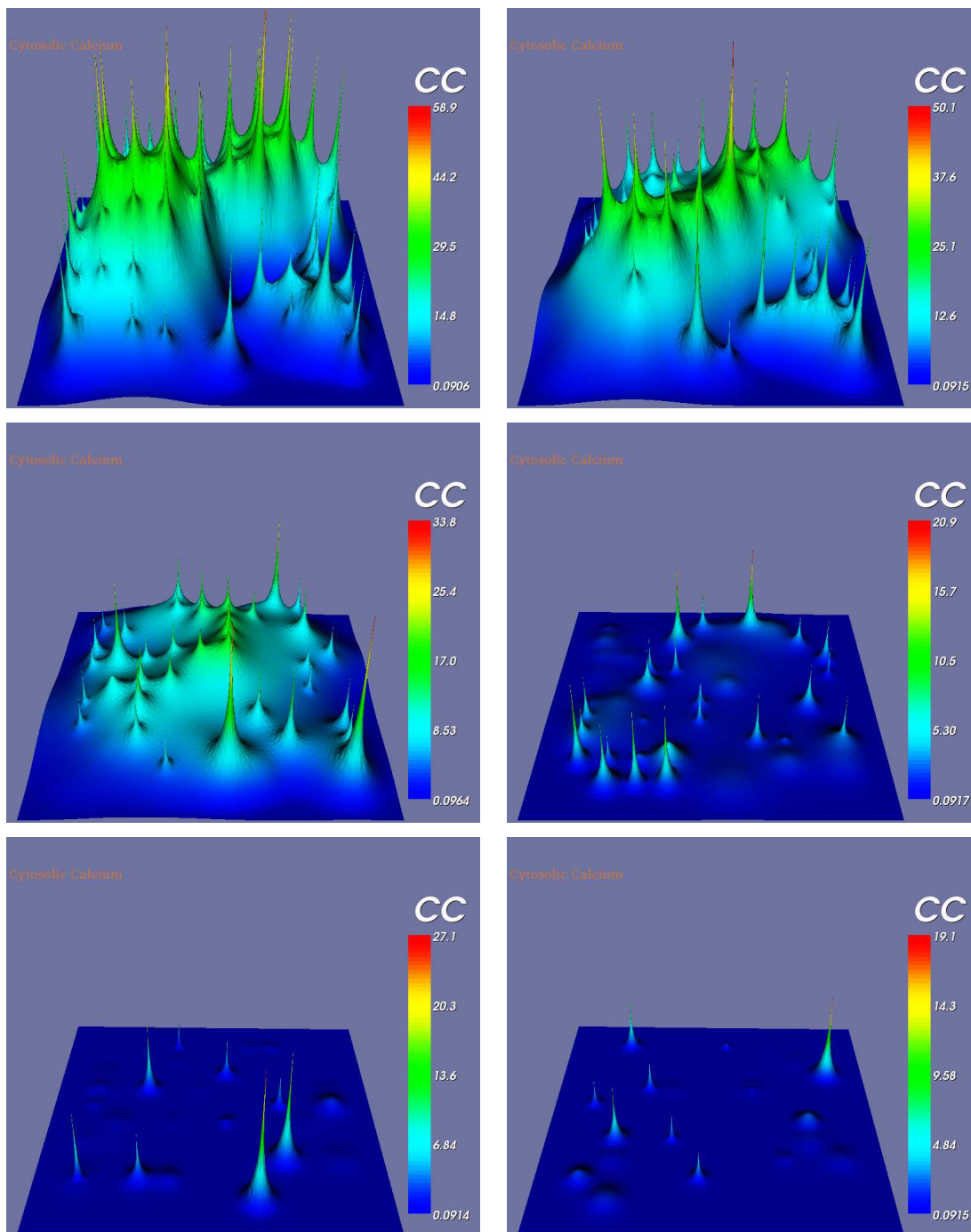


Figure 6.25: Solution of cytosolic concentration of 100 clusters with stochastic channel transition of channels at different time steps $t = 6.889927$ s, 7.189863 s, 7.589875 s, 8.589918 s, 10.589987 s and 14.089910 s.

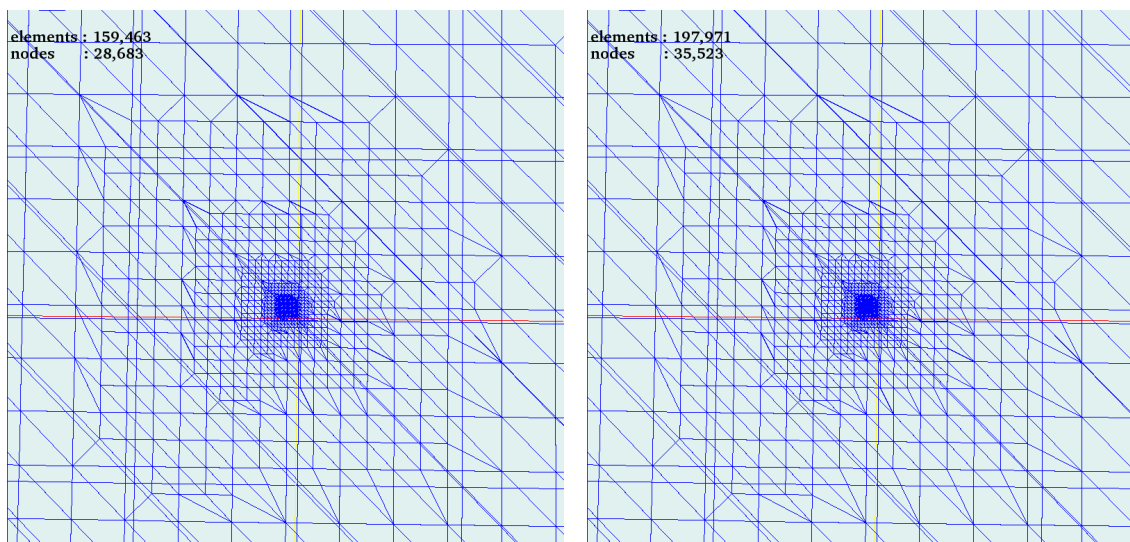


Figure 6.26: The zoom of the spatial grid near the channels of level 9 and level 12.

UG [12], see Figure 4.7. A close up view of the local resolution at the ER membrane of the unstructured finite element mesh of level 9 and 12 are shown in Figure 6.26.

The number of nodes, elements and the minimum volume of the element in the domain at different levels is presented in Table 6.3. It can be observed that the minimum volume of elements is very small with finer meshes where we want to put a fine mesh at the membrane.

levels	nodes	elements	min volume of element
0	729	3,072	$1.04167 \cdot 10^{08}$
8	26,403	146,627	3.10441
9	28,683	159,463	0.388051
10	30,963	172,299	0.0485064
11	33,243	186,135	0.0060633
12	35,523	197,971	0.000757912

Table 6.3: The number of nodes, elements and minimum volume of element in the domain, i.e. the element present at the membrane, at different mesh levels in cube.

In our numerical simulations we considered only one cluster arrangement. The numerical results of different grid structures with deterministic opening of one channel are tested. The left hand side of Figure 6.27 shows the average cytosolic Ca^{2+} against the time until $t = 0.1$ s and the maximum cytosolic Ca^{2+} over the number of nodes is presented in the right hand side of Figure 6.27. From these results, one can observe that mesh level 9 is

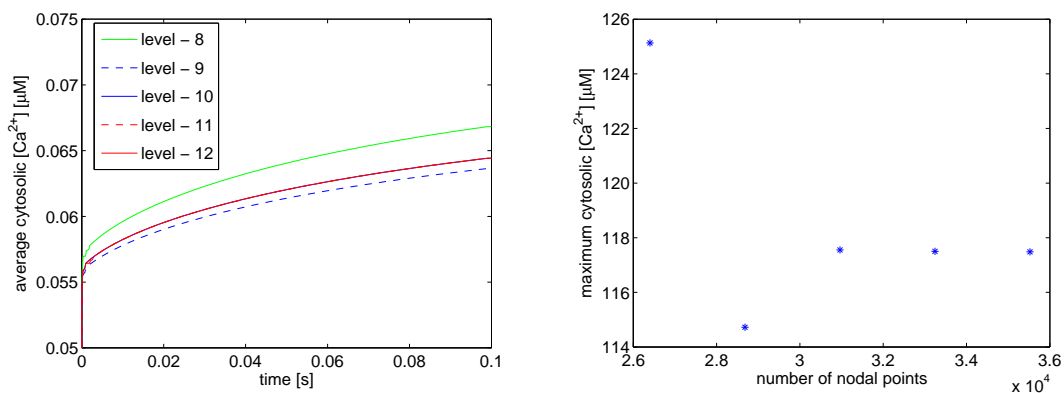


Figure 6.27: The average solution of cytosolic Ca^{2+} at different levels versus time for an one open channel at **left** and at **right** the maximum cytosolic Ca^{2+} versus levels at the stationary solution.

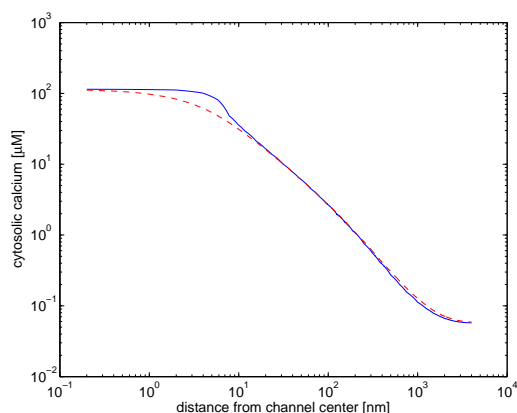


Figure 6.28: The stationary Ca^{2+} concentration for an open channel with a distance of $200nm$ from the channel center directly at the ER membrane (solid) and perpendicular to the membrane (dashed).

sufficient to consider for achieving the numerically convergent solutions. Therefore, the level 9 is considered in all other simulations.

In Figure 6.28, the steady state Ca^{2+} concentration for an open channel with distance of $4000nm$ from channel center directly at the ER membrane and perpendicular to the membrane. From this result, one can observe that the spatial profile of the Ca^{2+} concentration is similar at the vertical and the horizontal directions. The solution is plotted using log-log scale. The maximum Ca^{2+} concentration at the channel center is $114.400 \mu M$.

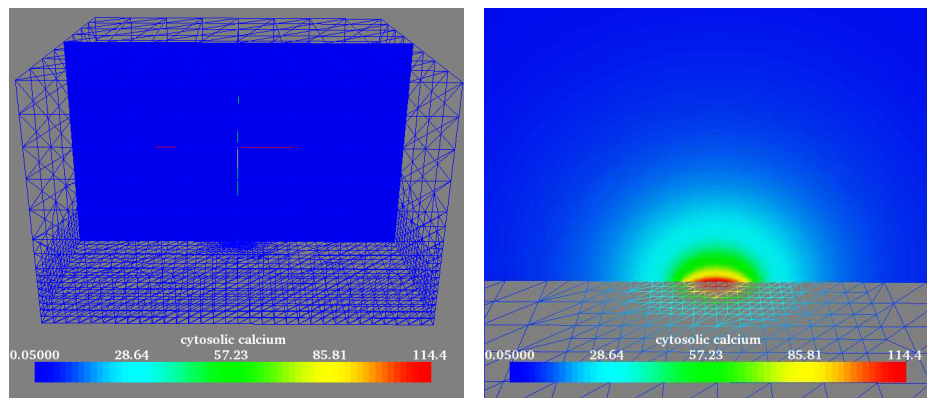


Figure 6.29: The spatial profile of the Ca^{2+} concentration at **left** and the zoom around the channel mouth at the ER membrane at **right** at time $t = 0.1$ s.

The spatial profile of the Ca^{2+} concentration is plotted in the left hand side of Figure 6.29 at time $t = 0.1$ s and the zoom around the channel mouth at the ER membrane is depicted in the right hand side of Figure 6.29. Here, we can see a strong profile of the Ca^{2+} concentration from the channel center to some hundreds of nano meters. Usually this strong localization restricts the time steps in numerical simulations.

Next, the numerical results with the hybrid simulation are presented, which solves the deterministic and stochastic equations simultaneously. The algorithm is based on a recently introduced approach for simulating hybrid models of chemical reaction kinetics in spatially homogenous systems [2], as explained in Section 3.3. Here, the numerical results are presented based on the hybrid method for a single channel system. Analogously we can generalize to multi channel systems.

A stochastic model is adopted for the gating of subunits. This stochastic model is based on the DeYoung-Keizer-model for the subunit dynamics [29]. An IP_3R consists of four identical subunits. There are three binding sites on each subunit in the framework of that model: An activating site for Ca^{2+} , an inhibiting Ca^{2+} site and an IP_3 binding site. The three binding sites allow 8 different states X_{ijk} of each subunit. The index i indicates the state of the IP_3 site, j the one of the activating Ca^{2+} site and k the state of the inhibiting Ca^{2+} site. An index is 1, if an ion is bound and 0 if not. In 3D simulations, one extra state X_{ACT} is considered where it is assumed that the channel is open, if at least three of the subunits are in X_{ACT} , i.e. they have bound to Ca^{2+} and IP_3 at the activating site. This additional state enables us to fit short mean open and mean close times. These could not be fitted using standard DeYoung-Keizer-model. For more results regarding open

probability, mean and close times can be found in paper by Rüdiger et al. [80]. Then we associate stochastic variables X_{000} , X_{001} , \dots , X_{ACT} to each channel. These variables count the number of subunits which are in the respective states.

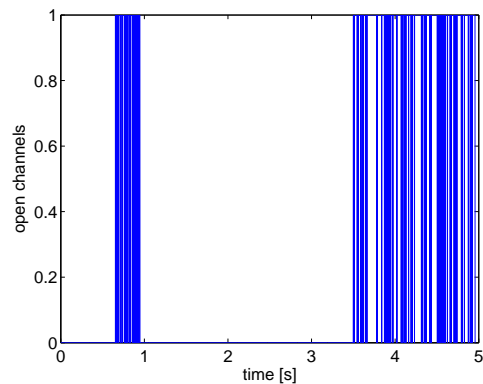


Figure 6.30: The opening and closing of the channel in one cluster versus time.

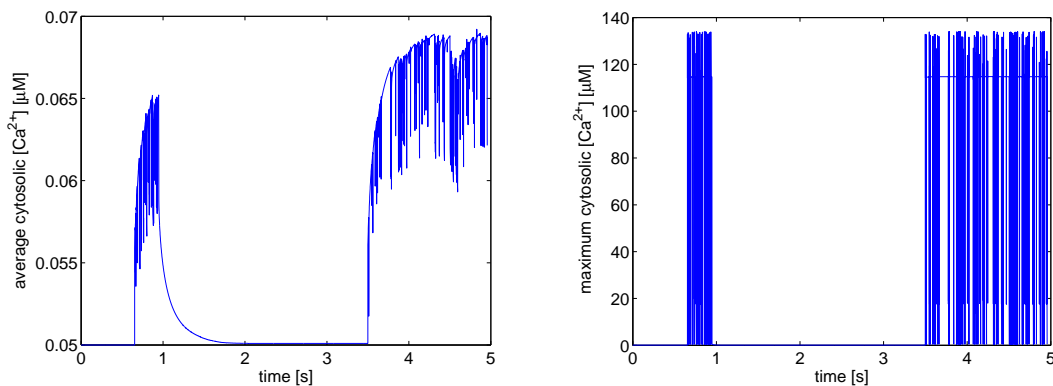


Figure 6.31: The cytosolic Ca^{2+} concentration over time; **left**: average value, **right**: maximum value at the channel mouth.

In Figure 6.30, the opening and closing of single channel against the time until $t = 5$ s are depicted. Here we can observe the rapid changings of the channel opening and closings. The corresponding average and the maximum cytosolic Ca^{2+} concentrations at the channel mouth are depicted in Figure 6.31.

6.3 Numerical results using domain decomposition methods

In sequential computation the domain and the multigrid hierarchy are stored in one processor's memory. This single processor performs all the computational operations on the multigrid mesh hierarchy to solve a numerical problem. In our transient numerical simulations to compute the solution till time 100 s in two space dimensions, the CPU time required is around 60 days on Opteron Linux machine with 2GB RAM and 2GHz of processor. To reduce the computational time and to increase the number of mesh elements to millions the use of parallel computer architectures is mandatory. In the parallel numerical simulations the most important parts are the parallel grid manager and the parallel implementation of linear solvers. For the domain decomposition we have used the graph partitioning package CHACO [50]. In our problem data partitioning is the most important factor. The load balancing scheme Recursive Inertial Bisection (RIB) serves well for this problem. In the context of UG, load migration is realized based on the parallel programming model DDD (Dynamic Distributed Data), see Birken [20]. Load balancing has been achieved as follows: the meshes of level-0 to level-5 have been kept on one processor and the level-6 mesh has been distributed to all processors. We use a linear solver with preconditioning at the finest level. For solving the linear system at each stage of the Rosenbrock method we have used the BiCGSTAB method [92] with point-block ILU preconditioning that has been damped with $\omega = 0.9$. The mesh decomposition to different processors is shown in Figure 6.32. Computations for this problem have been carried out on HP-UX B.11.11 U 9000/800 machines with 2GB RAM for each processor and these are connected to a 64 node cluster with 3GFOLPS processor speed at Institute for Analysis and Numerics, Magdeburg.

The domain decomposition of 100 clusters mesh is shown in Figure 6.32 for 16, 32 and 56 processors. In this figure we can clearly observe the domain decomposition of clusters when the number of processors increases, see Nagaiah et al. [69] for some more discussion.

In the numerical simulations the opening of channel is considered in a prescribed way. The Test Case 3 is consider for the numerical simulations. Here we tested the efficiency for 1, 4, 16, 32 and 56 processors. In this case 5 channels are opened in all clusters up to time $t = 1 \cdot 10^{-3}$ s and all channels are closed after $t = 1 \cdot 10^{-3}$ s. The average solution of cytosolic calcium concentration is shown up to time $t = 2 \cdot 10^{-3}$ s. Here we can see good agreement between all processors. But solution deviates when the number of processors increase, see Figure 6.33. Because this is mainly depends on the decomposition of mesh.

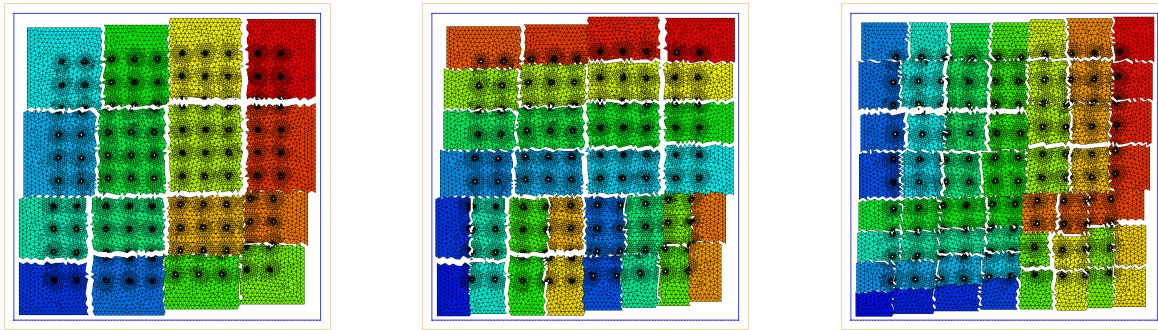


Figure 6.32: Domain decomposition of 100 clusters adaptive mesh using 16, 32 and 56 processors

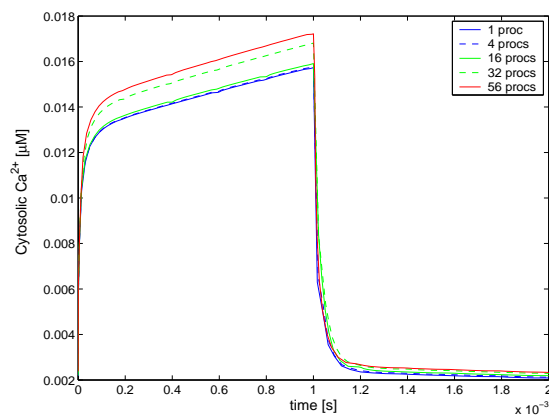


Figure 6.33: Efficiency test for 1, 4, 16, 32 and 56 processors.

no. of procs	unknowns	time steps	cpu time	efficiency
1	133,296	10	26m 46s	1.0
16	133,296	10	2m 16s	0.7381
32	133,296	10	1m 2s	0.8095
48	133,296	10	38s	0.8805
56	133,296	10	32s	0.8962

Table 6.4: Comparison of CPU times on different number of different processors.

What we have observed is that, when domain decomposition of the clusters is divided to many processors then loss in accuracy starts.

Performance data of the simulations are presented in Table 6.4. For comparison, the time step size is kept constant in all simulations. The first column shows the number of

processors used and the last column shows the efficiency of the processors. This efficiency can be calculated by using the relation $\frac{1}{P} \frac{T(1)}{T(P)}$, where $T(1)$ and $T(P)$ are total CPU time for 1 processor and P processors. Because of the data structure of the programming package the efficiency increases if we increase the number of processors. The increase in efficiency for 56 processors is 89.62%.

Chapter 7

Summary

In this work the numerical behavior of two specific problems has been studied effectively. They are the heat and mass transfer in fluidized beds as well as a calcium signalling task in cell biology. The semilinear partial differential equations that arise from the modeling of the above problems have been solved very efficiently using a finite element method.

In the first application problem, the numerical solutions of heat and mass transfer in fluidized beds with liquid spray injection were computed proficiently. The governing equations were derived in Chapter 2 using mass and energy balances of air, particles, and liquid. This model is an improved model in comparison to the model presented in Heinrich [47]. Also a sophisticated numerical method was applied to solve the model equations. The mathematical model equations of liquid nozzle spray were presented for two and three dimensions in Sections 2.3 and 2.3 respectively. Here more than one nozzle can be arranged arbitrarily in the fluidized bed. These model equations are more flexible to compute the numerical solution of balance equations on unstructured meshes and also more suitable to compute parallel numerical solutions. In numerical simulations the balance equations are considered in Cartesian coordinates. In the present work, the mathematical solution of the continuum model was approximated by using a finite element method for the space discretization and the implicit Euler method for the time discretization in order to improve the numerical efficiency as presented in Chapter 4.

A parametric study has been conducted to see the influence of liquid film thickness, ratio of heat transfer coefficient, air inlet temperature, air mass flow rate, liquid mass flow rate and diameter of particles on heat and mass transfer in fluidized beds. Furthermore, the time-dependent two-dimensional distributions of air humidity, air temperature, particle wetting, liquid film temperature and particle temperature were simulated, which showed

almost isothermal behavior and good mixing conditions in the fluidized beds. The numerical convergence of approximate solutions to the balance equations was presented using a semi implicit method, an implicit method and different linear solvers. These tests demonstrate that the numerical method gives convergent solutions. The experimental order of convergence (EOC) of almost 1 is obtained for the first order implicit Euler method with finer meshes, see Chapter 5. This result reveals that the method has a convergence order that agrees with theoretical considerations.

The three dimensional numerical results were presented in Section 5.2. Two test cases were considered for numerical simulations, the required process parameters were taken from the doctoral theses of Heinrich [47] and Henneberg [51] respectively. The steady state solutions of air humidity, air temperature, degree of wetting, liquid film temperature, particle temperature and saturation of air humidity were presented for both test cases. Numerical results with four spray nozzles were also given. In the heat transfer zone, which is above the distributor plate, the air temperature decreases due to energy absorbed by the wetted surface of the particles. The liquid on the particles evaporates through its direct contact with the hot air. Due to this the air humidity rises. In the numerical simulations this area is identified as a stiff region which controls the time step. The transient air temperature simulation results are qualitatively in good agreement with experimental results. The increase of the number of spray nozzles requires the use of more mesh points, which ultimately requires more computational power. For this reason, parallel methods were used to get faster and efficient results using domain decomposition methods. The parallel numerical results with domain decomposition methods showed good load balancing with the recursive inertial bisection (RIB) algorithm when compared to the recursive coordinate bisection (RCB) algorithm. The parallel efficiency showed a good load balancing for different numbers of processors.

The second application problem was concerned with efficient and accurate numerical solutions of deterministic equations including stochastic channel transition. The numerical convergence of solutions of the deterministic equations was tested extensively by consideration of one open channel in one cluster, see Section 6.1. These convergence results showed evidence of correctness of the numerical method along with the code implementation. A comparison of the *W*-method and the ROS3P method was presented at different tolerances. According to these results we conclude that both methods give convergent solutions but the *W*-method takes less CPU time as compared to the ROS3P method. It was also observed that the grid adaption plays an important role when a channel is opened for a

while. Moreover, in this case a fixed mesh may not be capable to maintain accurate and convergent solutions unless a very fine mesh is used in the vicinity of the channel area. The grid adaptivity was applied for one cluster and 16 cluster arrangements successfully. In the present work a recent hybrid method was adopted which solve the deterministic and stochastic equations simultaneously, see Alfonsi et al. [2]. The hybrid algorithm uses two time scales, one is for solving the deterministic equations and second is for solving the stochastic equations. The numerical results were presented using hybrid method for different cluster arrangements in subsections 6.1.2 and 6.1.3.

In 3D calculations, the model uses an additional active state compared to the standard DYK model. This additional state enables us to fit short mean opening and mean closing times. These could not be fitted using the DYK model. We refer to recent a paper of Rüdiger et al. [80] for more results on the opening probability, mean opening and mean closing times as well as bursts of rapid opening and closing of channels. The numerical results presented in Section 6.2 demonstrate that the numerical method is able to compute convergent solutions for this problem.

Longer computations, e.g. a simulation time until 100 s for larger cluster arrangements, on one machine take around 50 days. To reduce this computational time and to get faster results, parallel computations were applied to this problem. The domain decomposition and mapping of the finite element domain onto the parallel machine was accomplished by a graph partitioning tool CHACO [50]. The parallel numerical solutions were presented using the recursive inertial bisection (RIB) algorithm. For these numerical simulations the opening of channel was considered in a prescribed way. As a test case, one channel was opened in a each cluster of 100 cluster arrangements. It was observed in the numerical simulations that the efficiency of the algorithm increased with increasing numbers of processors. The most important task is the assignment of the domain decomposition of the discretized ER membrane on to several processors. This is very challenging. From the results, we observed that the numerical solutions are deviating with increasing number of processors. This mostly occurs when the distribution of the one cluster is assigned to more than one processors. Study of this problem is in progress.

In the future, we intend to do numerical simulations for larger cluster arrangements efficiently, e.g. by introducing sparse structures in the code implementation to solve these reaction-diffusion systems. Then the computational time will reduce drastically. The work on extension of the grid adaptivity for larger cluster arrangements is in progress. And also the work on extension of the parallelization of the hybrid method has began. Considering

all these tasks, these methods are to be applied to three dimensions extensively. Extension of the 3D simulations to get accurate and efficient solutions using the domain decomposition methods are a very challenging problem.

Regarding the first application problem, in the simulations, one can include the effect of direct evaporation in the spray liquid area and also include porosity terms in the balance equations. In the future, it is possible to couple the nozzle spray equation, in more than one space dimension, to the balance equations. Also solving the 3D balance equations including more than one spray nozzle is very interesting using parallel architectures.

Appendix A

Simulation Parameters for Heat and Mass Transfer in Fluidized Beds

All these parameters taken from Ph. D thesis of Henneberg [51].

Dimensionless numbers

$$\text{Archimedes number } Ar = \frac{gd_P^3(\rho_P - \rho_A)}{\nu_A^2 \rho_A}$$

$$\text{Nusselt number } Nu = \frac{\alpha L}{\lambda_A}$$

$$\text{Prandtl number } Pr = \frac{c_P \nu_A \rho_A}{\lambda_A}$$

$$\text{Reynolds number } Re = \frac{w_A d_P}{\nu_A}$$

$$\text{Schmidt number } Sc = \frac{\nu_A}{D_V}$$

$$\text{Sherwood number } Sh = \frac{\beta d_P}{D_V}$$

Calculation of the parameters

Cross section surface of the apparatus

$$A_{App} = \frac{\pi}{4} d_{App}^2$$

Reynolds number at the fluidization point

$$Re_{mf} = \frac{Ar}{1400 + 5.22\sqrt{Ar}}$$

Minimal fluidization velocity

$$w_{mf} = \frac{Re_{mf} \nu_A}{d_P}$$

Porosity of the fluidized bed

$$\epsilon = \left[\frac{18Re + 0.36Re^2}{Ar} \right]^{0.21} = \frac{V_{void}}{V_{fb}} = \frac{V_{void}}{V_{void} + V_P} \quad \text{where } V_P = \frac{m_P}{\rho_P}$$

Fluidized bed height

$$H_{fb} = \frac{V_{fb}}{A_{App}} = \frac{m_P}{A_{App}\rho_P(1 - \epsilon)}$$

Overall surface of the solid particles

$$A_P = \frac{6A_{App}H_{fb}(1 - \epsilon)}{d_P}$$

Mass transfer coefficient

$$\beta = \frac{D_V Sh}{d_P}$$

Sherwood number

$$Sh = 2 + 0.72\sqrt{Re}\sqrt[3]{Sc}$$

Heat transfer coefficient

$$\alpha_{AP} = \frac{Nu_{AP}\lambda}{d_P}$$

Nusselt number

$$Nu_{AP} = (2 + \sqrt{Nu_{lam}^2 + Nu_{tur}^2})[1 + 1.5(1 - \epsilon)]$$

$$Nu_{lam} = 0.664Pr^{1/2}Re_\epsilon^{1/2}$$

$$Nu_{tur} = 0.037 \frac{PrRe_\epsilon^{0.8}}{1 + 2.443Re_\epsilon^{-0.1}(Pr^{2/3} - 1)}$$

Parameter values

Dry air

Specific gas constant

$$R_A = 287.22$$

Density

$$\rho_A = \frac{P}{R_A(\theta_A + 273.15)}$$

Specific heat capacity of the air

$$c_A = 1006.256 + 2.120536 \cdot 10^{-2}\theta_A + 4.180195 \cdot 10^{-4}\theta_A^2 - 1.521916 \cdot 10^{-7}\theta_A^3$$

Thermal conductivity of air

$$\lambda_A = 24.5211 \cdot 10^{-3} + 7.501414 \cdot 10^{-5}\theta_A - 2.593344 \cdot 10^{-8}\theta_A^2 + 5.292884 \cdot 10^{-11}\theta_A^3$$

Dynamic viscosity

$$\eta_A = 1.705568 \cdot 10^{-5} + 4.511012 \cdot 10^{-8} \theta_A - 8.766234 \cdot 10^{-12} \theta_A^2 - 3.382035 \cdot 10^{-15} \theta_A^3$$

Kinematic viscosity

$$\nu_A = \frac{\eta_A}{\rho_A}$$

Prandtl number

$$Pr_A = \frac{\eta_A c_A}{\lambda_A}$$

Water

Density of liquid

$$\rho_L = 1006.0 - 0.26 \theta_L - 0.0022 \theta_L^2$$

Specific heat capacity of the liquid

$$c_L = 4174.785 + 1.785308 \cdot 10^{-2} \theta_L - 5.097403 \cdot 10^{-4} \theta_L^2 + 4.216721 \cdot 10^{-5} \theta_L^3$$

Specific heat evaporation of the water at 0°C

$$\Delta h_V = 2500000.0$$

Saturation pressure

$$p_{sat} = \exp \left(23.462 - \frac{3978.205}{233.349 + \theta_L} \right)$$

Nomenclature and Simulation Parameters

Notations

A	surface area, (m^2)
A^*	volume based surface, (m^2/m^3)
c_p	specific heat capacity at constant pressure, ($J/(kgK)$)
d	diameter, (m)
\mathbf{D}	dispersion matrix, (m^2/s)
F	liquid film thickness, (m)
f	heat transfer ratio, (dimensionless)
h	enthalpy, (J)
Δh_V	specific heat of evaporation of the water, (J/kg)
$\Delta h_{V,0}$	specific heat of evaporation of the water at 0°C , (J/kg)
H	bed height, (m)
L	length, (m)
m	mass, (kg)
\dot{m}	mass flow, (kg/s)
\tilde{M}	molar mass, (kg/mol)
p	pressure, (Pa)
q	area based heat, J/m^2
\dot{q}	area based heat flow, $J/(m^2 s)$
r	radius, (m)
R	distance, (m)
\tilde{R}	universal gas constant, ($J/(molK)$)
s	length, (m)
t	time, (s)
V	volume, (m^3)
w	velocity, (m/s)
Y	humidity, (kg/kg)

Greek symbols

α	heat transfer coefficient, ($W/(m^2K)$)
β	mass transfer coefficient, (m/s)
ϵ	porosity of the fluidized bed, (m^3/m^3)
ϵ^*	solid particle volume fraction, (m^3/m^3)
θ	temperature, ($^{\circ}C$)
κ	concentration, (kg/m^3)
$\tilde{\kappa}$	concentration flow, $kg/(m^3s)$
λ	thermal conductivity, ($W/(mK)$)
ν	kinematic viscosity, (m^2/s)
τ	final time, (s)
ρ	density, (kg/m^3)
ϕ	wetting efficiency, dimensionless

Subscripts

A	air
App	apparatus
dr	drop
in	inflow
fb	fluidized bed
L	liquid
nozz	nozzle
P	particle
sat	saturation
V	vapor
W	wall
0	state at time $t = 0$
∞	saturation value

Fluidized bed parameters			
width	L	0.4	m
height	H_{fb}	0.2	m
Bed material (glass spheres)			
total mass	m_P	18	kg
diameter	d_P	1.16	mm
density	ρ_P	2471	kg/m ³
specific heat capacity	c_P	750	J/(kg K)
thermal conductivity	λ_P	0.8	W/(m K)
Fluidization air			
mass flow rate	\dot{m}_A	0.304	kg/s
inlet humidity	$Y_{A,in}$	0.008	kg/kg
inlet temperature	$\theta_{A,in}$	80	°C
Liquid spraying			
spraying rate (source flow)	\dot{m}_L	5.9	kg/h
liquid inlet temperature	$\theta_{L,in}$	20	°C
Other parameters			
apparatus wall temperature	θ_W	75	°C
liquid film thickness	F	100	μm
heat transfer ratio	f	1	-
Initial parameters for the simulation			
air humidity	$Y_{A,0}$	0.008	kg/kg
air temperature	$\theta_{A,0}$	80	°C
degree of wetting	ϕ_0	1e-8	-
liquid film temperature	$\theta_{L,0}$	20	°C
particle temperature	$\theta_{P,0}$	80	°C

Table A.1: Parameters values used for simulations in 2D

Fluidized bed parameters			
diameter	d_{fb}	0.4	m
height	H_{fb}	0.2	m
Bed material (glass spheres)			
total mass	m_P	20	kg
diameter	d_P	0.00116	m
density	ρ_P	2471	kg/m ³
specific heat capacity	c_P	750	J/(kg K)
thermal conductivity	λ_P	0.8	W/(m K)
Fluidization air			
mass flow rate	\dot{m}_A	0.378	kg/s
inlet humidity	$Y_{A,in}$	0.008	kg/kg
inlet temperature	$\theta_{A,in}$	80	°C
Liquid spraying			
spraying rate (source flow)	\dot{m}_L	13.79	kg/h
liquid inlet temperature	$\theta_{L,in}$	20	°C
Other parameters			
liquid film thickness	F	100	μm
heat transfer ratio	f	1	-
Initial parameters for the simulation			
air humidity	$Y_{A,0}$	0.008	kg/kg
air temperature	$\theta_{A,0}$	80	°C
degree of wetting	ϕ_0	1e-8	-
liquid film temperature	$\theta_{L,0}$	20	°C
particle temperature	$\theta_{P,0}$	80	°C

Table A.2: Parameters values used for simulation of experiment 1 in 3D

Fluidized bed parameters			
diameter	d_{fb}	1.5	m
height	H_{fb}	0.6	m
Bed material (glass spheres)			
total mass	m_P	370	kg
diameter	d_P	0.0033	m
density	ρ_P	1377	kg/m ³
specific heat capacity	c_P	980	J/(kg K)
thermal conductivity	λ_P	0.17	W/(m K)
Fluidization air			
mass flow rate	\dot{m}_A	7.0	kg/s
inlet humidity	$Y_{A,in}$	0.005	kg/kg
inlet temperature	$\theta_{A,in}$	60	°C
Liquid spraying			
spraying rate (source flow)	\dot{m}_L	90	kg/h
liquid inlet temperature	$\theta_{L,in}$	20	°C
Other parameters			
apparatus wall temperature	θ_W	75	°C
liquid film thickness	F	100	μm
heat transfer ratio	f	1	-
Initial parameters for the simulation			
air humidity	$Y_{A,0}$	0.008	kg/kg
air temperature	$\theta_{A,0}$	80	°C
degree of wetting	ϕ_0	1e-8	-
liquid film temperature	$\theta_{L,0}$	20	°C
particle temperature	$\theta_{P,0}$	80	°C

Table A.3: Parameters values used for simulation of experiment 2 in 3D

Appendix B

Hybrid Algorithm and Simulation Parameters used for Ca^{2+} Dynamics

The algorithmic realization of our hybrid approach is given below.

1. Initialization

- Set $t_{old} = 0$, $\Delta t > 0$, $c_{old} = c_0$, $X = X_0$, $g_{old} = 0$ and draw a uniform random number r_1 in $[0,1]$ defining $\xi = \ln(1/r_1)$.

2. Deterministic step

- Compute c_{new} and g_{new} based on c_{old} , g_{old} and Δt .
- If the local error criterion is not met, reduce the step size Δt and go to 2., otherwise define $t_{new} = t_{old} + \Delta t$ and set the new step size Δt according to the time stepping code prediction.

3. If $g_{new} < \xi$ (no stochastic event)

- Set $c_{old} = c_{new}$, $g_{old} = g_{new}$, $t_{old} = t_{new}$, and go to 2.

4. If $g_{new} \geq \xi$ (some stochastic event in the time interval $[t_{old}, t_{new}]$)

- Determine the event time $t_s \in [t_{old}, t_{new}]$ by (linear) interpolation, and compute the corresponding calcium concentration c_s at the event time t_s by (linear) interpolation.
- Draw a uniform random number r_2 in $[0, 1]$ and determine the stochastic event R_i according to Eq. (3.21) based on c_s .

5. If the next event R_i is non-channel transition

- Perform the stochastic event R_i to determine the new channel state X .
- Set $g_{old} = 0$ and recompute g_{new} based on c_s , g_{old} and the remaining time $(t_{new} - t_s)$.
- Draw a new uniform random number r_1 in $[0, 1]$ defining $\xi = \ln(1/r_1)$, and go to 3.

6. If the next event R_i is a channel transition

- Perform the channel transition R_i to determine the new state X .
- Set $g_{new} = 0$, and draw a new uniform random number r_1 in $[0, 1]$ defining $\xi = \ln(1/r_1)$.
- Set $t_{new} = t_s$, and define new step size $\Delta t = \Delta t_{channel}$ (a sufficiently small number).
- Set $c_{old} = c_s$, and go to 2.

Algorithm 3: Hybrid algorithm

Parameter	Value	Unit
leak flux coefficient P_l	0.025	nm s^{-1}
channel flux coefficient P_{ch}	3.0×10^3	nm s^{-1}
single channel radius R_s	0.018	μM
pump flux coefficient P_p	200	$\mu\text{M s}^{-1}$
pump dissociation coefficient K_d	0.04	μM
diffusion coefficient D of free cytosolic Ca^{2+}	200	$\mu\text{m}^2 \text{s}^{-1}$
diffusion coefficient D of free ER Ca^{2+}	200	$\mu\text{m}^2 \text{s}^{-1}$
diffusion coefficient D_{dye} of dye buffer	70	$\mu\text{m}^2 \text{s}^{-1}$
diffusion coefficient D_m of mobile buffer	40	$\mu\text{m}^2 \text{s}^{-1}$
diffusion coefficient D_s of dye buffer	0.01	$\mu\text{m}^2 \text{s}^{-1}$
on-rates of fast buffers:		
k_s^+	200	$(\mu\text{M s})^{-1}$
k_m^+	400	$(\mu\text{M s})^{-1}$
k_{dye}^+	100	$(\mu\text{M s})^{-1}$
dissociation constants of buffers $K_i = \frac{k_i^-}{k_i^+}$:		
K_s	2	μM
K_m	0.25	μM
K_{dye}	0.16	μM
total concentrations of buffers:		
B_s	80	μM
B_m	1	μM
B_{dye}	46	μM
subunit kinetics (model A), note $b_i = a_i d_i$, $i=1, \dots, 5$		
IP3 binding		
a_1, a_3	20	$(\mu\text{M s})^{-1}$
d_1	0.13	μM
d_3	0.13	μM
inhibiting, with IP3		
a_2	0.030373	$(\mu\text{M s})^{-1}$
d_2	3.776	μM
inhibiting, without IP3		
a_4	0.303073	$(\mu\text{M s})^{-1}$
d_4	0.5202	μM
activating		
a_5	2.222	$(\mu\text{M s})^{-1}$
d_5	0.3	μM

Table B.1: Parameters used in 2D numerical simulation

Parameter	Value	Unit
leak flux coefficient P_l	3.3613	nm s^{-1}
channel flux coefficient P	6.32×10^6	nm s^{-1}
single channel radius R_s	6	nm
pump flux coefficient P_p	40000	$\mu\text{M s}^{-1}$
pump dissociation coefficient K_d	0.2	μM
Ca concentration in ER	700	μM
diffusion coefficient D of free cytosolic Ca^{2+}	200	$\mu\text{m}^2 \text{s}^{-1}$
diffusion coefficient D_m of mobile buffer	200.0	$\mu\text{m}^2 \text{s}^{-1}$
diffusion coefficient D_{dye} of dye buffer	15.0	$\mu\text{m}^2 \text{s}^{-1}$
on-rates of fast buffers: k_s^+ k_m^+ k_{dye}^+	50 5 150	$(\mu\text{M s})^{-1}$ $(\mu\text{M s})^{-1}$ $(\mu\text{M s})^{-1}$
dissociation constants of buffers $K_i = \frac{k_i^-}{k_i^+}$: K_s K_m K_{dye}	2 0.15 2	μM μM μM
total concentrations of buffers: B_s B_m B_{dye}	80 300 40	μM μM μM
subunit kinetics (model A), note $b_i = a_i d_i$, $i=1, \dots, 5$ IP3 binding a_1, a_3 d_1 d_3 inhibiting, with IP3 a_2 d_2 inhibiting, without IP3 a_4 d_4 activating a_5 d_5 open conformational transition a_0 b_0	80 0.008 0.5 0.04 12 0.4 0.192 15 0.8 550 80	$(\mu\text{M s})^{-1}$ μM μM $(\mu\text{M s})^{-1}$ μM $(\mu\text{M s})^{-1}$ μM $(\mu\text{M s})^{-1}$ μM μM μM

Table B.2: Parameters used in 3D numerical simulation

Bibliography

- [1] R. A. Adams. *Sobolev Spaces*. Cambridge University Press, 2001.
- [2] A. Alfonsi, E. Cancs, G. Turinici, B. Di Ventura, and W. Huisinga. Exact simulation of hybrid stochastic and deterministic models for biochemical systems. *INRIA Rapport de Recherche, Thèmes NUM et BIO*, 5435, 2004.
- [3] I. Babuška and W. C. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM Journal on Numerical Analysis*, 15(4):736–754, 1978.
- [4] R. E. Bank, A. Sherman, and A. Weiser. Refinement algorithms and data structures for regular local mesh refinement. *Scientific Computing*, 1983.
- [5] R. E. Bank and R. K. Smith. A posteriori error estimates based on hierarchical bases. *SIAM Journal on Numerical Analysis*, 30(4):921–935, 1993.
- [6] R. E. Bank and A. Weiser. Some a posteriori error estimators for elliptic partial differential equations. *Math. Comput.*, 44:283–301, 1985.
- [7] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA, 1994.
- [8] P. Bastian. Parallel adaptive multigrid methods. Technical Report 93–60, Interdisziplinäres Zentrum für Wissenschaftliches Rechnen, 1993.
- [9] P. Bastian. *Parallele adaptive Mehrgitterverfahren*. PhD thesis, Teubner-Verlag, 1996.
- [10] P. Bastian. Load balancing for adaptive multigrid methods. *SIAM J. Sci. Stat. Comput.*, 19(4):1303–1321, 1998.

- [11] P. Bastian. *Numerical computation of multiphase flow in porous media*. Habilitationsschrift, 1999.
- [12] P. Bastian, K. Birken, S. Lang, K. Johannsen, N. Neuß, H. Rentz-Reichert, and C. Wieners. UG: A flexible software toolbox for solving partial differential equations. *Computing and Visualization in Science*, 1:27–40, 1997.
- [13] R. D. Becher and E.-U. Schlünder. Fluidized bed granulation: gas flow, particle motion, and moisture distribution. *Chem. Eng. Proc.*, 36:261–269, 1997.
- [14] M. Berridge. Elementary and global aspects of calcium signalling. *J. Physiol.*, 499:291–306, 1997.
- [15] M. J. Berridge. Calcium oscillations. *J. Biol. Chem.*, 265(17):9583–9586, 1990.
- [16] M. J. Berridge. Inositol trisphosphate and calcium signalling. *Nature*, 361:315, 1993.
- [17] M. J. Berridge, M. D. Bootman, and P. Lipp. Calcium - a life and death signal. *Nature*, 395:645–648, 1998.
- [18] M. J. Berridge, P. Lipp, and M. D. Bootman. The versatility and universality of calcium signalling. *Nature Rev. Mol. Cell Biol.*, 1:11–22, 2000.
- [19] I. Bezprozvanny, J. Watras, and B. E. Ehrlich. Bell-shaped calcium-response curves of Ins(1,4,5)P₃- and calcium-gated channels from endoplasmatic reticulum of cerebellum. *Nature*, 351:751–754, 1991.
- [20] K. Birken and P. Bastian. Distributed Dynamic Data (DDD) in a parallel programming environment - Specification and functionality. Technical Report RUS–22, Rechenzentrum der Universität Stuttgart, 1994.
- [21] M. D. Bootman, T. J. Collins, C. M. Peppiatt, L. S. Prothero, L. MacKenzie, P. DeSmet, M. Travers, S. C. Tovey, J. T. Seo, M. J. Berridge, F. Ciccolini, and P. Lipp. Calcium signalling - an overview. *Seminars in Cell & Developmental Biology*, 12(1):3–10, 2001.
- [22] F. A. Bornemann, B. Erdmann, and R. Kornhuber. A posteriori error estimates for elliptic problems in two and three space dimensions. *SIAM Journal on Numerical Analysis*, 33(3):1188–1204, 1996.

- [23] D. Braess. *Finite elements: Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge University Press, 2001.
- [24] W. Briggs. *A Multigrid Tutorial*. SIAM, 1987.
- [25] S. Bruhns and J. Werther. An investigation of the mechanism of liquid injection into fluidized beds. *AIChE Journal*, 51(3):766–775, 2005.
- [26] N. Callamaras, J. S. Marchant, X.-P. Sun, and I. Parker. Activation and co-ordination of InsP_3 -mediated elementary Ca^{2+} events during global Ca^{2+} signals in *Xenopus* oocytes. *Journal of Physiology*, 509(1):81–91, 1998.
- [27] P. G. Ciarlet. *The finite element method for elliptic problems*. Society for Industrial and Applied Mathematics, 2002.
- [28] P. G. Ciarlet and J. L. Lions, editors. *Hand book of numerical analysis*, volume II. North-Holland, 1991.
- [29] G. DeYoung and J. Keizer. A single-pool inositol 1,4,5-trisphosphate-receptor-based model for agonist-stimulated oscillations in Ca^{2+} concentration. *Proc. Natl. Acad. Sci. USA*, 89:9895–9899, 1992.
- [30] C. J. Dixon, N. M. Woods, T. E. Webb, and A. K. Green. Evidence that rat hepatocytes coexpress functional p2y1 and p2y2 receptors. *Br. J. Pharmacol.*, 129:764–770, 2000.
- [31] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. John Wiley & Sons, 2003.
- [32] G. Dupont and S. Swillens. Quantal release, incremental detection, and long-period Ca^{2+} oscillations in a model based on regulatory Ca^{2+} -binding sites along the permeation pathway. *Biophys. J.*, 71:1714–1722, 1995.
- [33] K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems iv: Nonlinear problems. *SIAM Journal on Numerical Analysis*, 32(6):1729–1749, 1995.
- [34] M. Falcke. On the role of stochastic channel behavior in intracellular Ca^{2+} dynamics. *Biophys. J.*, 84(1):42–56, 2003.

- [35] M. Falcke. Reading the patterns in living cells - the Physics of Ca^{2+} signaling. *Advances in Physics*, 53(3):255–440, 2004.
- [36] M. Falcke, L. Tsimring, and H. Levine. Stochastic spreading of intracellular Ca^{2+} release. *Phys. Rev. E*, 62:2636–2643, 2000.
- [37] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.*, 22:403–434, 1976.
- [38] D. T. Gillespie. Exact stochastic simulating of coupled chemical reactions. *J. Phys. Chem.*, 81:2340–2361, 1977.
- [39] D T. Gillespie. *Markov Processes - An Introduction for Physical Scientists*. Academic Press, 1992.
- [40] V. Gnielinski. Wärme- und Stoffübertragung in Festbetten. *Chem.-Ing.-Tech.*, 52:228–236, 1980.
- [41] K. Gustafsson. Control-theoretic techniques for stepsize selection in implicit runge-kutta methods. *ACM Trans. Math. Softw.*, 20(4):496–517, 1994.
- [42] K. Gustafsson, M. Lundh, and G. Söderlind. A pi stepsize control for the numerical solution of ordinary differential equations. *BIT*, 28(2):270–287, 1988.
- [43] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer-Verlag, 1985.
- [44] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff problems*. Springer Series in Computational Mathematics, 1987.
- [45] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer Series in Computational Mathematics, 1991.
- [46] W. Heineken. *Adaptive Verfahren zur numerischen Berechnung von Reaktions-Diffusions-Systemen*. PhD thesis, Otto-von-Guericke-University Magdeburg, 2005.
- [47] S. Heinrich. *Modellierung des Wärme- und Stoffübergangs sowie der Partikelpopulationen bei der Wirbelschicht-Sprühgranulation*. PhD thesis, Otto-von-Guericke-University Magdeburg, VDI-Fortschrittbericht Reihe 3, No. 675, VDI-Verlag, 2001.

- [48] S. Heinrich, J. Blumschein, M. Henneberg, M. Ihlow, M. Peglow, and L. Mörl. Study of multi-dimensional temperature and concentration distributions in liquid sprayed fluidized beds. *Chem. Eng. Sci.*, 55:5135–5160, 2003.
- [49] S. Heinrich and L. Mörl. Fluidized bed spray granulation – a new model for the description of particle wetting and of temperature and concentration distribution. *Chem. Eng. Proc.*, 38:635–663, 1999.
- [50] B. Hendrickson and R. Leland. The CHACO user’s guide 1.0. Technical Report SAND93–2339, Sandia National Laboratories, 1993.
- [51] M. Henneberg. *Untersuchung des Flüssigkeitseintrages auf die Temperaturverteilung in Gas/Feststoff-Wirbelschichten*. PhD thesis, Otto-von-Guericke-University Magdeburg, 2004.
- [52] R. Hong, L. Hongzhong, C. Maoyu, and Z. Jiyu. Numerical simulation and verification of a gas-solid jet fluidized bed. *Powder Technol.*, 87:73–81, 1996.
- [53] M. Ihlow, J. Drechsler, M. Henneberg, M. Peglow, S. Heinrich, and L. Mörl. Chapter 16: Reaction assisted granulation in fluidized beds. In: *Integrated Chemical Processes - Synthesis, Operation, Analysis, and Control (Eds.: K. Sundmacher, A. Seidel-Morgenstern, A. Kienle)*, WILEY-VCH, pages 453–534, 2005.
- [54] Q.-X. Jiang, E. C. Thrower, D. W. Chester, B. E. Ehrlich, and F. J. Sigworth. Three-dimensional structure of the type 1 inositol 1,4,5-trisphosphate receptor at 24 a° resolution. *EMBO J.*, 21:3575–3581, 2002.
- [55] C. Johnson. *Numerical Approximation of Partial Differential Equations*. Cambridge Press, 1994.
- [56] P. Knabner and L. Angermann. *Numerik partieller Differentialgleichungen*. Springer-Verlag, Berlin, Heidelberg, 2000.
- [57] J. Lang. *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems*, volume 16 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2001.
- [58] J. Lang and J. Verwer. ROS3P - an accurate third-order rosenbrock solver designed for parabolic problems. *BIT*, 41(4):730–737, 2001.

- [59] S. Lang. *Parallele Numerische Simulation instationärer Probleme mit adaptiven Methoden auf unstrukturierten Gittern*. PhD thesis, Dissertation Universität Stuttgart, 2001.
- [60] S. Lang, C. Wieners, and G. Wittum. The application of adaptive parallel multigrid methods to problems in nonlinear solid mechanics. In E. Stein, editor, *Error-Controlled Adaptive Finite Element Methods in Solid Mechanics*, pages 347–384. Wiley, New-York, 2002.
- [61] J. M. Link, L. A. Cuypers, N. G. Deen, and J. A. M. Kuipers. Flow regimes in a spout-fluid bed: a combined experimental and simulation study. *Chem. Eng. Proc.*, 60:57–72, 2005.
- [62] F. Löffler. *Staubabscheiden*. Thieme, Stuttgart, 1988.
- [63] D. D. Mak and J. K. Foskett. Single-channel kinetics, inactivation, and spatial distribution of inositol trisphosphate (IP₃) receptor in *Xenopus* oocyte nucleus. *J. Gen. Physiol.*, 109:571–587, 1997.
- [64] D. D. Mak, S. McBride, V. Raghuram, Y. Yue, S. K. Joseph, and J. K. Foskett. Single-channel properties in endoplasmic reticulum membrane of recombinant type 3 inositol trisphosphate receptor. *J. Gen. Physiol.*, 115:241–255, 2000.
- [65] J. S. Marchant and I. Parker. Role of elementary Ca²⁺ puffs in generating repetitive Ca²⁺ oscillations. *The EMBO Journal*, 20(1 & 2):65–76, 2001.
- [66] L. Massimilla and M. Filla. Analysis of the variables controlling gas jet penetration in fluidized beds. *Ind. Eng. Chem. Fundam.*, 23(1):131–132, 1984.
- [67] L. Massimilla, M. Filla, and S. Vacarro. Gas jets in fluidized beds: The influence of particle size, shape and density on gas and solids entrainment. *Int. J. of Multiphase Flow*, 9:259–267, 1983.
- [68] Ch. Nagaiah, S. Rüdiger, G. Warnecke, and M. Falcke. Adaptive numerical solution of intracellular calcium dynamics using domain decomposition methods. *in preparation*.
- [69] Ch. Nagaiah, S. Rüdiger, G. Warnecke, and M. Falcke. Parallel numerical solution of intracellular calcium dynamics. In *proceedings of 17th International Conference on Domain Decomposition Methods (submitted)*.

- [70] Ch. Nagaiah, G. Warnecke, S. Heinrich, and M. Peglow. Numerical computation of heat and mass transfer in fluidized beds with liquid injection. In *PAMM (accepted)*, 2006.
- [71] Ch. Nagaiah, G. Warnecke, S. Heinrich, and M. Peglow. Numerical simulation of temperature and concentration distributions in fluidized beds with liquid injection. *Chem. Eng. Sci.*, doi: 10.1016/j.ces.2006.11.046, 2006.
- [72] M. T. Nair. *Functional Analysis - A First Course*. Prentice-Hall of India, New Delhi, 2002.
- [73] D. J. Patil, M. van Sint Annaland, and J. A. M. Kuipers. Critical comparison of hydrodynamic models for gas-fluidized beds - part i: bubbling gas-solid fluidized beds operated with a jet. *Chem. Eng. Proc.*, 60:57–72, 2005.
- [74] J. W. Putney and G. S. J. Bird. The inositolphosphate-calcium signaling system in nonexcitable cells. *Endocrine Reviews*, 14(5):610–631, 1993.
- [75] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer Series in Computational Mathematics, 1994.
- [76] E. B. Ridgway, J. C. Gilkey, and L. F. Jaffe. Free calcium increases explosively in activating medaka eggs. *Proc. Natl. Acad. Sci. USA*, 74:623–627, 1977.
- [77] R. Rodriguez. Some remarks on zienkiewicz-zhu estimator. *Numerical Methods for Partial Differential Equations*, 10(5):625–635, 1994.
- [78] T. A. Rooney, E. J. Sass, and A. P. Thomas. Characterization of cytosolic calcium oscillations induced by phenylephrine and vasopressin in single fura-2-loaded hepatocytes. *J. Biol. Chem.*, 264:17131–17141, 1989.
- [79] H. H. Rosenbrock. Some general implicit processes for the numerical solution of differential equations. *Computer J.*, 5:329–330, 1963.
- [80] S. Rüdiger, J. W. Shuai, W. Huisinga, Ch. Nagaiah, G. Warnecke, I. Parker, and M. Falcke. Hybrid stochastic and deterministic simulations of calcium blips. *Biophys. J (accepted)*, 2006.
- [81] E.-U. Schlünder and E. Tsotsas. Wärmeübertragung in Festbetten, durchmischten schüttgütern und Wirbelschichten. *Stuttgart: Thieme Verlag*, 1988.

- [82] B. A. Schmitt and R. Weiner. Matrix-free w-methods using a multiple arnoldi iteration. *Appl. Num. Math.*, 18:307–320, 1995.
- [83] V. V. Shaidurov. *Multigrid Methods for Finite Elements*. Kluwer Academic Publishers, 1995.
- [84] L. Shampine. *Numerical solution of ordinary differential equations*. Chapman & Hall, 1994.
- [85] J. Smoller. *Shock Waves and Reaction-Diffusion Equations*. Springer-Verlag, New York, 1995.
- [86] T. Steihaug and A. Wolfbrandt. An attempt to avoid exact jacobian and nonlinear equations in the numerical solution of stiff differential equations. *Math. Comp.*, 33:521–534, 1979.
- [87] C. W. Taylor. Inositol trisphosphate receptors: Ca^{2+} -modulated intracellular Ca^{2+} channels. *Biochimica and Biophysica Acta*, 1436:19–33, 1998.
- [88] D. Thomas, P. Lipp, M. J. Berridge, and M. D. Bootman. Hormone-evoked elementary Ca^{2+} signals are not stereotypic, but reflect activation of different size channel clusters and variable recruitment of channels within a cluster. *J. Biol. Chem.*, 273(42):27130–27136, 1998.
- [89] R. Thul. *Analysis of intracellular reaction diffusion systems: The stochastic medium Calcium*. PhD thesis, Frei University, Berlin, November 2004.
- [90] R. Thul and M. Falcke. Release currents of IP_3 receptor channel clusters and concentration profiles. *Biophys. J.*, 86:2660–2673, 2004.
- [91] S. C. Tovey, P. de Smet, P. Lipp, K. W. Young, L. Missiaen, H. De Smedt, J. B. Parys, M. J. Berridge, J. Thuring, A. Holmes, and M. D. Bootman. Calcium puffs are generic InsP_3 -activated elementary calcium signals and are downregulated by prolonged hormonal stimulation to inhibit cellular calcium responses. *J. Cell Science*, 114(22):3979–3989, 2001.
- [92] H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13:631–644, 1994.

- [93] R. Verfürth. A review of a posteriori error estimation and adaptive mesh-refinement techniques. *Wiley & Teubner*, 1996.
- [94] P. Wesseling. *An Introduction to Multigrid Methods*. John Wiley, 1992.
- [95] C. Wieners. Parallel linear algebra and the application to multigrid methods. In W. Hackbusch and G. Wittum, editors, *Notes on Numerical Fluid Mechanics*, volume 56. Vieweg Verlag, to appear, 1999.
- [96] K. Wintermantel. Process and product engineering - achievements, present and future challenges. *Chem. Eng. Proc.*, pages 1601–1620, 1999.
- [97] W.-C. Yang and D. L. Keairns. Momentum dissipation and gas entrainment into a gas jet in a fluidized bed. *AIChE Journal*, 77(205):28–36, 1981.
- [98] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Int. J. Num. Meth. Eng*, 24:337–357, 1987.

Curriculum Vitae

Personal details

Name : Chamakuri Nagaiah
Sex : Male
Marital status : Single
Date of birth : 03-06-1979
Place of birth : Maddirala, Nalgonda, A.P
Nationality : Indian

Education and Experience

1996-1999 Bachelor of Science in Mathematics, Physics, Chemistry
S. R & B. G. N. R Govt. Degree college,
Kakatiya University, Khammam, India.

1999-2001 Master of Science in Applied Mathematics
Osmania University, Hyderabad, India.

2001-2003 Master of Technology in
Industrial Mathematics and Scientific Computing
Indian Institute of Technology, Chennai, India.

03/2003- 2007 Ph. D Student in Institute for Analysis and Numerics
Otto-von-Guericke University, Magdeburg, Germany.