Numerical study of drying in porous media

Suresh Kumar Nadupuri

Fakultät für Mathematik

Otto-von-Guericke Universität Magdeburg

Numerical study of drying in porous media

Dissertation
zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)

genehmigt durch die Fakultät für Mathematik
der Otto-von-Guericke-Universität Magdeburg

von **M.Sc., M.Tech. Suresh Kumar Nadupuri**
geb. am 29. August 1974 in Visakhapatnam, India

Gutachter:
Prof. Dr. Gerald Warnecke
Prof. Dr. G.D. Veerappa Gowda

Eingericht am: 08.01.2007
Verteidigung am: 26.02.2007

Acknowledgements

I would like to express my deep and sincere thanks to my supervisor Prof. Gerald Warnecke for giving me oppurtunity to carry out Ph.D studies at University of Magdeburg. His valuable guidance, inspirational help, kind behaviour, continuous support helped me to finish my work successfully. I express my thanks to Prof. E. Tsotsas and Dr. T. Metzger for their valuable suggestions in engineering aspects. I express my deep gratitude to Prof. G.D. Veerappa Gowda for his valuable advices and co-operation.

I would like to thank to Prof. S. Sundar for his constant encouragement. I convey my thanks to all my present and former group members for their timely help, valuable discussions and encouragement. At the institue of analysis and numerics, I experienced a friendly and very good working experience and i convey my thanks to all the members of the institute for their co-operation. My thanks to Piotr skrzypacz for his useful discussions. I like to thank Ch. Nagaiah for his help and discussions. My thanks to Dr. W. Grambow for his technical assistance. Also i would like to thank Prof. A. Bertram and other members of Graduiertenkolleg 828 for organising useful seminars and workshops. I also thank to DFG for providing me financial support for my stay in Germany. I extend my thanks to my friends for their moral support.

Last but not least, I am grateful to my parents and sisters for their best wishes, love, care and continuous encouragement at all stages of my life, no matter how far away we are from each other. I dedicate this work to my beloved parents.

Contents

Chapter 1

Introduction

Efficient discretization strategies in obtaining the numerical solution of time dependent stiff partial as well as ordinary differential equations is important in many practical problems. Adaptive and local time stepping methods are useful in many cases to minimise the computational cost. The present work is focussed on solving a scalar and coupled quasilinear partial differential equations of parabolic type which govern the isothermal drying of porous media [36]. Drying is a complex process of heat and mass transfer playing an important role in different industries. Although the investigation of drying process experimentally, theoretically and numerically has been realised since many years [32, 33, 34, 35, 45, 49], the coupling of heat and mass transfer, the numerical and other aspects are still a challenging problem.

The equations governing the drying process are a complex system which is challenging for the numerical simulation. The numerical difficulties include nonlinearities, strong coupling, solving many variables defined by a nonlinear set of equations of state. Moreover the definitions of equations of state change according to a drying state. Perré and Turner used a control volume finite element technique in solving the equations governing the drying process. We draw our attention in using higher order time stepping methods and study various time stepping strategies in solving the equations governing the drying process at isothermal conditions. We consider a cell-centered finite volume scheme and as a first step we use a fixed time step size. The change of drying states cause steep local gradients due to the rapid increase or decrease in many variables which causes the reduction of time step sizes. In such cases, in order to increase the efficiency, variable time stepping methods have been proven sucessful. For this, we have chosen the method of lines approach and used an implicit fifth order time stepping ODE method, the so called Radau5 method in one space dimension. In Radau5 method, a simplified Newton iterations is used in solving the nonlinear system of equations with variable time step control, for details see the text book of Hairer and Wanner [18]. While moving to higher dimensions, we considered the dimensional splitting approach. To maintain symmetry in the solution and better accuracy, Strang [41, 42] proposed two splitting procedures. We have developed algorithms in the two dimensional case using the dimensional splitting technique which can be extended

easily to three dimensions. Attaining positivity preservation in the numerical solution of partial differential equations is another important factor in many practical problems because many physical variables should become negative. Hence a positivity and stability analysis has been done for a scalar quasilinear parabolic problem.

An efficient time stepping method and a suitable time stepping strategy play an important role in the study of attaining efficiency in the computational cost. Among the methods which are capable of integrating stiff differential equations efficiently, linearly implicit methods have been proven successful because they completely avoid solving nonlinear systems of algebraic equations. If the semi-discrete ODE system is stiff, explicit time integration methods are no longer efficient due to severe time step restrictions that are required for a stable solution. Many implicit methods have better stability properties and therefore are more efficient, despite the fact that large algebraic systems have to be solved at every time step. However, in some cases, the stiffness occurs only in a local domain. The reason for such a behaviour is only that small part of a domain lies in the area of large diffusion. The basic idea is to use an explicit scheme in the nonstiff region and an implicit scheme in the stiff region or in some cases the entire system at a particular time can be switched from implicit to explicit and vice versa. In literature, this way of switching of explicit and implicit schemes are called as partitioning methods, see [1, 8, 10, 15, 28, 19, 48]. We apply a partitioning strategy for a scalar quasilinear parabolic problem. However these schemes are not efficient in some cases where the problem is highly stiff at all parts of the domain. In such cases, local time stepping methods are efficient in which the time step is controlled by a local stability condition. Subunits of many coupled systems typically behave on different time scales. The idea of local time stepping methods is to use small time steps in highly stiff regions and large time steps in nonstiff and mildly stiff regions. By using local time stepping with explicit methods, different time steps are chosen at different grid points usually according to a local CFL condition. By using local time stepping with implicit methods, different time steps are chosen at different grid points based on a local temporal error criterion. The numerical results show that these local time stepping schemes are efficient in saving computational cost, although the schemes are slightly nonconservative at the grid interfaces. In literature these methods are also known as multirate time stepping methods. Multirate timestepping avoids the necessity to take small global timesteps restricted by the maximum error at a particular grid point. The details on such type of methods can be seen in [3, 11, 12, 17, 39, 43]. We consider a time stepping strategy of Savcenco et al. [39] in solving the coupled parabolic problem governing the isothermal drying process in which the time step size is chosen according to the local temporal variation. In this method, the total time interval is divided adaptively into finite subintervals and uses local time steps in each subinterval.

Structure of the thesis

In Chapter 2, a short overview of the drying process of porous media and its applications are presented. The basic concepts behind the modelling of the drying process and the work done on drying during the past are discussed briefly. Then we present a mathematical model of Perré [36] for the drying process of porous media at isothermal conditions. A set of coupled quasilinear parabolic partial differential equations representing water and air balance of the drying process are presented. Next, we give the equations for the state variables which are collected from basic definitions, laws and experiments. Further, we explain the important phenomena in the drying process such as capillary migration, change of drying states and the sorption equilibrium. At the end of this chapter, we give the list of dependent variables and the material constants that are used for the numerical simulation.

In Chapter 3, we are concerned with the numerical simulation of the isothermal drying model in the one dimensional case. First, we introduce a cell centered finite volume method. Then we explain the finite volume discretization for a nonlinear parabolic problem with explicit, semi-implicit and fully implicit time discretizations. Further, the approximation of diffusion coefficients and the derivative terms that appear in the fluxes are presented. Then we discuss the stability conditions for an explicit scheme and give the CFL condition for a parabolic problem. Next, we give Newton's method to solve the nonlinear system of algebraic equations. Also we discuss the construction of the Jacobain matrix numerically and the corresponding functional evaluations required for a numerical Jacobain are presented. Then, we introduce direct and iterative linear solvers which are used to solve linear system of algebraic equations. Next, we pose a quasilinear scalar parabolic problem with initial, boundary conditions and give the complete set of equations of state. This scalar problem is resulting by reducing the coupled drying model of Perré [36] under some assumptions. Next, we give the discretization for the scalar problem. An algorithm for solving the scalar quasilinear parabolic problem using implicit scheme is presented. Then we present the numerical results for the scalar problem and compare the efficiency with explicit, implicit and semi-implicit time discretizations. Next, we pose the coupled problem, which exactly govern the isothemal drying of porous media. The discretization for the coupled problem and an algorithm to solve the quasilinear coupled parabolic problem using fully implicit scheme is presented. The numerical aspects are discussed and the profiles of various variables are given presenting the physical trend of the drying process. Next, we present the method of lines (MOL) approach in solving time dependent partial differential equations. Then we give the semi-discrete DAE system of equations for the coupled problem which are obtained after the space discretization. Next an implicit time integration method, the so called Radau5 with variable time stepping criterion is briefly presented. Finally the numerical aspects with MOL approach using the Radau5 method are given.

Chapter 4 is devoted to the positivity and stability analysis for a quasilinear scalar parabolic problem. First, we consider a quasi-linear diffusion equation. Next, we define an M-matrix and using the properties of M-matrices we prove the positivity and stability for the quasi-

linear initial and boundary value problems with finite difference semi-implicit time discretization. Next we carry out the analysis for a particular case of a quasilinear parabolic problem. Then, we give a simple diffusion equation as a test case for the positivity. Also, we show that the reduced scalar drying problem is a particular case of the analysis of the quasilinear parabolic problem. At the end, we pose the coupled drying problem to mention the difficulty in proving positivity analysis for such complex coupled systems.

Chapter 5 is concerned to the numerical approach in the two dimensional case. At first, the concept of operator splitting is discussed for a scalar ODE problem. Then, we extend the concept of operator splitting to the study of vectors and operators. Then we discuss about the splitting error and the advantages of the splitting approach. Next, we describe the dimensional splitting approach in solving multi dimensional problems. Then we give the discretization in two dimensions using dimensional splitting and present the domain using symmetry. In the next section, two kinds of symmetric splitting procedures of Strang [41, 42] are presented. Finally we present the numerical results in two space dimensions. The saturation and air pressure profiles at different drying times are presented.

In Chapter 6, we study time stepping strategies which are useful in finding an efficient numerical solution of time dependent parabolic partial differential equations. At first, we discuss about linearly implicit methods and its advantages. Then we give two linearly implicit time integration methods with variable time step control which are a W-method and a Rosenbrock method. Next we pose a scalar parabolic problem for test case and the coupled drying problem. In the next section, a partitioning strategy which partitions the time discretization into explicit and implicit parts based on local temporal error is presented. Then, the numerical observations using the partitioning strategy are discussed. In the final section, a brief introduction to the local time stepping methods is given. Then, we present a local time stepping strategy of Savcenco et al. [39] for solving stiff time dependent parabolic partial differential problems. The processing of a time slab, dealing the interface boundaries, estimation of a proper length for a time slab are discussed. Finally the numerical aspects in solving the coupled drying problem using local time stepping strategy are presented.

Chapter 2

Mathematical model: isothermal drying of porous media

A mathematical model governing the drying process in porous media under isothermal conditions is presented in this chapter. In the first section, we give a brief introduction of the drying process and its applications. We consider the drying model developed by Perré [36] for the specific case of isothermal conditions. In Section 2.2, we give the partial differential equations that govern the drying process at the macroscopic level. Then, we give the definitions of all the physical variables that are involved in the macroscopic equations.

2.1 Drying of porous media

Drying involves transfer of heat and mass inside the porous medium. The drying of a porous material, i.e. removal of water from the pores by evaporation, is a complex process. The porous medium under consideration consists of three phases, i.e. a solid phase, a liquid phase consisting of free and sorbed water and a gaseous phase which is a mixture of water vapour and air. The laws for heat and mass transfer in porous media are formulated on the basis of conservation of momentum, energy and mass. The most important transport phenomena are capillary flow, gaseous diffusion, and heat conduction. We take into account the exchanges between them. In the case of isothermal drying, heat transfer is assumed to be perfect so that the heat required for evaporation can be supplied without any temperature gradient. This approximation is reasonable for slow drying under moderate drying conditions. In order to model the drying process, it is necessary to analyse the transport in both the liquid and gas phases through the medium.

The primary objective of understanding the drying process is to predict the distribution of moisture content and internal total gaseous pressure within the capillary porous body as they evolve in time. Mathematical modelling and numerical simulation allow us to avoid expensive and repetitive experiments. Also they can be used to understand the underlying physics associated with the transport phenomena in the drying process. There are two

most important phenomena during the drying process. The first is capillary migration which ensures good moisture transport and high drying rates at the start of the process. The second is the sorption which plays an important role during the final stages of drying when equilibrium is approached. The transport phenomena in porous media can be modelled using a continuous or discrete approach. The discrete approach aims to describe the emptying of individual pores during drying. However, since the geometry of the porous medium is complex, the equations are usually written at the macroscopic scale. At this level, the porous medium is treated as a continuous medium.

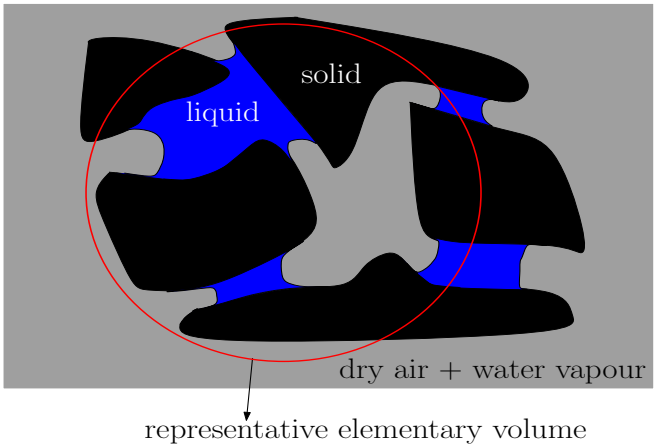


Figure 2.1: Partially saturated porous medium.

Drying has a wide range of applications such as drying of fruits and vegetables, paper and wood, ceramics, concrete, pharmaceuticals etc. There are several types of drying processes, see [16, 35]. We use convective drying. This means that hot dry air flow provides the necessary energy required for evaporation to remove the moisture content. A drying model based on the continuous approach was developed by Whitaker who used the volume averaging technique to derive a system of macroscopic transport equations from a set of basic transport laws at the microscopic level for gas, liquid and solid, see [49, 50, 51]. The theory of Whitaker was later applied to several porous media problems in developing the drying modelling, for instance see [7, 36, 44]. In the present study, we consider a drying model developed by Perré [36] based on Whitaker's theory of drying. We consider an isotropic porous medium at isothermal conditions. The goal of this study is to explore the possibilities of numerical simulation in representing the physics that occur during the transport process in the porous medium in an efficient way. For the reader interested in drying, see [27, 33, 34, 37].

2.2 Governing equations

The equations governing the drying process are highly nonlinear and strongly coupled parabolic partial differential equations. We consider the drying model of Perré [36] at isothermal conditions, i.e. we take temperature as constant. The porous medium is assumed to be isotropic. Then the water and air balance equations are given by

Water balance equation

$$\frac{\partial[\psi S\rho_l + \psi(1-S)\rho_v]}{\partial t} = \nabla\cdot(\rho_g D_{eff}\nabla y_v + \frac{\rho_l K}{\mu_l}k_l\nabla P_l + \frac{K}{\mu_g}\rho_v k_g\nabla P_g). \tag{2.1}$$

Air balance equation

$$\frac{\partial[\psi(1-S)\rho_a]}{\partial t} = \nabla\cdot(\rho_g D_{eff}\nabla y_a + \frac{K}{\mu_g}\rho_a k_g\nabla P_g). \tag{2.2}$$

The primary variables are the liquid saturation S and air density ρ_a. The variables ρ_g, D_{eff}, $y_v, k_l, P_l, \rho_v, k_g, P_g, y_a$ that appear on the right hand side of equations (2.1) and (2.2) are nonlinear functions of S and ρ_a. To make the system complete, these dependent variables are collected from definitions, basic laws and experiments which are explained in the next subsection. Further, the equations contains the constants $\psi, \rho_l, K, \mu_l, \mu_g$. The present problem is challenging for numerical simulations and the numerical aspects will be discussed in next chapters.

Initial and boundary conditions

Initially, the porous medium is assumed to be filled with some prescribed saturation and air pressure. The total gas pressure is at atmospheric pressure

$$S(t_0) = S_0,\ \rho_a(t_0) = \rho_{a0} \text{ and } P_g(t_0) = 10^5\text{Pa}. \tag{2.3}$$

The boundary conditions for the external surfaces of the porous body should be specified. The driving potential at these surfaces is given by

$$J_w\cdot\bar{n} = \beta\frac{P_g}{RT}\tilde{M}_v\ln(\frac{1-\tilde{y}_{vo}}{1-\tilde{y}_v}). \tag{2.4}$$

Here J_w represents the flux vector for the water balance equation and $\bar{n}$ is the outer normal vector. The flux J_w is the amount of water going away from the external boundaries. It was proposed by Bird [5]. Here, we take $\tilde{y}_{vo}$=0, which is the molar fraction of bulk air. The mass transfer coefficient β depends on the air flow field outside the porous medium. The gaseous pressure at the boundaries is always fixed at atmospheric gas pressure and therefore the following algebraic equation serves as the Dirichlet boundary condition for the air balance equation

$$\rho_a = \frac{(100000 - P_v)\tilde{M}_a}{RT}. \tag{2.5}$$

2.2.1 Other equations

At first, we give the state variables to characterize the gas phase. Vapour pressure, air pressure and the total gas pressure are defined by ideal gas laws

$$P_v = \frac{\rho_v \tilde{R} T}{\tilde{M}_v}, P_a = \frac{\rho_a \tilde{R} T}{\tilde{M}_a} \text{ and } P_g = \frac{\rho_g \tilde{R} T}{\tilde{M}_g}. \tag{2.6}$$

The gas phase is a mixture of air and vapour phases and hence the gas pressure is the total pressure, i.e. we have

$$P_g = P_v + P_a, \tag{2.7}$$

and the molar mass of the gas is given by

$$M_g = M_a + (M_v + M_a)\frac{P_v}{P_g}. \tag{2.8}$$

The mass fractions of vapour and air in the gaseous phase are given by

$$y_v = \frac{\rho_v}{\rho_g} \text{ and } y_a = \frac{\rho_a}{\rho_g}, \tag{2.9}$$

and the molar fraction of vapour is the following

$$\tilde{y}_v = \frac{P_v}{P_g}. \tag{2.10}$$

These equations help to understand the gradients of y_v, y_a and P_g in equation (2.1) and equation (2.2).

Capillary migration (∇P_l)

The evaporation starts near or at the surface. In unsaturated porous media, there exist gas and liquid phases. The pressure of the wetting fluid is less than that of the non-wetting phase, due to the curvature of the interface between the liquid and gas. The difference in these pressures is called capillary pressure. The curvature of the meniscus increases as the saturation of the wetting phase decreases. This is the reason for the transfer of liquid from wet to drier parts. This phenomenon is known as capillary migration. These capillary forces are responsible for most of the liquid flow at initial stages of the drying process. The liquid pressure is given by

$$P_l = P_g - P_c. \tag{2.11}$$

The saturation S is proportional to the moisture content X given by

$$S = \frac{\rho_s X}{\rho_l \psi}. \tag{2.12}$$

There is a certain amount of moisture content which is strongly bound to the porous medium which is called sorbed water. The sorbed water is localized, i.e. not free flowing because of the strong interaction with the solid phase. Its value depends on vapour pressure described by sorption isotherm. It has a maximum of critical mosture content X_{crit}. The moisture content for free water is given by

$$X_{fw} = X - X_{crit}, \tag{2.13}$$

and finally the expression for capillary pressure determined by Perre [36] has the following form

$$P_c = 40\sigma e^{(8.4057)10^{-0.3476X_{fw}}}. \tag{2.14}$$

Change of drying states and sorption equilibrium

The important factor in the process of drying is the change of drying states. The definitions of all the variables change according to the drying states. We define the states of drying in the following way throughout the thesis. If the moisture content X is greater than the critical moisture content X_{crit}, this is the first drying state. If X is less than X_{crit}, then this is called the second drying state. And if $X < X_{crit}$ in some part of the porous medium and $X \geq X_{crit}$ in the rest of the porous medium, we call it an intermediate state of the drying process. If $X > X_{crit}$, it means that free water exists, transport is by liquid convection and the gas phase is saturated with vapour. So, there is little space available for gas flow at the start of the drying process. There is no free water if $X < X_{crit}$, the transport is only in the gas phase by diffusion. In this state, there is no space for the liquid flow. Therefore, the relative permeabilities of liquid and gas are defined as

$$k_l = \begin{cases} S_{fw}^3 & : \text{ if } X \geq X_{crit} \\ 0 & : \text{ if } X < X_{crit}, \end{cases} \tag{2.15}$$

and

$$k_g = \begin{cases} 1 + (2S_{fw} - 3)S_{fw}^2 & : \text{ if } X \geq X_{crit} \\ 1 & : \text{ if } X < X_{crit}. \end{cases} \tag{2.16}$$

Here, the free water saturation S_w is taken as

$$S_{fw} = \max\left(0, \frac{X - X_{crit}}{X_{sat} - X_{crit}}\right). \tag{2.17}$$

The value X_{sat} is the saturated moisture content, i.e. the moisture content when all the void space is full of liquid water. The effective diffusivity and binary diffusion coefficient are given by

$$D_{eff} = D_{va}0.2k_g \text{ and } D_{va} = \frac{2.26}{P_g}\left(\frac{T}{273}\right)^{1.81} \tag{2.18}$$

respectively. The sorption equilibrium plays an important role during the change of a drying state. This is responsible for most of the liquid flow at the final stages of the drying

process. It causes a rapid reduction in the vapour pressure at the second stage of the drying process. The relative humidity is defined by

$$\phi = \frac{P_v}{P_v^*}. \tag{2.19}$$

Figure 2.2 represents the sorption isotherm, for the material concrete. It can be described by

$$\phi = \begin{cases} 1 & : \text{ if } X \geq X_{crit} \\ \frac{X}{X_{crit}}[2 - \frac{X}{X_{crit}}] & : \text{ if } X < X_{crit}. \end{cases} \tag{2.20}$$

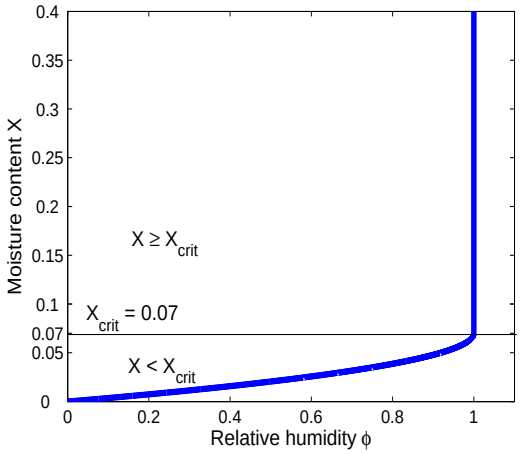


Figure 2.2: Sorption isotherm for light concrete.

It is clear from equations (2.19) and (2.20) that during the first drying state, the vapour pressure is constant and that the sorption causes a sudden rapid reduction in vapour pressure during the second drying state.

2.2.2 Nomenclature

The list of variables

The primary variables are saturation S and density of air ρ_a. The following table gives the list of the dependent variables and the nomenclature

S.No.	symbol	Nomenclature
1	ρ_v	density of vapour
2	ρ_g	density of gas
3	P_a	air pressure
4	P_v	vapour pressure
5	P_l	liquid pressure
6	P_g	gas pressure
7	P_c	capillary pressure
8	y_v	mass fraction of vapour
9	y_a	mass fraction of air
10	$\tilde{M}_g$	molar mass of gas
11	$\tilde{y}_v$	molar fraction of vapour
12	ϕ	relative humidity
13	k_l	relative permeability of liquid
14	k_g	relative permeability of gas
15	X	moisture content
16	X_{fw}	moisture content of free water
17	S_{fw}	saturation of free water
18	D_{eff}	effective diffusivity
19	D_{va}	binary diffusion coefficient of vapour and air

Table 2.1: List of variables.

The list of constants

The following table represent the material constants that are used for the numerical simulation of a concrete material.

S.No.	Symbol	Nomenclature	Value
1	ψ	porosity	0.8
2	ρ_l	density of water	998 $\frac{kg}{m^3}$
3	ρ_s	density of solid	$500\frac{kg}{m^3}$
4	K	absolute permeability	$2.10^{-13}m^2$
5	S_{crit}	critical saturation	0.04383
6	X_{crit}	critical moisture content	0.07
7	X_{sat}	saturated moisture content	1.59
8	R	gas constant	$8314.4Jkmol^{-1}k^{-1}$
9	$\tilde{M}_a$	molar mass of air	28.96 $\frac{kg}{kmol}$
10	$\tilde{M}_v$	molar mass of vapour	$18.02\frac{kg}{kmol}$
11	${p_v}^*$	saturation vapour pressure	2333.9 $pascal$
12	μ_l	viscosity of liquid	$10^{-3}Pa-sec$
13	μ_g	viscosity of gas	$(17.8)10^{-6}Pa-sec$
14	T	temperature	293K
15	σ	surface tension	$0.0726\frac{N}{m}$
16	β	mass transfer coefficient	$0.01\frac{m}{sec}$

Table 2.2: List of constants.

Chapter 3

Numerical simulation in the one dimensional case

In this chapter, the numerical approach and the simulation results for the drying problem in one space dimension are presented. The governing equations are a quasilinear scalar parabolic partial differential equation as well as a coupled parabolic system of partial differential equations. There are several numerical methods to solve the continuous time-dependent partial differential equations such as finite difference methods, finite volume methods and finite element methods. The present simulation work is done by using a cell centered finite volume method. In the first section of this chapter, we describe the finite volume method with explicit, implicit and semi-implicit discretizations. Then, we explain the Newton method to linearize the nonlinear system of equations which are obtained with implicit and semi-implicit time discretizations. In the next section, some methods which we have used to solve the linear system of algebraic equations are presented. In Section 3.3, we give the algorithm and present the numerical results for the quasilinear scalar problem. In Section 3.4, we present the algorithm and the numerical results for the coupled drying problem (isothermal drying). In the final section of this chapter, we describe the method of lines approach and give the corresponding numerical results using a higher order implicit time stepping method which is the so called Radau5 method, see Hairer and Wanner [18].

3.1 Finite volume method

The finite volume method is a numerical method for solving time dependent as well as stationary partial differential equations. It calculates the values of the conserved variables by averaging over a control volume. These methods are of a class of discretization schemes that are highly successful in calculating the solution of a wide range of systems of conservation laws. They are widely used in fluid mechanics, biological models, chemical engineering models, meteorology etc. The primary advantages of this method are intrinsic local conservation properties of the resulting schemes and the numerical robustness through the obtention of discrete maximum principles. Another advantage of the finite volume method

over the finite difference method is that it does not require a structured mesh, although the computations on a structured mesh are simpler. Furthermore, the finite volume method is preferable to the other methods as a result of the fact that the boundary conditions can be applied in accordance to the conservation laws. This is true because the values of the conserved variables are located within the volume element, and not at the nodes or surfaces. The details of these methods can be found in many text books, for instance see Versteeg and Malalasekera [46], LeVeque [24], Patankar [30].

The finite volume method consists of the following key steps

1. Division of the domain under consideration into a finite set of control volumes which is called the grid generation.

2. Converting the governing partial differential equations into integral form and then integrating them over all the control volumes of the solution domain.

3. Calculating the fluxes by the approximation of convective, diffusive and source terms in the governing equations.

4. Finally, the system of algebraic equations obtained in Step 3 has to be solved by either direct or by iterative methods.

Figure 3.1 represents a typical cell centered control volume mesh in one space dimension. Let $h = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ be the length of each control volume. The circles denote the location of the center of the control volume or cell, where $j - \frac{1}{2}$ and $j + \frac{1}{2}$ represent the cell faces or edges. The boundary cells can be handled in several ways. Here we take the length of the boundary control volume as $h/2$. For treating the boundary cells in different ways, we refer to Patankar [30].

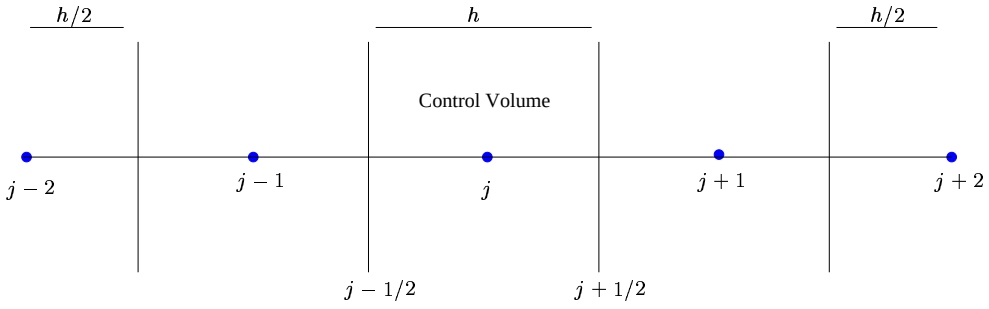


Figure 3.1: Typical control volume mesh in 1-d.

3.1.1 Finite volume discretization

Let $u = u(x,t);\ x,t \in \mathbb{R},\ t \geq 0$. Consider the following partial differential equation as a model problem for which we present the cell-centered finite volume discretization,

$$\frac{\partial \theta(u)}{\partial t} = \frac{\partial}{\partial x}\left[\sum_{i=1}^{k}\left(a_i(u)\frac{\partial f_i(u)}{\partial x}\right)\right]. \tag{3.1}$$

Consider the one-dimensional control volume which is shown in the Figure 3.1. Integrate the partial differential equation (3.1) over the control volume from $x_{j-\frac{1}{2}}$ to $x_{j+\frac{1}{2}}$ and over a time interval of t_n to t_{n+1}, then we have

$$\int_{t_n}^{t_{n+1}}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}}\frac{\partial \theta(u)}{\partial t}dxdt = \int_{t_n}^{t_{n+1}}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}}\frac{\partial}{\partial x}\left[\sum_{i=1}^{k}\left(a_i(u)\frac{\partial f_i(u)}{\partial x}\right)\right]dxdt,$$

Let $\tau{=}t_{n+1} - t_n$ be the time step size. Then we get,

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}}\left[\int_{t_n}^{t_{n+1}}\frac{\partial \theta(u)}{\partial t}dt\right]dx = \tau\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}}\frac{\partial}{\partial x}\left[\sum_{i=1}^{k}\left(a_i(u)\frac{\partial f_i(u)}{\partial x}\right)\right]^{[n+1]}dx + o(\tau).$$

Definition 3.1 *Let $\varphi : [t_n, t_n+\tau] \to \mathbb{R}$ be any function with $t_n{\geq}0$ and $\tau{>}0$. Then for an explicit scheme we define $\varphi(t)^{[n+1]} := \varphi(t_n)$ and for an implicit scheme we define $\varphi(t)^{[n+1]} := \varphi(t_n+\tau)$.*

For a semi-implicit scheme, we take the diffusion coefficients a_i in the equation (3.1) explicitly and the derivative terms implicitly. Let $h = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ be the space step size, then by omiting the error terms we have

$$\frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}}\left[\theta(u)^{n+1} - \theta(u)^n\right]dx = \frac{\tau}{h}\left[\sum_{i=1}^{k}\left(a_i(u)\frac{\partial f_i(u)}{\partial x}\right)\right]_{x_{j+\frac{1}{2}}}^{[n+1]} - \frac{\tau}{h}\left[\sum_{i=1}^{k}\left(a_i(u)\frac{\partial f_i(u)}{\partial x}\right)\right]_{x_{j-\frac{1}{2}}}^{[n+1]}. \tag{3.2}$$

We define

$$\boldsymbol{\theta}(u)_j^{n+1} = \frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}}\theta(u)^{n+1}dx \text{ and } \boldsymbol{\theta}(u)_j^{n} = \frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}}\theta(u)^{n}dx,$$

the averaged values accross a control volume. Finally, we obtain

$$\boldsymbol{\theta}(u)_j^{n+1} - \boldsymbol{\theta}(u)_j^{n} = \frac{\tau}{h}\left[\sum_{i=1}^{k}\left(a_i(u)\frac{\partial f_i(u)}{\partial x}\right)\right]_{x_{j+\frac{1}{2}}}^{[n+1]} - \frac{\tau}{h}\left[\sum_{i=1}^{k}\left(a_i(u)\frac{\partial f_i(u)}{\partial x}\right)\right]_{x_{j-\frac{1}{2}}}^{[n+1]}. \tag{3.3}$$

The two terms in the right hand side of equation (3.3) are called the flux terms at the right and left faces of a cell.

Approximating the diffusion coefficients and derivatives in the fluxes

The diffusion coefficients at the cell faces $j-\frac{1}{2}$ and $j+\frac{1}{2}$ in Figure 3.1 are approximated by taking the average of u over the adjacent cells i.e.,

$$a(u)_{j-\frac{1}{2}} = \left(\frac{a(u)_j + a(u)_{j-1}}{2}\right) \text{ and } a(u)_{j+\frac{1}{2}} = \left(\frac{a(u)_j + a(u)_{j+1}}{2}\right). \tag{3.4}$$

It is also possible to approximate the diffusion coefficients with a higher order averaging, for instance a fourth order averaging is given by equation (3.5), see Kurganov [22]

$$a(u)_{j-\frac{1}{2}} = \frac{-a(u)_{j+2} + 9a(u)_{j+1} + 9a(u)_j - a(u)_{j-1}}{16}. \tag{3.5}$$

The derivatives appearing in the flux terms in equation (3.3), at the cell faces $j-\frac{1}{2}$ and $j+\frac{1}{2}$ are approximated using the central differences. At the cell faces, we have

$$\left[\frac{\partial f(u)}{\partial x}\right]_{j-\frac{1}{2}} = \frac{f(u)_j - f(u)_{j-1}}{h} \text{ and } \left[\frac{\partial f(u)}{\partial x}\right]_{j+\frac{1}{2}} = \frac{f(u)_{j+1} - f(u)_j}{h}.$$

Explicit discretization

The explicit time discretization of the partial differential equations are easy to implement, since these discretizations do not lead to any nonlinear system of algebraic equations. Though these methods are accurate, they are poor in efficiency. The stability or instability of an explicit discretization depends on the mesh ratio. Courant, Friedrich and Lewy derived a stability condition which is the so called CFL condition for the convergence of an explicit difference scheme in terms of the concept of a domain of dependence. For more details on stability and CFL condition, we refer to Morton [26].
Let us consider the equation

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

Now suppose that, we discretize the equation (3.6) by a cell centered finite volume scheme which is explained for equation (3.1), then the CFL condition for (3.6) becomes

$$\max(a(u))\frac{\tau}{h^2} \leq \frac{1}{2}. \tag{3.7}$$

As mesh size decreases i.e., as $h \to 0$, the time step sizes τ are reduced in order to satisfy the inequality (3.7).

Implicit discretization

Implicit discretization methods are in general unconditionally stable schemes. These methods are stable with large time steps and therefore more efficient, but less accurate when compared to explicit discretization. Implicit discretizations of partial differential equations

lead to a large system of algebraic equations which can be linear or nonlinear. This system of equations has to be solved by iterative or direct methods which will be presented in the next section. For solving the nonlinear system of algebraic equations, we first use Newton's method to linearize the system and then we apply the direct or iterative methods to solve the linearized system.

Newton's method

We want to solve a nonlinear system of algebraic equations given in the form $\mathbf{F}(\mathbf{u})=0$, where $\mathbf{u}$ is the unknown vector. Newton's method is an iterative method which is used to solve the nonlinear system of algebraic equations in terms of transition from a current iterate $\mathbf{u_n}$ to a new iterate $\mathbf{u_{n+1}}$. The Newton method is given by

$$\begin{aligned} \mathbf{u_{n+1}} &= \mathbf{u_n} - \mathbf{z}, \\ \mathbf{z} &= \left(\frac{\partial \mathbf{F}}{\partial \mathbf{u}}\right)^{-1} \mathbf{F}(\mathbf{u}). \end{aligned} \tag{3.8}$$

Here $\frac{\partial \mathbf{F}}{\partial \mathbf{u}}$ is the Jacobian matrix. A proper initial guess of the solution $\mathbf{u}$ is necessary for the convergence to the exact root. In order to compute the solution at a new state from the current iterate, one must evaluate the Jacobian matrix and define an accuracy criterion to terminate the iterative process (3.8). The iterative process is terminated when the difference in the two consecutive iterations $\mathbf{u_n}$ and $\mathbf{u_{n+1}}$, is less than the prescribed tolerance. The construction of the Jacobian may be exact if it is easy to compute the derivatives analytically, otherwise the Jacobian can be approximated by using the finite differences. Evaluation of $\frac{\partial \mathbf{F}}{\partial \mathbf{u}}$ by using finite differences should be expected to cost N times the cost of an evaluation of $\mathbf{F}$ because each column in $\frac{\partial \mathbf{F}}{\partial \mathbf{u}}$ requires an evaluation of $\mathbf{F}$ to form the difference approximation. Thus if one uses a direct method to solve the linear system of algebraic equations $\left(\frac{\partial \mathbf{F}}{\partial \mathbf{u}}\right)\mathbf{z} = \mathbf{F}(\mathbf{u})$ and finite differences for the construction of Jacobian, then the cost of a Newton's step can be roughly estimated as N+1 evaluations of the function $\mathbf{F}(\mathbf{u})$ plus the number of floating-point operations that occur in solving the linear system $\left(\frac{\partial \mathbf{F}}{\partial \mathbf{u}}\right)\mathbf{z} = \mathbf{F}(\mathbf{u})$. For rigourous information about the Newton method and its properties, we refer to Deuflhard [14] and Kelley [21].

3.2 Methods to solve the linear system of equations

There exist several methods to solve the linear system of algebraic equations, which can be direct methods or iterative methods. In our computational work, we use four methods to solve the linear algebraic systems. Let $\mathbf{Az} = \mathbf{b}$ represent the system of algebraic equations, where $\mathbf{A}$ is an $N \times N$ matrix, $\mathbf{z}$ is the unknown vector and $\mathbf{b}$ is the known right hand side vector.

Gauss elimination method

This is a well known method that performs the elementary row operations to put the augmented matrix into the upper triangular form. Then, one can easily solve the unknown vector $\mathbf{z}$ using back substitution. This method is found in many text books, for details of the method and algorithms, we refer to William [38]. The innermost loops of Gauss-Jordan elimination are executed N^3 times, the corresponding loops in Gaussian elimination are executed only $N^3/3$ times.

LU decomposition

Gauss elimination leads to the decomposition of the matrix $\mathbf{A}$ as a product of two matrices,

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{U},$$

where $\mathbf{L}$ is a lower triangular matrix which has elements only on the diagonal and below and $\mathbf{U}$ is upper triangular which has elements only on the diagonal and above. Then we have

$$\mathbf{Az} = (\mathbf{L} \cdot \mathbf{U})\mathbf{z} = \mathbf{L} \cdot (\mathbf{U} \cdot \mathbf{z}) = \mathbf{b}.$$

The solution $\mathbf{z}$ is obtained first by solving $\mathbf{Ly} = \mathbf{b}$ by forward substitution and then by solving $\mathbf{Uz} = \mathbf{y}$ by backward substitution. This is also a well-known method and can be found in many text books of linear algebra and numerics.

Thomas algorithm

In most of the cases, we consider the matrix $\mathbf{A}$ is a tri-diagonal matrix. Especially the discretization of a scalar partial differential equation in one space dimension leads to a system of algebraic equations $\mathbf{Az} = \mathbf{b}$, where the matrix $\mathbf{A}$ is a tri-diagonal matrix. This algorithm is easily and efficiently carried out by a special variant of the well-known Gaussian elimination algorithm. We use the Thomas algorithm to solve the linear algebraic system of equations obtained while solving the scalar drying problem in one dimension. This algorithm can be found in many text books, so it is not presented here, we refer to Morton [26], Weickert [47]. This method is stable for every strictly diagonally dominant matrix and efficient in comparison to the Gauss-elimination method. This procedure requires $5N$-4 multiplications/divisions and $3N$-3 substraction. Hence the CPU effort is linear in N.

BiCGSTAB

In many situations iterative methods are preferred over direct methods because an accurate approximation to the solution is obtained with less computational effort. The biconjugate gradient stabilized method is an iterative method developed to solve the algebraic system of equations where the matrix $\mathbf{A}$ need not be positive definite and symmetric whereas in the conjugate gradient method, the matrix $\mathbf{A}$ should be symmetric and positive definite. In our numerical work, we use BiCGSTAB without and also with preconditioning. BiCGSTAB

requires two matrix-vector products and four inner products, for details of this method see Van der Vorst [13].

3.3 Numerical results - scalar problem

In this section, we give the numerical approach and the corresponding numerical results for a quasilinear scalar parabolic problem. The scalar problem results from a simplification of the drying problem of Perré under some assumptions in order to start with a slightly simpler case and later we extend it to the complicated coupled problem. In this section, we carry the numerical results with explicit, implicit and semi-implicit time discretizations using a cell-centered finite volume method and present the corresponding drying profiles. The numerical simulations are carried out for the drying of an isotropic porous medium. We consider the drying of concrete.

3.3.1 One dimensional scalar problem

First we give the governing partial differential equation and the equations of state for the scalar problem. We consider only the water balance equation from the drying model of Perré [36] and we assume that the density of air ρ_a and the temperature T as constants. The primary variable is the saturation S. The governing equation is a quasilinear parabolic partial differential equation. Though this reduced problem might slightly affect the physics of the problem, we wish to start with a slightly simpler case because the coupled problem is highly complex. Also we wish to analyze such type of a fully quasilinear problem. The reduced problem under consideration consists of 18 secondary variables.

Governing equation

The governing partial differential equation is given by

$$\frac{\partial[\psi S\rho_l + \psi(1-S)\rho_v]}{\partial t} = \frac{\partial}{\partial x}(\rho_g D_{eff}\frac{\partial}{\partial x}y_v + \frac{\rho_l K}{\mu_l}k_l\frac{\partial}{\partial x}P_l + \frac{K}{\mu_g}\rho_v k_g\frac{\partial}{\partial x}P_g). \tag{3.9}$$

Initial and boundary conditions

Initially the porous medium is saturated, i.e., the porous medium consists of certain amount of moisture content, which is given by $S_0 = 0.8$ and the pressure of the gas P_g is 10^5 pa. Let the length of the sample be L. Hence the boundary conditions at the external ends 0 and L of the domain must be specified. They are given by a nonlinear flux which is a function of the saturation S. The boundary conditions are symmetric. This flux function has the following form

$$J_w \cdot \bar{n} = \beta\frac{P_g}{RT}M_v \ln(\frac{1}{1-\tilde{y}_v}), \tag{3.10}$$

where $\bar{n}$ is the outward unit normal vector and J_w is the boundary flux.

3.3.2 Equations of state

To make the system complete, the equations of state are collected from basic laws, experiments and definitions. These equations are given as follows,

1. $y_v = \frac{\rho_v}{\rho_g}$,
2. $y_a = \frac{\rho_a}{\rho_g}$,
3. $\tilde{y}_v = \frac{P_v}{P_g}$,
4. $\tilde{M}_g = \tilde{M}_a + (\tilde{M}_v - \tilde{M}_a)\frac{P_v}{P_g}$,
5. $D_{eff} = D_{va} 0.2 k_g$,
6. $D_{va} = \frac{(2.26)}{P_g}(\frac{T}{273})^{1.81}$,
7. $S = \frac{\rho_s X}{\rho_l \psi}$,
8. $\phi = \frac{P_v}{P_v^*}$,
9. X_{fw}=$X - X_{crit}$,
10. $P_c = 40\sigma e^{(8.4057)10^{-0.3476 X_{fw}}}$,
11. $P_l = P_g - P_c$,
12. $P_g = P_v + P_a$,
13. $P_g = \frac{\rho_g \tilde{R} T}{\tilde{M}_g}$,
14. $P_v = \frac{\rho_v \tilde{R} T}{\tilde{M}_v}$.
15. $S_{fw} = \max\left(0, \frac{X - X_{crit}}{X_{sat} - X_{crit}}\right)$.

Change of drying states

The variables relative humidity ϕ, permeability of water k_l and permeability of gas k_g are defined with respect to the drying states, see Chapter 2. So, during the intermediate stage of drying, the nodes which are in the first drying state are governed by some equations of state and the remaining nodes which are in the second drying state are governed by another set of equations. Thus the definitions of all the 18 variables change according to a drying state. The relative humidity and permeabilities are governed by the following equations

1. $\phi = [\frac{X}{X_{crit}}][2 - \frac{X}{X_{crit}}]$ if $X \leq X_{crit}$ else $\phi = 1$,
2. $k_l = S_{fw}^3$ if $X \geq X_{crit}$ else $k_l = 0$ and
3. $k_g = 1 + (2S_{fw} - 3)S_{fw}^2$ if $X \geq X_{crit}$ else $k_g = 1$.

3.3.3 Complexity of the problem

The drying problem under consideration, the scalar as well as the coupled problem is challenging for numerical simulations as it poses a lot of difficulties. Some of them are listed here.

1. Fully nonlinearity,
2. many variables which leads to many equations of state,
3. large real time computations, for instance hours, days or months,
4. local rapid increase and decrease of variables due to different states of drying,
5. strongly coupling, the coupled problem will be discussed in the next section.

3.3.4 Discretization

We discretize the partial differential equation (3.9) using a cell centered finite volume method which is presented in Section 3.1. We consider a one dimensional mesh as shown in Figure 3.1. The diffusion coefficients which appear in the fluxes are averaged over the cell faces, and the derivatives appearing in the fluxes are approximated by central differences. We integrate the equation (3.9) over a control volume $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}] \times [t_n, t_{n+1}]$, then we have

$$\int_{t_n}^{t_{n+1}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \frac{\partial[\psi S\rho_l + \psi(1-S)\rho_v]}{\partial t} dxdt = \int_{t_n}^{t_{n+1}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \frac{\partial}{\partial x}(\rho_g D_{eff} \frac{\partial}{\partial x} y_v + \frac{\rho_l K}{\mu_l} k_l \frac{\partial}{\partial x} P_l + \frac{K}{\mu_g} \rho_v k_g \frac{\partial}{\partial x} P_g) dxdt.$$

Let $h = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ be the space step size and N be the number of cells, i.e. $h = \frac{L}{N}$, where L is the length of the domain. The length of the boundary cells is $\frac{h}{2}$. To explain the discretization we use a simpler notation for the diffusion coefficients in (3.9). We denote the diffusion coefficients for the scalar problem (3.9) by

$$a_1(S) = \rho_g D_{eff},\ a_2(S) = \frac{\rho_l K}{\mu_l} k_l \text{ and } a_3(S) = \frac{K}{\mu_g} \rho_v k_g. \tag{3.11}$$

Then, for $1 \leq j \leq N-1$ we obtain the following nonlinear algebraic equation

$$\begin{aligned} [\psi S\rho_l + \psi(1-S)\rho_v]_j^{n+1} - [\psi S\rho_l + \psi(1-S)\rho_v]_j^n = \\ \frac{\tau}{h}[a_1(S)\frac{\partial}{\partial x} y_v + a_2(S)\frac{\partial}{\partial x} P_l + a_3(S)\frac{\partial}{\partial x} P_g]_{j+\frac{1}{2}}^{[n+1]} - \\ \frac{\tau}{h}[a_1(S)\frac{\partial}{\partial x} y_v + a_2(S)\frac{\partial}{\partial x} P_l + a_3(S)\frac{\partial}{\partial x} P_g]_{j-\frac{1}{2}}^{[n+1]}, \end{aligned} \tag{3.12}$$

at the left boundary $j = 0$, we get

$$\begin{aligned} [\psi S\rho_l + \psi(1-S)\rho_v]_0^{n+1} - [\psi S\rho_l + \psi(1-S)\rho_v]_0^n = \\ \frac{2\tau}{h}\left([a_1(S)\frac{\partial}{\partial x} y_v + a_2(S)\frac{\partial}{\partial x} P_l + a_3(S)\frac{\partial}{\partial x} P_g]_{\frac{1}{2}}^{[n+1]} - (J_w)_0\right), \end{aligned} \tag{3.13}$$

and at the right boundary $j = N$, we obtain

$$[\psi S\rho_l+\psi(1-S)\rho_v]_N^{n+1} - [\psi S\rho_l + \psi(1-S)\rho_v]_N^n = \frac{2\tau}{h}\left(-(J_w)_N - [a_1(S)\frac{\partial}{\partial x}y_v + a_2(S)\frac{\partial}{\partial x}P_l + a_3(S)\frac{\partial}{\partial x}P_g]_{N-\frac{1}{2}}^{[n+1]}\right). \tag{3.14}$$

Implicit and semi-implicit schemes lead to a large system of nonlinear algebraic equations which can be linear or nonlinear. In the present problem, the system of algebraic equations (3.12), (3.13) and (3.14) is nonlinear. We therefore use the Newton method to linearise the nonlinear system of equations and then we apply Thomas algorithm for solving the linearized system. If we use an explicit scheme, we can avoid solving the large nonlinear algebraic systems, but these schemes demand very small time step sizes for stability restrictions. In the present problem, even an explicit scheme demands the Newton iteration to find out the solution S, since the time dependent term is a nonlinear function of saturation S which is given by

$$[\psi S\rho_l + \psi(1-S)\rho_v] = \begin{cases} (A-B)S + B & : \text{ if } S \geq S_{crit} \\ A_1S^3 + B_1S^2 + C_1S & : \text{ if } S < S_{crit}. \end{cases} \tag{3.15}$$

The constants A, B, A_1, B_1 and C_1 are given by

$$A = \psi\rho_l,\ B = 0.013811, A_1 = \frac{\psi^3\rho_l^2 M_v P_v^*}{RTX_{crit}^2\rho_s^2}, B_1 = -A_1 - \frac{2A_1X_{crit}\rho_s}{\psi\rho_l}, C_1 = \psi\rho_l - A_1 - B_1.$$

The material constants involved in the above equations can be obtained from the Table 2.2. The efficiency of the explicit scheme compared with the efficiency of an implicit scheme can be observed from the Table 3.1.

3.3.5 CFL condition for the explicit scheme

As described for the equation (3.6), a necessary condition is required for the problem (3.9) to attain the stability which is called the CFL Condition. For the problem (3.9), in order to obtain the stability of $[\psi S\rho_l + \psi(1-S)\rho_v]$, we consider the following restriction for the time step size using an explicit scheme

$$\max\left[a_1(S), a_2(S), a_3(S)\right]\frac{\tau}{h^2} \leq \text{CFL}. \tag{3.16}$$

3.3.6 Construction of Jacobian matrix for the implicit scheme

To solve the nonlinear system of algebraic equations, Newton's method is the best choice which requires the construction of a Jacobian matrix. Though analytical Jacobian is more accurate than the finite difference Jacobian, due to many equations of state, highly nonlinearity and due to the switching of the drying states, the construction of the Jacobian matrix analytically is very complicated. So, we use a finite difference approximation to

construct the Jacobian matrix numerically. Bringing the right hand side of the equations (3.12), (3.13) and (3.14) to the left hand side, we denote this system of algebraic equations by

$$F_j(S_0, S_1, ..., S_N) = 0, \text{ for } j = 0, 1, ...N. \tag{3.17}$$

Then for any $\epsilon > 0$, the derivates in the jacobian matrix are estimated by

$$\frac{\partial F_j(S_0, S_1, ..., S_N)}{\partial S_i} = \frac{F_j(S_0, S_1, ..., S_i + \epsilon, ..., S_N) - F_j(S_0, S_1, ..., S_i, ..., S_N)}{\epsilon}, \tag{3.18}$$

for $j = 0, 1, ...N$ and for $i = j - 1, j, j + 1$. The Jacobian matrix obtained for this scalar problem (3.9) is a tri-diagonal matrix. Therefore, we use the Thomas algorithm to solve the linearized system of equations. The convergence of the Newton's iterations depends on the suitable choice of ϵ. The choice of large ϵ avoids the roundoff errors and a small ϵ avoids poor approximation of the derivative. A proper choice of ϵ gives faster convergence. We have taken ϵ=10^{-6} in our numerical computations.

3.3.7 Algorithm

Here we present the algorithm, which describes the implementation of the fully implicit scheme to solve the problem (3.9). Let $\mathbf{S}$ denote the vector $[S_0\ S_1 .... \ S_N]$, where N is the number of cells, $\mathbf{F(S)} = [F_0(\mathbf{S})\ F_1(\mathbf{S})\ ...F_N(\mathbf{S})]$ denotes the vector of algebraic functions and $\mathbf{J_c}$ denotes the Jacobian matrix. Then the algorithm reads

1. Initialize the 19 variables.
2. Do the time loop.
3. Start the Newton iterations
   - initial guess $\mathbf{S}^{n+1} = \mathbf{S}^n$,
   - initialize the nonlinear function set $F_j(\mathbf{S}^{n+1})$=0 for $j = 0, 1, 2, ..., N$,
   - compute the increment $\zeta S_j^{n+1} = S_j^n + \epsilon$, for any small $\epsilon > 0$ and compute the increments in all the 18 variables for this ζS_j,
   - approximate the derivatives in the Jacobian
   $$\frac{dF_j}{dS_i} = \frac{F_j(\zeta S_i) - F_j(S_i)}{\epsilon} \text{ for } i = j - 1, j, j + 1,$$
   - assemble the Jacobian matrix $\mathbf{J_c}$,
   - compute $(\mathbf{J_c})^{-1}\mathbf{F(S)}$ using Thomas algorithm for tridiagonal matrix
   - compute $\mathbf{S}^{n+1} = \mathbf{S}^n - \mathbf{J_c}^{-1}\mathbf{F(S)}$,
   - update the remaining 18 variables,
   - check if $(S_j^{n+1} - S_j^n) >$ tolerance for all $j = 0, 1, ..., N$,

- stop the Newton iterations if the desired tolerance is reached.

4. Check for the desired average moisture content.

5. End of the time loop when the drying is complete.

3.3.8 Simulation results

In this section, we present the numerical results for the scalar drying problem in one space dimension. We consider a concrete material of length 0.05m. We take $h = \frac{0.05}{N}$ as the fixed space step size, where N denotes the number of cells. First we start with an explicit time discretization, since an explicit discretization is easier to implement. But the time step sizes for the explicit schemes are very small. They are controlled by a CFL condition as defined in (3.16). The time step sizes with semi-implicit and implicit schemes are larger and fixed with $\tau = 1000$ during the first drying state and $\tau = 50$ during the intermediate and second drying states. A smaller time step is needed during the intermediate drying state. This reduction in the time step is due to the change of drying states. Because the change of a drying state changes the definition of the variables, and there is a rapid reduction in some variables and a rapid increase in other variables. The change of a drying state causes steep gradients at the neighbourhood of a node where the change of a drying state take place. These factors make the Jacobian matrix ill-conditioned with the larger time steps after the first drying state. Thus for larger time steps, the Newton's method is not convergent. The time step selection here is just based on trail and error. Off course in such cases, a proper time step selection strategy is useful to increase the efficiency of the code which we will discuss in the last section of this chapter. Figure 3.2 shows the comparison plot of the solution (saturation S) with explicit, implicit and semi-implicit discretizations after a real time of 4.62 days. The initial saturation S_0 is taken as 0.8 and at the end of 4.62 days, the average saturation reaches to 0.000025. The right hand side figure of Figure 3.2 is the zoom of the left hand side figure. Figure 3.3 represents the average saturation profiles with different mesh sizes for a drying period of 3.47 days. Figure 3.3 shows the convergence of the solution as the mesh size decreases. The right hand side figure of Figure 3.3 is the zoom of the left hand side figure.

The following table represents the CPU times using a 200 cell mesh, taken by explicit, semi-implicit and implicit discretizations.

No.	Scheme	Total Newton iterations	CPU time (min)
1	explicit	—-	42.73
2	semi-implicit	27885	9.83
3	implicit	29044	11.99

Table 3.1: Efficiency table for the scalar problem.

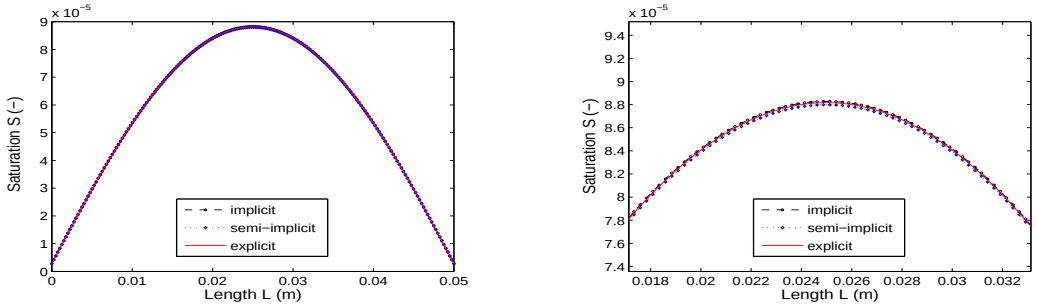


Figure 3.2: Saturation profile with explicit, semi-implicit and implicit discretizations.

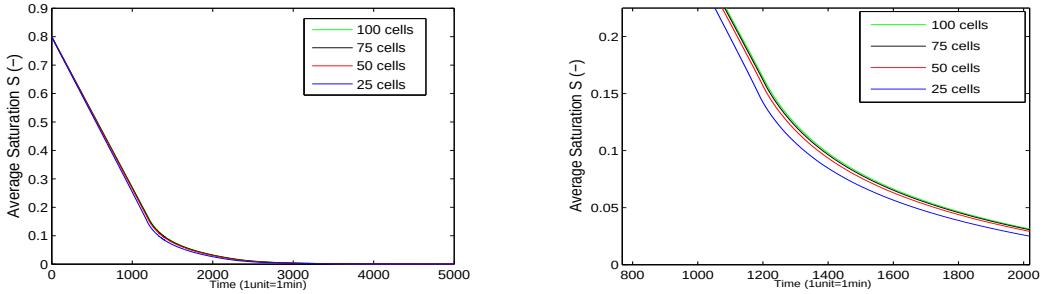


Figure 3.3: Convergence of the solution as the mesh size decreases.

3.4 Numerical results - coupled problem

This section presents the numerical results for the coupled drying problem. We consider the drying of an isotropic porous medium at isothermal conditions. We assume the temperature as constant in the drying model of Perré [36].

3.4.1 Coupled problem in one dimension

The governing equations are two coupled parabolic partial differential equations for the water and air balance. The primary variables are saturation S and air density ρ_a. In one dimension, the governing equations for water balance and air balance are given by the following equations

$$\frac{\partial[\psi S\rho_l + \psi(1-S)\rho_v]}{\partial t} = \frac{\partial}{\partial x}(\rho_g D_{eff}\frac{\partial}{\partial x}y_v + \frac{\rho_l K}{\mu_l}k_l\frac{\partial}{\partial x}P_l + \frac{K}{\mu_g}\rho_v k_g\frac{\partial}{\partial x}P_g), \tag{3.19}$$

$$\frac{\partial[\psi(1-S)\rho_a]}{\partial t} = \frac{\partial}{\partial x}(\rho_g D_{eff}\frac{\partial}{\partial x}y_a + \frac{K}{\mu_g}\rho_a k_g \frac{\partial}{\partial x}P_g). \tag{3.20}$$

These two equations are highly nonlinear and strongly coupled. The variables ρ_v, ρ_g, D_{eff}, y_v, y_a, k_l, k_g, P_l and P_g that appear in (3.19) and (3.20) are unknowns and these variables are nonlinear functions of S and ρ_a. All these variables are defined in equations of state.

Initial and boundary conditions

The porous medium is filled with certain amount of initial concentration at atmospheric gas conditions. The initial conditions are given by

$$S_0 = 0.8 \text{ and } \rho_a = 1.16103 \ \frac{\text{kg}}{\text{m}^3}.$$

The boundary condition for the water balance equation (3.19) is a nonlinear flux which is given by (3.10). The boundary condition for the air balance equation (3.20) is a Dirichlet condition. This Dirichlet condition for ρ_a can be obtained by solving the nonlinear algebraic equation which is given by

$$\rho_a = \frac{(P_g - P_v)M_a}{RT} \quad \text{on } \partial\Omega. \tag{3.21}$$

At each time level, the nonlinear equation (3.21) should be evaluated along with the nonlinear algebraic system of equations obtained after the discretization of the partial differential equations (3.19) and (3.20). Also at the boundary, the gas pressure P_g is always kept at atmospheric gas conditions i.e.,

$$P_g = 10^5 \text{ Pa on } \partial\Omega.$$

Equations of state

The coupled problem consist of two primary variables saturation S, density of air ρ_a and 19 secondary variables. The 18 equations of state are the same as given in Section 3.3 for the scalar problem. Additionally, we have the following equation of state

$$P_a = \frac{\rho_a \tilde{R} T}{\tilde{M}_a}.$$

3.4.2 Discretization and algorithm

We use a cell-centered finite volume scheme which is explained in Section 3.1. As a first step, we use an explicit discretization in time which is easy to implement. It is well known that explicit schemes demand a very small time step size for the stability restrictions. In the coupled case, the explicit scheme is very expensive as it demands a very small time step size unlike the scalar problem. The time step demanded by an explicit scheme is

approximately 1000 times smaller than that of an implicit scheme for the coupled problem. Hence, we do not consider an explicit discretization for the coupled case.

We use the following notation for diffusion coefficients in the coupled problem. The diffusion coefficients in (3.19) are denoted by

$$a_1(S,\rho_a) = \rho_g D_{eff},\ a_2(S) = \frac{\rho_l K}{\mu_l} k_l \text{ and } a_3(S) = \frac{K}{\mu_g}\rho_v k_g.$$

and the diffusion coefficents in air balance equation (3.20) are denoted by

$$b_1(S,\rho_a) = \rho_g D_{eff} \text{ and } b_2(S,\rho_a) = \frac{K}{\mu_g}\rho_a k_g.$$

The discretization of the water balance equation (3.19) is given by (3.12), (3.13) and (3.14) with $a_1(S)$ replaced by $a_1(S,\rho_a)$, where we consider only fully implicit discretization in time. Here we give the discretization of the air balance equation (3.20). For $1 \le i \le N-1$, after discretizing the equation (3.20), we obtain

$$\begin{aligned}[\psi(1-S)\rho_a]_j^{n+1} - [\psi(1-S)\rho_a]_j^n = \frac{\tau}{h}[b_1(S,\rho_a)\frac{\partial}{\partial x}y_a + b_2(S,\rho_a)\frac{\partial}{\partial x}P_g]_{j+\frac{1}{2}}^{[n+1]} \\ -\frac{\tau}{h}[b_1(S,\rho_a)\frac{\partial}{\partial x}y_a + b_2(S,\rho_a)\frac{\partial}{\partial x}P_g]_{j-\frac{1}{2}}^{[n+1]},\end{aligned} \tag{3.22}$$

at the boundary nodes i=0 and $i = N$, we have

$$[P_a]_j^{[n+1]} = \frac{[\rho_a]_j^{[n+1]}\tilde{R}T}{\tilde{M}_a};\ \text{for } j = 0, N. \tag{3.23}$$

Thus it is necessary to solve the nonlinear algebraic system of equations (3.12), (3.13), (3.14), (3.22) and (3.23). To obtain the numerical solution of the coupled system of partial differential equations, one can adopt two strategies, the coupled and uncoupled strategies. Let $\mathbf{S} = [S_0\ S_1\ ...\ S_N]$ and $\boldsymbol{\rho_a} = [\rho_{a0}\ \rho_{a1}\ ...\ \rho_{aN}]$ denote the unknown vectors. Furthermore, let $\mathbf{F}(\mathbf{S},\boldsymbol{\rho_a}) = [F_0(\mathbf{S},\boldsymbol{\rho_a})\ F_1(\mathbf{S},\boldsymbol{\rho_a})\ ...\ F_N(\mathbf{S},\boldsymbol{\rho_a})]$ represent the nonlinear system of algebraic equations obtained from the discretization of the water balance equation, i.e., $\mathbf{F}(\mathbf{S},\boldsymbol{\rho_a}) = \mathbf{0}$ represents the set of equations (3.12), (3.13) and (3.14). Let $\mathbf{G}(\mathbf{S},\boldsymbol{\rho_a}) = \mathbf{0}$ represent the nonlinear system of equations obtained from the discretization of the air balance equation i.e., (3.22) and (3.23). For instance, if we first solve the water balance equation independently to compute the saturation $\mathbf{S}$, that is first to solve $\mathbf{F}(\mathbf{S^{n+1}},\boldsymbol{\rho_a}^{\mathbf{n}}) = \mathbf{0}$ and then solve the air balance equation for calculating $\boldsymbol{\rho_a}$ using the already computed $\mathbf{S}$, i.e., next solving $\mathbf{G}(\mathbf{S^{n+1}},\boldsymbol{\rho_a}^{\mathbf{n+1}}) = \mathbf{0}$, then this strategy is called the uncoupled strategy. Though this strategy takes less memory and is also slightly easier to implement than the fully coupled strategy, here we have noticed that the solution does not converge with larger time steps. This shows the strongly coupled behavior of the drying problem. Hence we adopt the fully coupled strategy, where we solve the complete nonlinear system of algebraic equations (3.12), (3.13), (3.14), (3.22) and (3.23) together, i.e., we solve $\mathbf{F}(\mathbf{S^{n+1}},\boldsymbol{\rho_a}^{\mathbf{n+1}}) = \mathbf{0}$ and $\mathbf{G}(\mathbf{S^{n+1}},\boldsymbol{\rho_a}^{\mathbf{n+1}}) = \mathbf{0}$.

Algorithm

Here we give the algorithm which describes the implementation of the coupled problem with fully coupled strategy i.e., we solve the nonlinear system $\mathbf{F}(\mathbf{S^{n+1}}, \boldsymbol{\rho_a}^{\mathbf{n+1}}) = \mathbf{0}$ and $\mathbf{G}(\mathbf{S^{n+1}}, \boldsymbol{\rho_a}^{\mathbf{n+1}}) = \mathbf{0}$. The algorithm reads

1. Initialize the 21 variables.
2. Do the time loop.
3. Start the Newton iterations (outer iteration)
   - initial guess $S_j^{n+1} = S_j^n$ and $\rho_{a_j}^{n+1} = \rho_{a_j}^n$, $j = 0, 1, 2..., N$.
   - initialize the nonlinear function set $F_j(\mathbf{S}^{n+1}, \rho_{\mathbf{a}}{}^{n+1})$=0 and $G_j(\mathbf{S}^{n+1}, \rho_{\mathbf{a}}{}^{n+1})$=0, for $j = 0, 1, 2, ...N$.
   - Compute the increments $\zeta_1\mathbf{S}_i = [S_0, S_1, .., S_i + \epsilon, .., S_N]$, for a small $\epsilon > 0$ and compute all the remaining variables for this $\zeta_1\mathbf{S}_i$ using the equations of state. Similarly compute $\zeta_2\boldsymbol{\rho_a}_i^{n+1} = [\rho_{a0}, \rho_{a1}, ..., \rho_{ai} + \epsilon, ..., \rho_{aN}]$ and compute all the remaining variables for this $\zeta_2\boldsymbol{\rho_a}_i$ for all i=0,1,2,...,N,
   - compute $\mathrm{F}_j(\zeta_1\mathbf{S}_i, \boldsymbol{\rho_a}), \mathrm{F}_j(\mathbf{S}, \zeta_2\boldsymbol{\rho_a}), \mathrm{G}_j(\zeta_1\mathbf{S}_i, \boldsymbol{\rho_a}), \mathrm{G}_j(\mathbf{S}, \zeta_2\boldsymbol{\rho_a})$ for all $j = 0, 1, 2, ...N$ and for $i = j - 1, j, j + 1$,
   - approximate the entries in the Jacobian matrix

$$\frac{dF_j}{dS_i} = \frac{F_j(\zeta_1\mathbf{S}_i, \boldsymbol{\rho_a}) - F_j(\mathbf{S}, \boldsymbol{\rho_a})}{\epsilon} \text{ for } i = j - 1, j, j + 1$$

$$\frac{dF_j}{d\rho_{ai}} = \frac{F_j(\mathbf{S}, \zeta_2\boldsymbol{\rho_a}_i) - F_j(\mathbf{S}, \boldsymbol{\rho_a})}{\epsilon} \text{ for } i = j - 1, j, j + 1$$

$$\frac{dG_j}{dS_i} = \frac{G_j(\zeta_1\mathbf{S}_i, \boldsymbol{\rho_a}) - G_j(\mathbf{S}, \boldsymbol{\rho_a})}{\epsilon} \text{ for } i = j - 1, j, j + 1$$

$$\frac{dG_j}{d\rho_{ai}} = \frac{G_j(\mathbf{S}, \zeta_2\boldsymbol{\rho_a}_i) - G_j(\mathbf{S}, \boldsymbol{\rho_a})}{\epsilon} \text{ for } i = j - 1, j, j + 1$$

   - assemble the Jacobian matrix $[\mathbf{J_c}]_{(2N+2)\times(2N+2)}$ and compute the vector $(\mathbf{b})_{(2N+2)\times 1}$=$[\mathbf{F}(\mathbf{S}, \boldsymbol{\rho_a})\ \mathbf{G}(\mathbf{S}, \boldsymbol{\rho_a})]$,
   - compute $\mathbf{z}_{(2N+2)\times 1} = [\mathbf{J_c}]^{-1}(\mathbf{b})_{(2N+2)\times 1}$ by an iterative (inner iteration) or a direct method.
   - calculate $\mathbf{S}^{n+1} = \mathbf{S}^n - \mathbf{z}_{(N+1)\times 1}$, the first N+1 entries of $\mathbf{z}_{(2N+2)\times 1}$ and $\rho_{\mathbf{a}}{}^{n+1} = \rho_{\mathbf{a}}{}^n - \mathbf{z}_{(N+1)\times 1}$, the last N+1 entries of $\mathbf{z}_{(2N+2)\times 1}$
   - update the remaining 19 variables,
   - check if $(\mathbf{S}^{n+1} - \mathbf{S}^n) >$ (tolerance factor) and $(\rho_{\mathbf{a}}{}^{n+1} - \rho_{\mathbf{a}}{}^n) >$ (tolerance factor).

4. Stop the Newton iterations, if the desired tolerance is reached.

5. Compute the average moisture content.

6. End the time loop if the desired moisture content is reached.

The Jacobian matrix

For instance, if we consider 6 nodes, the vector $\mathbf{b}$ is given by
$\mathbf{b} = [F_0, F_1, F_2, F_3, F_4, F_5, G_0, G_1, G_2, G_3, G_4, G_5]^T$.

The Jacobian matrix $\mathbf{J_c}$ is a full matrix. It is assembled in the following way

$$\mathbf{J_c} = \begin{pmatrix}
\frac{\partial F_0}{\partial S_0} & \frac{\partial F_0}{\partial S_1} & 0 & 0 & 0 & 0 & \frac{\partial F_0}{\partial \rho_{a\,0}} & \frac{\partial F_0}{\partial \rho_{a\,1}} & 0 & 0 & 0 & 0 \\
\frac{\partial F_1}{\partial S_0} & \frac{\partial F_1}{\partial S_1} & \frac{\partial F_1}{\partial S_2} & 0 & 0 & 0 & \frac{\partial F_1}{\partial \rho_{a\,0}} & \frac{\partial F_1}{\partial \rho_{a\,1}} & \frac{\partial F_1}{\partial \rho_{a\,2}} & 0 & 0 & 0 \\
0 & \frac{\partial F_2}{\partial S_1} & \frac{\partial F_2}{\partial S_2} & \frac{\partial F_2}{\partial S_3} & 0 & 0 & 0 & \frac{\partial F_2}{\partial \rho_{a\,1}} & \frac{\partial F_2}{\partial \rho_{a\,2}} & \frac{\partial F_2}{\partial \rho_{a\,3}} & 0 & 0 \\
0 & 0 & \frac{\partial F_3}{\partial S_2} & \frac{\partial F_3}{\partial S_3} & \frac{\partial F_3}{\partial S_4} & 0 & 0 & 0 & \frac{\partial F_3}{\partial \rho_{a\,2}} & \frac{\partial F_3}{\partial \rho_{a\,3}} & \frac{\partial F_3}{\partial \rho_{a\,4}} & 0 \\
0 & 0 & 0 & \frac{\partial F_4}{\partial S_3} & \frac{\partial F_4}{\partial S_4} & \frac{\partial F_4}{\partial S_5} & 0 & 0 & 0 & \frac{\partial F_4}{\partial \rho_{a\,3}} & \frac{\partial F_4}{\partial \rho_{a\,4}} & \frac{\partial F_4}{\partial \rho_{a\,5}} \\
0 & 0 & 0 & 0 & \frac{\partial F_5}{\partial S_4} & \frac{\partial F_5}{\partial S_5} & 0 & 0 & 0 & 0 & \frac{\partial F_5}{\partial \rho_{a\,4}} & \frac{\partial F_5}{\partial \rho_{a\,5}} \\
\frac{\partial G_0}{\partial S_0} & \frac{\partial G_0}{\partial S_1} & 0 & 0 & 0 & 0 & \frac{\partial G_0}{\partial \rho_{a\,0}} & \frac{\partial G_0}{\partial \rho_{a\,1}} & 0 & 0 & 0 & 0 \\
\frac{\partial G_1}{\partial S_0} & \frac{\partial G_1}{\partial S_1} & \frac{\partial G_1}{\partial S_2} & 0 & 0 & 0 & \frac{\partial G_1}{\partial \rho_{a\,0}} & \frac{\partial G_1}{\partial \rho_{a\,1}} & \frac{\partial G_1}{\partial \rho_{a\,2}} & 0 & 0 & 0 \\
0 & \frac{\partial G_2}{\partial S_1} & \frac{\partial G_2}{\partial S_2} & \frac{\partial G_2}{\partial S_3} & 0 & 0 & 0 & \frac{\partial G_2}{\partial \rho_{a\,1}} & \frac{\partial G_2}{\partial \rho_{a\,2}} & \frac{\partial G_2}{\partial \rho_{a\,3}} & 0 & 0 \\
0 & 0 & \frac{\partial G_3}{\partial S_2} & \frac{\partial G_3}{\partial S_3} & \frac{\partial G_3}{\partial S_4} & 0 & 0 & 0 & \frac{\partial G_3}{\partial \rho_{a\,2}} & \frac{\partial G_3}{\partial \rho_{a\,3}} & \frac{\partial G_3}{\partial \rho_{a\,4}} & 0 \\
0 & 0 & 0 & \frac{\partial G_4}{\partial S_3} & \frac{\partial G_4}{\partial S_4} & \frac{\partial G_4}{\partial S_5} & 0 & 0 & 0 & \frac{\partial G_4}{\partial \rho_{a\,3}} & \frac{\partial G_4}{\partial \rho_{a\,4}} & \frac{\partial G_4}{\partial \rho_{a\,5}} \\
0 & 0 & 0 & 0 & \frac{\partial G_5}{\partial S_4} & \frac{\partial G_5}{\partial S_5} & 0 & 0 & 0 & 0 & \frac{\partial G_5}{\partial \rho_{a\,4}} & \frac{\partial G_5}{\partial \rho_{a\,5}}
\end{pmatrix}$$

The solution vector $\mathbf{z}$ is given by
$\mathbf{z} = [S_0, S_1, S_2, S_3, S_4, S_5, \rho_{a0}, \rho_{a1}, \rho_{a2}, \rho_{a3}, \rho_{a4}, \rho_{a5}]^T$.

3.4.3 Simulation results

In this section, we present the numerical results for the coupled drying problem at isothermal conditions. We take the length of the concrete material as $L = 0.04m$. The space step size h is $\frac{0.04}{N}$, N is the number of cells. We take the time step size $\tau = 500$ when the saturation at all the nodes is less than the critical saturation S_{crit}, which we call the first drying state. During the intermediate drying state and second drying state, we take the time step size $\tau = 0.5$. The rapid reduction in the time step from 500 to 0.5 is due to the fastly changing variables at some local regions. This is due to the change of definitions of variables locally because of the change in drying states causes high diffusion at a small region of the domain at which a change in drying state occur. Also for larger time steps, the Jacobian matrix becomes ill-conditioned. For instance, at 15.87 hours of drying time using 50 cells with a small time step size of $\tau = 0.5$ during the second drying state, the condition number of the Jacobian matrix is 3127.2. Here, the step size selection is chosen to be fixed. The rapid reduction in the time step size causes poor efficiency. An adaptive time step strategy is essential which will be presented in the next section. Figure 3.4 represents the local average moisture content profile. From the Figure 3.4, we can see that the average moisture content decreases from an initial average moisture of 1.28 to 0.000001 as time increases. The drying time to reduce the moisture from 1.28 to 0.000001 is 4.05 days. Figure 3.5 represents the local moisture content profile from the surface of the domain to the center. Figure 3.6 represents the local gas pressure profiles. From Figure 3.6, we notice that the gas pressure goes down at the beginning of the drying process due to cooling effects, after some time the gas pressure goes up due to excessive reduction in vapour pressure. This is due to the definition of sorption equilibrium. At the end of the drying process, the gas pressure at all nodes comes to the atmospheric gas pressure conditions. Figure 3.7 and Figure 3.8 present the local vapour and local air pressures respectively. Figure 3.9 and 3.10 give the moisture content and air density profiles at various time levels from initial to final drying time.

Average moisture content profile

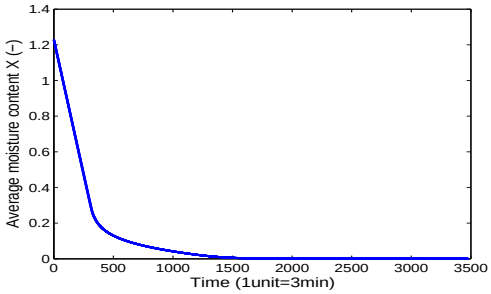


Figure 3.4: Average moisture content.

Local moisture content profile from surface to the center

The moisture content profiles are drawn for different cells from the surface to the center. The moisture content decreases in each cell as time increases.

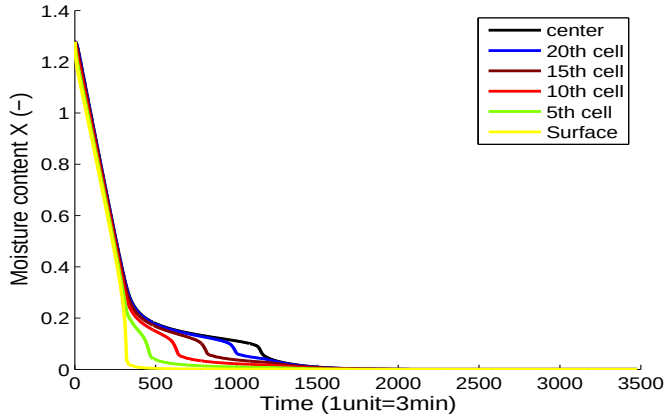


Figure 3.5: Local moisture content.

Local gas pressure profile

The gas pressure profiles drawn for different cells.

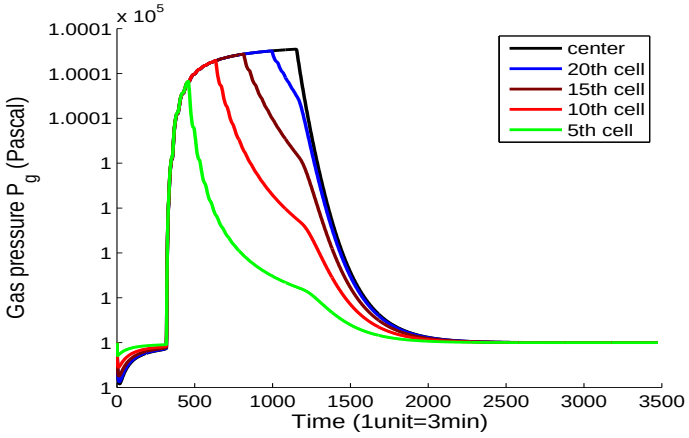


Figure 3.6: Local gas pressure.

Local vapour pressure profile

The vapour pressure profiles are drawn for different cells. The vapour pressure is constant which is equal to one at the initial stage of drying due to sorption equilibrium. When the moisture content is less than the critical moisture content level, then the vapour pressure starts decreasing with increase in time as shown in the following figure.

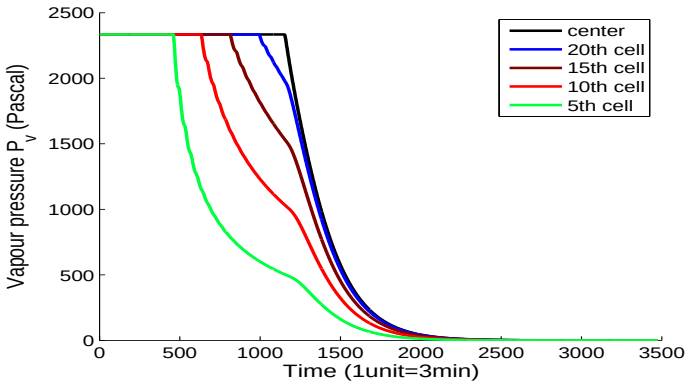


Figure 3.7: Local vapour pressure.

Local air pressure profile

The air pressure profiles are drawn for different cells. At the end of the drying when vapour is removed completely from the pores, all the cells are filled with air. This we can notice from the following figure where the air pressure increases as time increases.

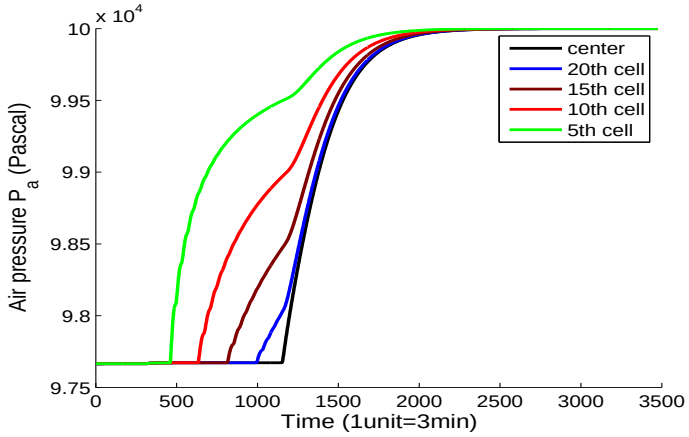


Figure 3.8: Local air pressure.

Moisture content at different time levels

Moisture content profiles at various time levels, drawn at an interval of 2 hours from initial time to the end of the drying.

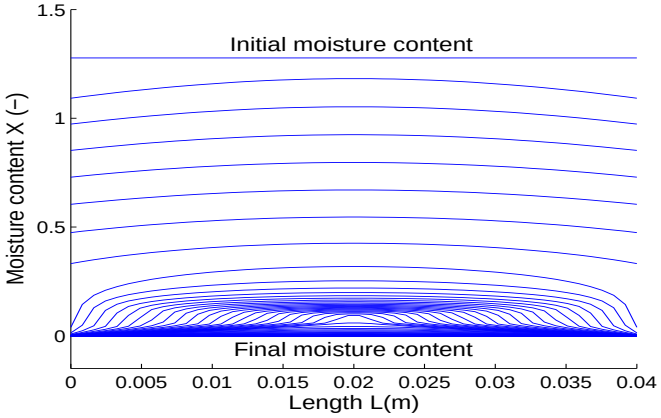


Figure 3.9: Moisture content profiles.

Air density at different time levels

Air density profiles at various time levels, drawn at an interval of 2 hours from initial time to the end of the drying.

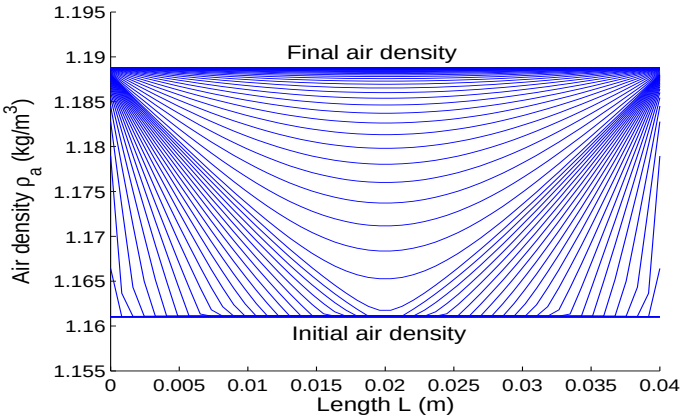
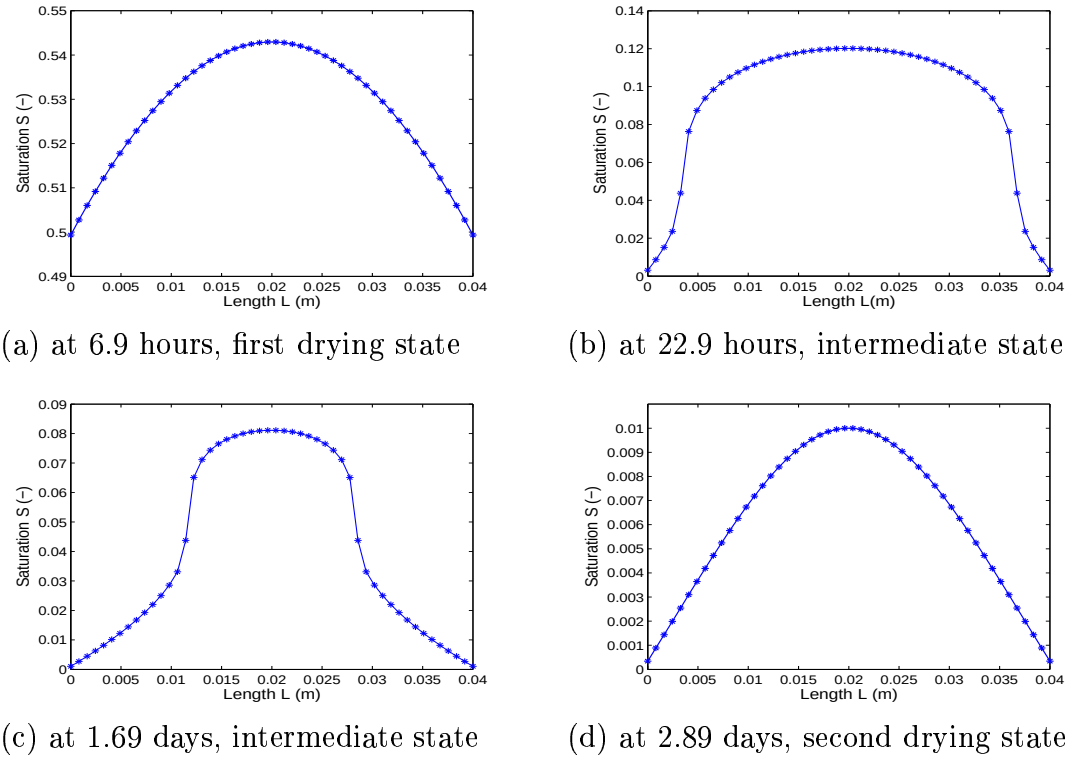


Figure 3.10: Air density profiles.

Saturation profiles at different drying times



(a) at 6.9 hours, first drying state

(b) at 22.9 hours, intermediate state

(c) at 1.69 days, intermediate state

(d) at 2.89 days, second drying state

Figure 3.11: Saturation profiles at different drying times.

From Figures 3.11(b) and 3.11(c), a rapid reduction can be seen in the saturation during the intermediate state of drying process.

3.5 Method of lines

The numerical solution of initial boundary value problems for time-dependent partial differential equations requires a discretization in space and time. Consider the PDE (3.1) and suppose that the PDE is discretized only in space subjected to initial and boundary conditions on a certain grid to yield a semi-discrete system of ODEs. This approach is called the method of lines MOL. The MOL approach, where space and time discretizations are considered independently, is conceptually simple and flexible. The popularity of this approach from a practical point of view is due to the fact that nowadays there exist many well developed ODE methods for instance see Shampine [40]. Let the semi-discrete system have the following form

$$\frac{\partial u}{\partial t} = f(u). \tag{3.24}$$

Now the semi-discrete system (3.24) is to be integrated with an appropriate ODE method. In this section we discuss an implicit Runge-Kutta time stepping method of order 5. For more details on time integration methods, we refer to Hundsdorfer and Verwer [20], Hairer and Wanner [18].

3.5.1 Semi-discrete system

Governing equations of isothermal drying

We write the governing equations of the coupled problem (3.19) and (3.20) in the following way

$$f(S)\frac{\partial S}{\partial t} = \left[\frac{\partial}{\partial x}\left(a_1(S,\rho_a)\frac{\partial}{\partial x}y_v + a_2(S)\frac{\partial}{\partial x}P_l + a_3(S)\frac{\partial}{\partial x}P_g\right)\right], \tag{3.25}$$

$$\psi(1-S)\frac{\partial \rho_a}{\partial t} - \psi\rho_a\frac{\partial S}{\partial t} = \frac{\partial}{\partial x}(b_1(S,\rho_a)\frac{\partial}{\partial x}y_a + b_2(S,\rho_a)\frac{\partial}{\partial x}P_g). \tag{3.26}$$

The function f(S) is given by

$$f(S) = \begin{cases} (A-B) & : \quad if\ S \geq S_{crit} \\ 3A_1S^2 + 2B_1S + C_1 & : \quad if\ S < S_{crit}. \end{cases}$$

Where A, B, A_1, B_1, C_1 are constants as given in 3.3.4. For computations we take the equation (3.26) in the following form

$$\psi(1-S)\frac{\partial \rho_a}{\partial t} = \frac{\partial}{\partial x}(b_1(S,\rho_a)\frac{\partial}{\partial x}y_a + b_2(S,\rho_a)\frac{\partial}{\partial x}P_g) + \psi\rho_a\frac{\partial S}{\partial t}. \tag{3.27}$$

The semi-discrete DAE system

We first discretize in space i.e., the right hand side of (3.25) and (3.27), using the numerical scheme given in Section 3.1. After discretization in space, we get a differential algebraic system. Let $h = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ be the space step size. Let N be the number of cells, i.e.

$h = \frac{L}{N}$, where L is the length of the domain. The length of the boundary cells is $\frac{h}{2}$. Then, for $1 \leq j \leq N-1$ we obtain the following ODE system of equations

$$\begin{aligned}\frac{\partial S_j}{\partial t} = &\frac{1}{hf(S_j)^{[n+1]}}[a_1(S,\rho_a)\frac{\partial}{\partial x}y_v + a_2(S)\frac{\partial}{\partial x}P_l + a_3(S)\frac{\partial}{\partial x}P_g]_{j+\frac{1}{2}}^{[n+1]} \\ &-\frac{1}{hf(S_j)^{n+1}}[a_1(S,\rho_a)\frac{\partial}{\partial x}y_v + a_2(S)\frac{\partial}{\partial x}P_l + a_3(S)\frac{\partial}{\partial x}P_g]_{j+\frac{1}{2}}^{[n+1]},\end{aligned} \tag{3.28}$$

at the left boundary $j = 0$, we obtain the following equation

$$\frac{\partial S_0}{\partial t} = \frac{2}{hf(S_0)^{n+1}}\left([a_1(S,\rho_a)\frac{\partial}{\partial x}y_v + a_2(S)\frac{\partial}{\partial x}P_l + a_3(S)\frac{\partial}{\partial x}P_g]_{\frac{1}{2}}^{[n+1]} - (J_w)_0\right), \tag{3.29}$$

and at the right boundary $j = N$, we get

$$\frac{\partial S_N}{\partial t} = \frac{2}{hf(S_N)^{n+1}}\left(-(J_w)_N - [a_1(S,\rho_a)\frac{\partial}{\partial x}y_v + a_2(S)\frac{\partial}{\partial x}P_l + a_3(S)\frac{\partial}{\partial x}P_g]_{N-\frac{1}{2}}^{n+1}\right). \tag{3.30}$$

Similarly we discretize the equation (3.27) in space, then for $1 \leq j \leq N-1$ we get the following system of ODEs

$$\begin{aligned}\frac{\partial \rho_{aj}}{\partial x} = &\frac{1}{h\psi(1-S_j)^{n+1}}([(b_1(S,\rho_a)\frac{\partial}{\partial x}y_a + b_2(S,\rho_a)\frac{\partial}{\partial x}P_g)]_{j+\frac{1}{2}}^{[n+1]}) - \\ &\frac{1}{h\psi(1-S_j)^{[n+1]}}([(b_1(S,\rho_a)\frac{\partial}{\partial x}y_a + b_2(S,\rho_a)\frac{\partial}{\partial x}P_g)]_{j+\frac{1}{2}}^{[n+1]}) + \psi\rho_a^{n+1}\frac{\partial S_j}{\partial t}.\end{aligned} \tag{3.31}$$

At the boundary nodes i=0 and $i = N$, we have the algebraic equations

$$[P_a]_j^{[n+1]} = \frac{[\rho_a]_j^{[n+1]}\tilde{R}T}{\tilde{M}_a}, \text{ where } j = 0, N. \tag{3.32}$$

3.5.2 Implicit Runge-Kutta method - Radau5 method

Not all type of Runge-Kutta methods are suitable for the solution of stiff ordinary differential equations. The system of ordinary differential equations obtained after the space discretization of the coupled drying equations (3.19) and (3.20) is a stiff DAE system. We have observed that the explicit ODE methods are not suitable for this problem, because these methods are not stable for stiff problems. Hence we use a 3-stage implicit Runge Kutta method of order 5 in time, which is called Radau5 method. We give very briefly the details of this method. For more details on solving stiff ODE and DAE systems, we refer to Hairer and Wanner [18]. Consider the ODE system given by (3.24). For this system, a q-stage implicit Runge Kutta method is given by

$$g_i = u_0 + \tau\sum_{j=1}^{q} a_{ij}f(t_0 + c_j\tau, g_j) \text{ for } i = 1, 2, ..., q, \tag{3.33a}$$

$$u_1 = u_0 + \tau \sum_{j=1}^{q} b_j f(t_0 + c_j\tau, g_j). \tag{3.33b}$$

Here q represents the number of stages of the method and we take q=3. We denote $\tau = t_{n+1} - t_n$ as the time step size, u_0=$u(t_0)$ and u_1=$u(t_1)$. The matrix $\mathbf{A} = (a_{ij})$ and the vectors b_j, c_j represent the Runge-Kutta coefficients. These coefficients can be found in Table 5.6 from the text book of Hairer and Wanner [18]. To reduce the round-off errors, the method works with smaller values

$$v_i = g_i - u_0. \tag{3.34}$$

Then the equation (3.33a) becomes

$$v_i = \tau \sum_{j=1}^{q} a_{ij} f(t_0 + c_j\tau, u_0 + v_j) \text{ for } i = 1, 2, ..., q. \tag{3.35}$$

Once we calculate the vector $\mathbf{v}$=$[v_1\ v_2\ ...\ v_q]$ by solving the system (3.35), then the solution $u = u_1$ at the new time step can be explicitly obtained from the equation (3.33b). This requires q additional function evaluations of the function f. This can be avoided if the Runge-Kutta (RK) matrix $\mathbf{A}$ is non singular. The equation (3.35) can be written as

$$\mathbf{v} = \mathbf{A}H(\mathbf{v}), \tag{3.36}$$

where $H(\mathbf{v})$=$[f(t_0 + c_1\tau, u_0 + v_1), ..., f(t_0 + c_1\tau, u_0 + v_1)]$ and $\mathbf{A}$ is the RK matrix. Finally, from (3.33b), we obtain

$$u_1 = u_0 + \sum_{i=1}^{q} d_i v_i. \tag{3.37}$$

Here $\mathbf{d} = \mathbf{b}\mathbf{A}^{-1}$, where $\mathbf{d} = [d_1\ d_2\ ...\ d_q]$ and $\mathbf{b} = [b_1\ b_2 ...\ b_q]$. Thus the solution to the ODE system (3.24) is obtained by solving the equation (3.37). To obtain the solution of (3.24), it is necessary to calculate $\mathbf{v}$ by solving the nonlinear system of equations (3.35). When we apply the Newton method to the system (3.35), it is required to evaluate the solution of a linear system at each iteration. The Jacobian matrix is given by

$$\begin{pmatrix} I - \tau a_{11}\frac{\partial f}{\partial u}(t_0 + c_1\tau, u_0 + v_1) & ... & -\tau a_{1q}\frac{\partial f}{\partial u}(t_0 + c_s\tau, u_0 + v_s) \\ \vdots & & \vdots \\ -\tau a_{q1}\frac{\partial f}{\partial u}(t_0 + c_1\tau T, u_0 + v_1) & ... & I - \tau a_{qq}\frac{\partial f}{\partial u}(t_0 + c_q\tau, u_0 + v_q) \end{pmatrix}. \tag{3.38}$$

This method uses a simplified Newton iterations by replacing all the Jacobians $\frac{\partial f}{\partial u}(t_0 + c_i\tau, u_0 + v_i)$ by an approximation $\mathbf{J_c} = \frac{\partial f}{\partial u}(t_0, u_0)$. Then the simplified Newton iterations applied to (3.35) yield

$$\begin{aligned} (\mathbf{I} - \tau\mathbf{A} \otimes \mathbf{J_c})\delta\mathbf{v}^k &= -\mathbf{v}^k + \tau(\mathbf{A} \otimes \mathbf{I})H(\mathbf{v}^k), \\ \mathbf{v}^{k+1} &= \mathbf{v}^k + \delta\mathbf{v}^k. \end{aligned} \tag{3.39}$$

Here $\mathbf{v^k}$ is the k^{th} approximation to the solution and $\delta\mathbf{v}^k$ are the increments. The stopping criterion of the Newton iterations is given by

$$\eta_k \|\delta\mathbf{v}^k\| \leq k(tol). \tag{3.40}$$

Here η_k is given by $\eta_k = \frac{\varphi_k}{1-\varphi_k}$, where φ_k is the convergence rate of the Newton iteration which is given by

$$\varphi_k = \frac{\|\delta\mathbf{v}^k\|}{\|\delta\mathbf{v}^{k-1}\|}, \ k \geq 1. \tag{3.41}$$

An essential gain of numerical work in solving the linear system (3.39) is obtained by the method introduced by Butcher [9] and Bickart [4]. The next important factor is the time step size selection. This method uses an embedded pair of methods to predict the time step size. Thus we need a lower order method which solves (3.24). Denote this solution by $\hat{u}_1$. Then the error is estimated by

$$err = (\mathbf{I} - \tau\gamma_0\mathbf{J_c})^{-1}(u_1 - \hat{u}_1). \tag{3.42}$$

For details of the lower order method $\hat{u}$, see the text book of Hairer and Wanner [18]. Here $\gamma_0 = \gamma^{-1}$, where γ is the real eigenvalue of the matrix $\mathbf{A}^{-1}$. The time step size can be now predicted using the formula

$$\tau_{new} = (f_c)\tau_{old}\left(\frac{tol}{\|err\|}\right)^{0.25}. \tag{3.43}$$

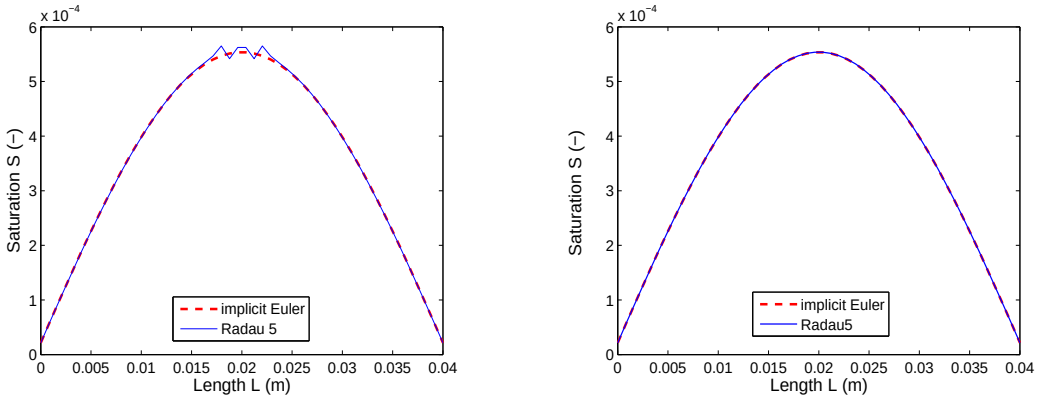
Here, tol is the prescribed tolerance and f_c is called the safety factor which depends on N_t, the total number of Newton iterations of the current step and the maximum number of Newton iterations $k_{\max}$, f_c is given by $\frac{0.9(2k_{\max}+1)}{2k_{\max}+N_t}$. For the rigourous details of this Radau5 method, we refer to Hairer and Wanner [18].

3.5.3 Numerical observations with the Radau5 method

In this section we present some numerical aspects of the coupled problem using the Radau5 method. All the material data are taken the same as in the case of Section 3.4. The simulations are done for a concrete material of length 0.04m. The simulations are carried out until a real drying time of 4.05 days. In 4.05 days the average saturation reduces from 1.28 to 0.000001.

Instability with large tolerances

The following figure shows the instability when using a large tolerance (tol). The Figure 3.12 represents the saturation profiles drawn at a real drying time of 4.05 days is reached. We can observe from the Figure 3.12(a), that the solution is non smooth at the peak of the curve with large tolerances. To avoid these bumps, the prescribed tolerance should be decreased. Through this the time step sizes are decreased. From the right hand figure



(a) tol=10^{-4} in Radau5 (b) tol=10^{-5} in Radau5

Figure 3.12: Saturation profiles.

3.12(b), we can observe that these bumps disappear with a tolerance of 10^{-5}. In these figures, the solution using cell-centered finite volume using the implicit Euler method is compared with that of the Radau5 method.

Time step sizes

The following figure represents the accepted time step sizes with Radau5 with a mesh of 100 cells.

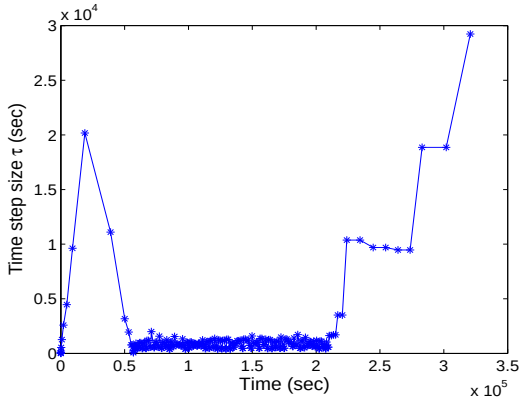


Figure 3.13: Accepted time step sizes.

It can be noticed from the these figures that there are time step reductions only at a certian period of drying process. These time step reductions are due to the change of drying states

which cause sudden increase or decrease in many variables locally. We observe these time step reductions during the intermediate state of drying, at this state the problem becomes highly stiff. An adaptive time step size selection using Radau5 is efficient compared to a fixed time step selection. Through our numerical computations in the one dimensional case, all the physical trends of the drying process are attained properly. A detailed study about these time step size reductions and suitable time stepping strategies has been studied in Chapter 6.

Chapter 4

Positivity of a parabolic problem

Positive solutions for ordinary as well as partial differential equations are important in a wide range of physical, chemical, biological and engineering problems. Many such model problems are quasilinear time dependent parabolic equations. Most of the physical quantities such as concentrations, densities, saturation etc., cannot become negative. For example, in the drying of porous medium, the problem which we study at present, most of the variables such as saturation, moisture content, densities of air, vapour, gas etc., always have positive values. This motivates us to investigate positivity preservation of the numerical solution for time-dependent quasilinear parabolic problems. We first start with a quasilinear parabolic initial value problem to prove the positivity of the numerical solution, then we extend the analysis to a boundary value problem. The boundary conditions are nonlinear mixed Neumann type. Finally, the analysis of positivity is extended to a particular case of a quasilinear scalar parabolic problem. We reduce the coupled drying problem of Perré [36] to a scalar quasilinear problem under some assumptions. We apply our analysis to the reduced problem as an application. Also we prove the stability of the above cases in the L_∞ norm.

4.1 Initial value problem

Let us consider the following continuous partial differential equation

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(a(u)\frac{\partial u}{\partial x}\right) \tag{4.1}$$

as a model problem. This is a simple quasilinear diffusion equation for which we will only consider non-negative solutions. To make the problem well-posed we assume $a(u)\geq 0$ for all admissible u. For this equation we pose the initial value problem at time t_0 with the initial data $u(x,t_0)=u_0(x)\geq 0$. Let us consider a finite volume or finite difference discretization for the above one dimensional parabolic problem (4.1). We consider a semi-implicit discretization by taking the values of the diffusion coefficients $a(u)$ explicitly from the previous time level and the derivative terms $\frac{\partial u}{\partial x}$ implicitly. Divide the space interval [0,L] into finite

subintervals $0 = x_0 < x_1 ..... < x_N = L$, of constant space step size $h = x_i - x_{i-1}$. We denote $\tau = t_{n+1} - t_n$ as the time step size. Then the semi-implicit finite difference discretized form of the equation (4.1) is given by

$$u_i^{n+1} - u_i^n = \frac{\tau}{h^2}\left[a(u_{i+1}^n)(u_{i+1}^{n+1} - u_i^{n+1}) - a(u_i^n)(u_i^{n+1} - u_{i-1}^{n+1})\right]. \tag{4.2}$$

We put $\lambda := \frac{\tau}{h^2}$ and obtain the following equation

$$\left[1 + \lambda(a(u_{i+1}^n) + a(u_i^n))\right] u_i^{n+1} - \lambda a(u_{i+1}^n) u_{i+1}^{n+1} - \lambda a(u_i^n) u_{i-1}^{n+1} = u_i^n. \tag{4.3}$$

The system of equations given by (4.3) can be represented as

$$\mathbf{A}\mathbf{u}^{n+1} = \mathbf{u}^n. \tag{4.4}$$

Where $\mathbf{u}^{n+1}$ is the unknown vector $[u_1^{n+1} u_2^{n+1} \ ... \ u_N^{n+1}]$ and the matrix $\mathbf{A}$ is given by $\mathbf{A} = (a_{ij})_{1 \le i,j \le N}$ with

$$a_{ij} = \begin{cases} 1 + \lambda\left(a(u_{i+1}^n) + a(u_i^n)\right) & : \quad i = j \\ -\lambda\left(a(u_{i+1}^n)\right) & : \quad i = j - 1 \\ -\lambda\left(a(u_i^n)\right) & : \quad i = j + 1 \\ 0 & : \quad \text{otherwise.} \end{cases} \tag{4.5}$$

Hence the matrix $\mathbf{A}$ is a tridiagonal matrix. The diagonal entries of the matrix $\mathbf{A}$ satisfy

$$a_{ii} = 1 + \lambda\left(a(u_{i+1}^n) + a(u_i^n)\right) > \ \lambda\left(a(u_i^n) + a(u_{i+1}^n)\right),$$

this implies

$$a_{ii} > \sum_{\substack{j=1 \\ j \neq i}}^{N} \mid a_{ij} \mid . \tag{4.6}$$

The treatment of the boundary nodes will be discussed in the next section.

Definition 4.1 *A matrix satisfying the above inequality 4.6 is called a strictly diagonally dominant matrix.*

Hence the matrix $\mathbf{A}$ given by (4.5) is a strictly diagonally dominant matrix.

Definition 4.2 *M-matrix*

A matrix $\mathbf{A}$ is said be an **M-matrix** if $\mathbf{A^{-1}} \geq 0$ and $a_{ij} \leq 0$ for i$\neq$j.

Note that the matrix $\mathbf{A}$ given by (4.5) satisfies the M-matrix property. A large number of properties of M-matrices are known by now, for interested reader on M-matrices, we refer to Ortega [29] or Windisch [52].

Definition 4.3 *Spectral Radius*

Let $\mathbf{A}$ be an $N \times N$ matrix with complex or real entries. Let $K_1, K_2, K_3, ...K_N$ be the eigenvalues of the matrix $\mathbf{A}$. Then the **spectral radius** $\rho(\mathbf{A})$ of the matrix $\mathbf{A}$ is given by $\rho(\mathbf{A}) = \max_{1 \le i \le n} |K_i|$.

Lemma 4.1 *Let* $\mathbf{A}$ *be a matrix such that* $a_{ij} \le 0$ *for* $i \ne j$. *Then* $\mathbf{A}$ *is an M-matrix if and only if* $a_{ii} > 0$, *for* $i=1,2....,N$ *and the matrix* $\mathbf{B} = \mathbf{I}\text{-}\mathbf{D^{-1}A}$, *where* $\mathbf{I} = diag(1, 1, ...1)$ *and* $\mathbf{D}=diag(a_{11}, a_{22}, ...a_{NN})$, *satisfies* $\rho(\mathbf{B}) < 1$.

Proof: See Ortega [29]. □

Lemma 4.2 *The solution of the initial value problem (4.1) preserves positivity with the discretization given by (4.2).*

Proof: We split the matrix $\mathbf{A}$ in the following way,

$$\mathbf{D^{-1}A = I - B}$$

which implies

$$\mathbf{A = DI - DB}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{D}$=diag$(a_{11}, a_{22}, ...a_{nn})$ is the diagonal matrix. It is given by

$$\mathbf{D} = d_{ij} = \begin{cases} 1 + \lambda \left(a(u_{i+1}^n) + a(u_i^n) \right) & : \quad i = j \\ 0 & : \quad \text{otherwise.} \end{cases}$$

The matrix $\mathbf{B}$ which consists of the off diagonal entries is given by

$$\mathbf{B} = b_{ij} = \begin{cases} \frac{-\lambda a(u_i^n)}{1+\lambda \left(a(u_{i+1}^n) + a(u_i^n) \right)} & : \quad i = j + 1 \\ \frac{-\lambda a(u_{i+1}^n)}{1+\lambda \left(a(u_{i+1}^n) + a(u_i^n) \right)} & : \quad i = j - 1 \\ 0 & : \quad \text{otherwise.} \end{cases}$$

From the definition of the matrix $\mathbf{B}$, we obtain $\|\mathbf{B}\|_\infty < 1$.
Therefore we get $\rho(\mathbf{B}) \le \|\mathbf{B}\|_\infty < 1$, which implies by Lemma 4.1 that the matrix $\mathbf{A}$ is an **M-matrix** and thus we obtain $\mathbf{A^{-1}} \ge 0$.
Consider the system of equations given by (4.4), then we have

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

since $\mathbf{A^{-1}} \ge 0$, and $u_i^n \ge 0$ for all i, it implies that

$$u_i^{n+1} \ge 0 \text{ for all } i.$$

Hence the solution $\mathbf{u}$ of the problem (4.1) obtained from the numerical scheme (4.2) is positivity preserving. □

The following is an exercise in Blum [6]. For the sake of completeness we give the proof.

Lemma 4.3 *Let* $\mathbf{A}=(a_{ij})_{1\le i,j\le n}$ *be a strictly diagonally dominant matrix, then the matrix* $\mathbf{A}$ *satisfies the following condition*

$$\|\mathbf{A^{-1}}\|_\infty \le \left[\min_{1\le i\le n}\left(|a_{ii}| - \sum_{\substack{j=1\\ j\ne i}}^{n}|a_{ij}|\right)\right]^{-1}.$$

Proof: Consider a system of equations given by $\mathbf{A}X = Y$ where $Y = (y_i)_{1\le i\le N}$ and $X = (x_i)_{1\le i\le N}$ are vectors of length N and $\mathbf{A}$ is an $N\times N$ matrix, then we can write

$$y_i = \sum_{j=1}^{n} a_{ij}x_j \text{ for } 1 \le i \le N.$$

Let

$$\|x\|_\infty = \max_{1\le i\le N}|x_i| = |x_k| \text{ for some } k. \tag{4.7}$$

Then we have,

$$\|y\|_\infty = \max_i |y_i| = \max_i |\sum_{j=1}^{n} a_{ij}x_j| \ge |\sum_{j=1}^{n} a_{kj}x_j|,$$

where k is defined as in (4.7). Now we obtain

$$\|y\|_\infty \ge |a_{kk}x_k| - \sum_{\substack{j=1\\ j\ne k}}^{n}|a_{kj}x_j| = |a_{kk}||x_k| - \sum_{\substack{j=1\\ j\ne k}}^{n}|a_{kj}||x_j|.$$

This implies that

$$\|y\|_\infty \ge |a_{kk}|\|x\|_\infty - \sum_{\substack{j=1\\ j\ne k}}^{n}|a_{kj}|\|x\|_\infty,$$

which follows from 4.7. It gives

$$\|y\|_\infty \ge \left(|a_{kk}| - \sum_{\substack{j=1\\ j\ne k}}^{n}|a_{kj}|\right)\|x\|_\infty.$$

Note that the term in the parenthesis is positive, since $\mathbf{A}$ is a strictly diagonally dominant matrix. From this, we obtain the following inequality

$$\frac{\|y\|_\infty}{\|x\|_\infty} \ge \left(|a_{kk}| - \sum_{\substack{j=1\\ j\ne k}}^{n}|a_{kj}|\right).$$

Consider $\frac{\|\mathbf{A^{-1}}y\|_\infty}{\|y\|_\infty}$, then we get

$$\frac{\|\mathbf{A^{-1}}y\|_\infty}{\|y\|_\infty} = \frac{\|x\|_\infty}{\|y\|_\infty} \leq \frac{1}{\left(|a_{kk}| - \sum_{\substack{j=1 \\ j\neq k}}^{n} |a_{kj}|\right)} \leq \frac{1}{\min_i \left(|a_{ii}| - \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|\right)}, \tag{4.8}$$

and therefore we obtain

$$\|\mathbf{A^{-1}}\|_\infty = \sup_{y\neq 0} \frac{\|\mathbf{A^{-1}}y\|_\infty}{\|y\|_\infty} \leq \frac{1}{\min_i \left(|a_{ii}| - \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|\right)}.$$

□

Lemma 4.4 *Consider the initial value problem (4.1). The numerical scheme given by (4.2) is unconditionally stable under the L_∞ norm i.e., $\|u^{n+1}\|_\infty \leq \|u^n\|_\infty$.*

Proof: Consider the algebraic system of equations given by (4.4), then we have

$$\|u^{n+1}\|_\infty \leq \|\mathbf{A^{-1}}\|_\infty \|u^n\|_\infty. \tag{4.9}$$

Since $\mathbf{A}$ is strictly diagonally dominant, it follows from Lemma 4.3 that

$$\|\mathbf{A^{-1}}\|_\infty \leq \left[\min_{1\leq i\leq n} \left(a_{ii} - \sum_{\substack{j=1 \\ i\neq j}}^{n} |a_{ij}|\right)\right]^{-1}. \tag{4.10}$$

From the definition 4.5 of the matrix $\mathbf{A}$, we have

$$\left[\min_{1\leq i\leq n} \left(a_{ii} - \sum_{\substack{j=1 \\ i\neq j}}^{n} |a_{ij}|\right)\right]^{-1} = 1. \tag{4.11}$$

From the equations (4.9), (4.10) and (4.11), we now get

$$\|u^{n+1}\|_\infty \leq \|u^n\|_\infty.$$

Hence the numerical scheme given by (4.2) is L_∞ stable. □

4.2 Extending the study of positivity to a boundary value problem

In this section we prove the positivity and stability for a quasilinear boundary value parabolic problem.

4.2.1 Model problem

We take the following partial differential equation as the model problem

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(a(u)\frac{\partial u}{\partial x}\right). \tag{4.12}$$

We consider the initial condition as $u(x,t_0) = u_0(x) \geq 0$. Further we assume that $a : \mathbb{R} \to \mathbb{R}_0^+$ i.e., we have $a(u) \geq 0$ for all u.
Let the length of the 1-d domain be L. The boundary conditions are the fluxes at the left and the right ends of the domain. These fluxes are given by

$$\left(a(u)\frac{\partial u}{\partial x}\right) \cdot \bar{n} = g_1(u) \quad \text{at} \quad x = 0$$

and

$$\left(a(u)\frac{\partial u}{\partial x}\right) \cdot \bar{n} = g_2(u) \quad \text{at} \quad x = L.$$

Where we assume that $g_1(u) \geq 0$ and $g_2(u) \geq 0$ for all $u \geq 0$. Here $\bar{n}$ denote the outer normal vector.

4.2.2 Discretization

Let h be the space step size, τ be the time step size and $N+1$ be the number of nodal points i.e., $h = \frac{L}{N}$. Let $x_0, x_1, ...x_N$ be the nodes in [0,L] and $x_{i+1} - x_i = h$. Let us discretize the continuous equation (4.12) with semi-implicit finite difference scheme. The discretized equation of (4.12) is given by

$$u_i^{n+1} = u_i^n + \lambda\left[a(u_{i+1}^n)(u_{i+1}^{n+1} - u_i^{n+1}) - a(u_i^n)(u_i^{n+1} - u_{i-1}^{n+1})\right], \quad \text{for} \quad 1 < i < N-1 \tag{4.13}$$

where we denote $\lambda = \frac{\tau}{h^2}$.
At the left and right boundaries, i.e., at $x = 0$ and $x = N$, we have

$$u_0^{n+1} = u_0^n + \lambda\left[a(u_1^n)(u_1^{n+1} - u_0^{n+1}) - hg_1(u_0^n)\right] \tag{4.14}$$

and

$$u_N^{n+1} = u_N^n + \lambda\left[-hg_2(u_N^n) - a(u_N^n)(u_N^{n+1} - u_{N-1}^{n+1})\right]. \tag{4.15}$$

Theorem 4.1 *The numerical solution of the problem (4.12) is positive with semi-implicit finite difference scheme if the solution u satisfies $h\lambda g_1(u_0^n) \leq u_0^n$ and $h\lambda g_2(u_N^n) \leq u_N^n$, where u_0 and u_N are the boundary values and $\lambda = \frac{\tau}{h^2}$. Also the scheme is unconditionally stable under L_∞ norm.*

Proof: Consider the problem (4.12). For this problem, we consider the semi-implicit discretization (4.13), (4.14) and (4.15) introduced above. Then the system of equations (4.13), (4.14) and (4.15) take the form

$$\mathbf{A}\mathbf{u}^{n+1} = \mathbf{u}^n + \mathbf{b}^n. \tag{4.16}$$

Where $\mathbf{A}$ is an $(N+1)\times(N+1)$ matrix and the corresponding matrix is given by

$$a_{ij} = \begin{cases} 1+\lambda\left(a(u^n_{i+1})+a(u^n_i)\right) & : \quad i=j (i\neq 0, i\neq N) \\ -\lambda\left(a(u^n_{i+1})\right) & : \quad i=j-1 \\ -\lambda\left(a(u^n_i)\right) & : \quad i=j+1 \\ 1+\lambda\left(a(u^n_1)\right) & : \quad i=j=0 \\ 1+\lambda\left(a(u^n_{N-1})\right) & : \quad i=j=N \\ 0 & : \quad \text{otherwise.} \end{cases}$$

The vector $\mathbf{b}$ is given by

$$\mathbf{b}^n = \begin{pmatrix} -h\lambda g_1(u^n_0) \\ 0 \\ .. \\ .. \\ 0 \\ -\lambda g_2(u^n_N) \end{pmatrix}$$

and the vector $\mathbf{u}^{n+1} = [u^{n+1}_0 u^{n+1}_1 .... u^{n+1}_N]$ is the unknown vector.

Since $\mathbf{A}$ is strictly diagonally dominant, it follows that $\mathbf{A^{-1}}$ exists. Hence from the Lemma 4.1, it follows that $\mathbf{A^{-1}} \geq 0$, since there exist a matrix $\mathbf{B}$ satisfying $\rho(\mathbf{B}) < 1$, it can be easily observed in the similar way as in the Lemma 4.2. The positivity of the solution $\mathbf{u}$ depends on the positivity of the right hand side of (4.16) i.e., the solution $\mathbf{u}^{n+1}$ is positive if $\mathbf{u}^n + \mathbf{b}^n \geq 0$.
Now we claim that $\mathbf{u}^n + \mathbf{b}^n \geq 0$, where $\mathbf{u}^n + \mathbf{b}^n$ is given by

$$\mathbf{u}^n + \mathbf{b}^n = \begin{pmatrix} u^n_0 - h\lambda g_1(u^n_0) \\ u^n_1 \\ .. \\ .. \\ u^n_{N-1} \\ u^n_N - h\lambda g_2(u^n_N) \end{pmatrix}. \tag{4.17}$$

Since $\mathbf{u}^n \geq 0$, we have $\mathbf{u}^n + \mathbf{b}^n$ is ≥ 0 if the following conditions are satisfied

$$u^n_0 - h\lambda g_1(u^n_0) \geq 0 \quad \text{and} \quad u^n_N - h\lambda g_2(u^n_N) \geq 0,$$

which implies

$$h\lambda g_1(u^n_0) \leq u^n_0 \quad \text{and} \quad h\lambda g_2(u^n_N) \leq u^n_N \ . \tag{4.18}$$

For a chosen λ and a suitable time step size τ , such that the above condition (4.18) holds, we obtain $\mathbf{u}^{n+1}\geq 0$. Hence $\mathbf{u}$ is positivity preserving.

Stability: Consider the system of equations given by (4.16), then we have

$$\|\mathbf{u}^{n+1}\|_\infty \leq \|\mathbf{A^{-1}}\|_\infty \|\mathbf{u}^n + \mathbf{b}^n\|_\infty \leq \|\mathbf{A^{-1}}\|_\infty \|\mathbf{u}^n\|_\infty,$$

where the last inequality follows from the structure of $\mathbf{u}^n + \mathbf{b}^n$ in (4.17). From the Lemmas 4.3 and 4.4, we have $\|\mathbf{A^{-1}}\|_\infty \leq 1$. Hence we obtain

$$\|\bar{\mathbf{u}}^{n+1}\|_\infty \leq \|\bar{\mathbf{u}}^n\|_\infty.$$

Therefore the numerical scheme (4.13) is stable under the L_∞ norm. □

4.3 A quasilinear parabolic problem

In this section we would like to study the positivity and L_∞ stability of a quasilinear parabolic problem with nonlinear boundary conditions.

4.3.1 Model problem

Consider the following quasilinear partial differential equation as the model problem

$$\frac{\partial \theta(u)}{\partial t} = \frac{\partial}{\partial x}\left[\sum_{j=1}^{k}\left(a_j(u)\frac{\partial f_j(u)}{\partial x}\right)\right]. \tag{4.19}$$

We consider the initial condition: $u(x,0)=u_0(x) \geq 0$. We assume that $a_j : \mathbb{R} \to \mathbb{R}_0^+$, i.e., $a_j(u) \geq 0$ for all u. The boundary conditions are given by $\sum_{j=1}^{k}\left(a_j(u)\frac{\partial f_j(u)}{\partial x}\right)\cdot \bar{n} = r_1(u)$ at $x = 0$ and $\sum_{j=1}^{k}\left(a_j(u)\frac{\partial f_j(u)}{\partial x}\right)\cdot \bar{n} = r_2(u)$ at $x = L$, where we assume that the functions $r_1(u)$ and $r_2(u)$ are nonnegative for any $u \geq 0$.
Note that by doing the substitutions $v := \theta(u), \tilde{a}_j(v) := a_j(\theta^{-1}(v)), \tilde{f}_j(v) := f_j(\theta^{-1}(v))$, the equation (4.19) reduces to the form

$$\frac{\partial v}{\partial t} = \frac{\partial}{\partial x}\left[\sum_{j=1}^{k}\left(\tilde{a}_j(v)\frac{\partial \tilde{f}_j(v)}{\partial x}\right)\right], \tag{4.20}$$

in the case that θ is invertible.

4.3.2 Discretization

Let h be the space step size, τ be the time step size and $N+1$ be the number of nodal points i.e., $h = \frac{L}{N}$. Let $x_0, x_1, ...x_N$ be the nodes in [0,L] and $x_{i+1} - x_i = h$. Then, the corresponding discretized equation of the partial differential equation (4.19) is given by

$$\begin{aligned}\frac{\theta(u_i^{n+1}) - \theta(u_i^n)}{\tau} &= \sum_{j=1}^{k} \left[a_j(u_{i+1}^n) \left(\frac{f_j(u_{i+1}^{n+1}) - f_j(u_i^{n+1})}{h^2} \right) \right] \\ &\quad - \sum_{j=1}^{k} \left[a_j(u_i^n) \left(\frac{f_j(u_i^{n+1}) - f_j(u_{i-1}^{n+1})}{h^2} \right) \right],\end{aligned} \tag{4.21}$$

for $1 < i < N-1$.
At the left and right boundaries i.e., at i=0 and i=N, we have

$$\theta(u_0^{n+1}) = \theta(u_0^n) + \frac{\tau}{h^2} \left(\sum_{j=1}^{k} \left[a_j(u_1^n) \left(f_j(u_1^{n+1}) - f_j(u_0^{n+1}) \right) \right] - h r_1(u_0) \right), \tag{4.22}$$

and

$$\theta(u_N^{n+1}) = \theta(u_N^n) + \frac{\tau}{h^2} \left(-h r_2(u_N) - \sum_{j=1}^{k} \left[a_j(u_N^n) \left(f_j(u_N^{n+1}) - f_j(u_{N-1}^{n+1}) \right) \right] \right). \tag{4.23}$$

Theorem 4.2 *Consider the quasilinear parabolic problem (4.19) which satisfies the following conditions*
(i) The function $\theta : [-b, b] \to \mathbb{R}$, $b > 0$ is strictly increasing, differentiable with $\theta(0) = 0$
(ii) The continuous functions $f_j : [a, b] \to \mathbb{R}$, $a \geq 0, b > 0$ for all j, are differentiable,
(iii) $a_j(u) \geq 0$ for all j and for all u.
Then the solution u is positivity preserving, with semi-implicit finite difference scheme if u satisfies $h\lambda r_1(u_0^n) \leq \theta(u_0^n)$ and $h\lambda r_2(u_N^n) \leq \theta(u_N^n)$, where u_0 and u_N are the boundary values and $\lambda = \frac{\tau}{h^2}$. Also the scheme is stable in L_∞ norm.

Proof: Consider the problem (4.19). For this problem, consider the semi-implicit discretization (4.21), (4.22) and (4.23) introduced above.

We put $v_i := \theta(u_i)$. Since θ is strictly increasing function, we know that θ^{-1} exists, so we get $u_i = \theta^{-1}(v_i)$. Substituting $\theta(u_i) = v_i$ and $u_i = \theta^{-1}(v_i)$ in the discretized equation (4.21), we obtain

$$\begin{aligned}\frac{(v_i^{n+1} - v_i^n)}{\triangle t} &= \sum_{j=1}^{k} \left[a_j(\theta^{-1}(v_{i+1}^n)) \left(\frac{f_j(\theta^{-1}(v_{i+1}^{n+1})) - f_j(\theta^{-1}(v_i^{n+1}))}{h^2} \right) \right] \\ &\quad - \sum_{j=1}^{k} \left[a_j(\theta^{-1}(v_i^n)) \left(\frac{f_j(\theta^{-1}(v_i^{n+1})) - f_j(\theta^{-1}(v_i^{n-1}))}{h^2} \right) \right].\end{aligned}$$

We define the two compositions $f_i \circ \theta^{-1} := \tilde{f}_i$ and $a_i \circ \theta^{-1} := \tilde{a}_i$, and get

$$\frac{(v_i^{n+1} - v_i^n)}{\triangle t} = \sum_{j=1}^{k} \left[\tilde{a}_j(v_{i+1}^n) \left(\frac{\tilde{f}_j(v_{i+1}^{n+1}) - \tilde{f}_j(v_i^{n+1})}{h^2} \right) \right] - \sum_{j=1}^{k} \left[\tilde{a}_j(v_i^n) \left(\frac{\tilde{f}_j(v_i^{n+1}) - \tilde{f}_j(v_{i-1}^{n+1})}{h^2} \right) \right].$$

By the mean value theorem, we can write $\tilde{f}_j(v_{i+1}) - \tilde{f}_j(v_i) = \tilde{f}_j'(\eta_{i+1})(v_{i+1} - v_i)$, $\eta_{i+1} \in]v_i, v_{i+1}[$. Let $\lambda := \frac{\tau}{h^2}$, then for $1 < i < N-1$, we obtain

$$v_i^{n+1} = v_i^n + \lambda \left(\sum_{j=1}^{k} \left[\tilde{a}_j(v_{i+1}^n) \tilde{f}_j'(\eta_{i+1}^{n+1})(v_{i+1}^{n+1} - v_i^{n+1}) \right] \right) - \lambda \left(\sum_{j=1}^{k} \left[\tilde{a}_j(v_i^n) \tilde{f}_j'(\eta_i^{n+1})(v_i^{n+1} - v_{i-1}^{n+1}) \right] \right),$$

at the boundaries $i = 0$ and $i = N$, the equations (4.22) and (4.23) reduce to the following form

$$v_0^{n+1} = v_0^n + \lambda \left(\sum_{j=1}^{k} \left[\tilde{a}_j(v_1^n) \tilde{f}_j'(\eta_1^{n+1})(v_1^{n+1} - v_0^{n+1}) \right] - hr(u_0) \right) \tag{4.24}$$

and

$$v_N^{n+1} = v_N^n + \lambda \left(-hr(u_N) - \sum_{j=1}^{k} \left[\tilde{a}_j(v_N^n) \tilde{f}_j'(\eta_N^{n+1})(v_N^{n+1} - v_{N-1}^{n+1}) \right] \right). \tag{4.25}$$

For $1 < i < N-1$, the system of algebraic equations are given by,

$$\begin{aligned} &- \left[\lambda \left(\sum_{j=1}^{k} \tilde{a}_j(v_i^n) \tilde{f}_j'(\eta_i^{n+1}) \right) \right] v_{i-1}^{n+1} \\ &+ \left[1 + \lambda \left(\sum_{j=1}^{k} \tilde{a}_j(v_{i+1}^n) \tilde{f}_j'(\eta_{i+1}^{n+1}) + \sum_{j=1}^{k} \tilde{a}_j(v_i^n) \tilde{f}_j'(\eta_i^{n+1}) \right) \right] v_i^{n+1} \\ &- \left[\lambda \left(\sum_{j=1}^{k} \tilde{a}_j(v_{i+1}^n) \tilde{f}_j'(\eta_{i+1}^{n+1}) \right) \right] v_{i+1}^{n+1} = v_i^n, \end{aligned} \tag{4.26}$$

at the boundaries $i = 0$ and $i = N$, we get the following equations

$$\begin{aligned} &\left[1 + \lambda \left(\sum_{j=1}^{k} \tilde{a}_j(v_1^n) \tilde{f}_j'(\eta_1^{n+1}) \right) \right] v_0^{n+1} - \left[\lambda \left(\sum_{j=1}^{k} \tilde{a}_j(v_1^n) \tilde{f}_j'(\eta_1^{n+1}) \right) \right] v_1^{n+1} = v_0^n - h\lambda r_1(u_0), \\ &- \left[\lambda \left(\sum_{j=1}^{k} \tilde{a}_j(v_N^n) \tilde{f}_j'(\eta_N^{n+1}) \right) \right] v_{N-1}^{n+1} + \left[1 + \lambda \left(\sum_{j=1}^{k} \tilde{a}_j(v_N^n) \tilde{f}_j'(\eta_N^{n+1}) \right) \right] v_N^{n+1} = v_N^n - h\lambda r_2(u_N). \end{aligned} \tag{4.27}$$

The algebraic system given by (4.26) and (4.27) can be represented as

$$\mathbf{A}\mathbf{v}^{n+1} = \mathbf{b}^n. \tag{4.28}$$

Where the matrix $\mathbf{A}$ is defined by $\mathbf{A} = (a_{ij})$

$$a_{ij} = \begin{cases} 1 + \lambda \left(\sum_{j=1}^{n} \tilde{a}_j(v_{i+1}^n) \tilde{f}_j'(\eta_{i+1}^{n+1}) + \sum_{j=1}^{n} \tilde{a}_j(v_i^n) \tilde{f}_j'(\eta_i^{n+1}) \right) & : \quad i = j \\ -\lambda \left(\sum_{j=1}^{n} \tilde{a}_j(v_{i+1}^n) \tilde{f}_j'(\eta_{i+1}^{n+1}) \right) & : \quad i = j-1 \\ -\lambda \left(\sum_{j=1}^{n} \tilde{a}_j(v_i^n) \tilde{f}_j'(\eta_i^{n+1}) \right) & : \quad i = j+1 \\ 1 + \lambda \left(\sum_{j=1}^{n} \tilde{a}_j(v_1^n) \tilde{f}_j'(\eta_1^{n+1}) \right) & : \quad i = j = 0 \\ 1 + \lambda \left(\sum_{j=1}^{n} \tilde{a}_j(v_N^n) \tilde{f}_j'(\eta_N^{n+1}) \right) & : \quad i = j = N \\ 0 & : \quad \text{otherwise.} \end{cases}$$

The vector $\mathbf{b}$ is given by

$$\mathbf{b}^n = \begin{pmatrix} v_0^n - h\lambda r_1(u_0^n) \\ v_1^n \\ .. \\ .. \\ v_{N-1}^n \\ v_N^n - h\lambda r_2(u_N^n) \end{pmatrix},$$

and the vector $\mathbf{v}$ is given by $[v_0\ v_1 .... v_N]$. The matrix $\mathbf{A}$ is a strictly dominant tridiagonal matrix, and therefore from Lemma 4.1, it follows that $\mathbf{A^{-1}} \geq 0$.
We get the positivity of $\mathbf{v}$, if $\mathbf{b}$ is positive. The vector $\mathbf{b}$ is positive when the following conditions are satisfied

$$v_0^n - h\lambda r_1(u_0^n) \geq 0 \quad \text{and} \quad v_N^n - h\lambda r_2(u_N^n) \geq 0, \tag{4.29}$$

which implies

$$h\lambda r_1(u_0^n) \leq v_0^n \quad \text{and} \quad h\lambda r_2(u_N^n) \leq v_N^n. \tag{4.30}$$

Hence $\mathbf{v}^{n+1} = [v_0^{n+1} v_1^{n+1} ..... v_N^{n+1}]$ is positive by choosing a suitable h and λ so as to satisfy the condition (4.30), which implies that $v_i^{n+1} \geq 0$ for all i.

Stability of the vector v: Consider the system of equations given by (4.28), then we get

$$\|\mathbf{v}^{n+1}\|_\infty \leq \|\mathbf{A^{-1}}\|_\infty \|\mathbf{b}^n\|_\infty \leq \|\mathbf{A^{-1}}\|_\infty \|\mathbf{v}^n\|_\infty.$$

The matrix $\mathbf{A}$ satisfies the condition $\|\mathbf{A^{-1}}\|_\infty \leq 1$, which follows from the Lemma 4.3. Thus we obtain

$$\|\mathbf{v}^{n+1}\|_\infty \leq \|\mathbf{v}^n\|_\infty. \tag{4.31}$$

Hence we have proved that $\mathbf{v}$ is stable in L_∞ norm.

Positivity of the vector u:
We now prove that the solution $\mathbf{u}$ of the problem (4.19) preserves positivity.
From the initial conditions of the problem (4.19), we have $u_i^n \geq 0$ for all i. It implies that $\theta(u_i^n) \geq 0$ for all i, which follows from the condition (i) of the Theorem 4.2. So we have

$$\theta(u_i^n) = v_i^n \geq 0 \text{ for all } i.$$

Since we have the positivity $\mathbf{v}$, it implies that

$$v_i^{n+1} = \theta(u_i^{n+1}) \geq 0 \text{ for all } i\text{=0,1,...}N.$$

From the condition (i) for $\theta(u)$ in the Theorem 4.2, we obtain

$$u_i^{n+1} \geq 0 \text{ for all } i.$$

Hence it proves that the solution $\mathbf{u}$ of the problem (4.19) preserves positivity.

L_∞ Stability of the vector u:
We now prove that the solution $\mathbf{u}$ of the problem (4.19) is stable in the L_∞ norm.
Since $\mathbf{v}$ is L_∞ stable, from (4.31), we have $\|\mathbf{v}^{n+1}\|_\infty \leq \|\mathbf{v}^n\|_\infty$. It implies

$$v_i^{n+1} \leq \max\left(v_0^n, ..., v_{i-1}^n, v_i^n, v_{i+1}^n, ..., v_N^n\right).$$

Since θ is an increasing function, it implies that θ^{-1} is also an increasing function, and thus we have $\theta^{-1}(v_i^{n+1}) \leq \theta^{-1}\left(\max(v_0^n, ..., v_{i-1}^n, v_i^n, v_{i+1}^n, ..., v_N^n)\right)$ which implies

$$\theta^{-1}(v_i^{n+1}) \leq \max(\theta^{-1}(v_0^n)....., \theta^{-1}(v_{i-1}^n), \theta^{-1}(v_i^n), \theta^{-1}(v_{i+1}^n), ..., \theta^{-1}(v_N^n)).$$

Hence we obtain

$$u_i^{n+1} \leq \max_{0\leq j\leq N}\left(\theta^{-1}(v_j^n)\right) \text{ for all } i\text{=0,1,...}N.$$

It implies that $u_i^{n+1} \leq \max_{0\leq j\leq N}(u_j^n)$ for all i, and thus we obtain

$$\|\mathbf{u}^{n+1}\|_\infty \leq \|\mathbf{u}^n\|_\infty.$$

Hence it follows that the solution $\mathbf{u}$ is L_∞ stable.

Hence we obtain that the solution $\mathbf{u}$ of the problem (4.19) is positive if $\mathbf{u}$ satisfies $h\lambda r_1(u_0^n) \leq \theta(u_0^n)$ and $h\lambda r_2(u_N^n) \leq \theta(u_N^n)$ and also we have proved that $\mathbf{u}$ is stable in the L_∞ the norm. □

4.4 Applications

4.4.1 Diffusion of vapour

Consider the diffusion of vapour in the porous medium. It has been assumed that there is no liquid in the porous medium, so the saturation remains constant. Also it is assumed that air flows slowly and there is no external heat supply. Hence there is no temperature differences and pressure differences in the gas. Gas in the porous medium is a mixture of vapour and air. The vapour should be diffused out of the porous medium.
Governing equation:

$$\frac{\partial \tilde{y}_v}{\partial t} = D_{eff} \triangle \tilde{y}_v,$$

with initial condition $\tilde{y}_v = 0.8$ and boundary condition $\frac{\partial \tilde{y}_v}{\partial n} = -\frac{\beta \tilde{y}_v}{D_{eff}}$.
Non-dimentionalising the above equation by introducing a reference time $T = \frac{L^2}{D_{eff}}$, characteristic length $L = 0.04m$, $D_{eff} = 25.10^{-6} \frac{m^2}{sec}$, the following equation has been derived

$$\frac{\partial \tilde{y}_v}{\partial t^*} = \triangle^* \tilde{y}_v, \tag{4.32}$$

with initial condition $\tilde{y}_v = 0.8$ and boundary condition $\frac{\partial \tilde{y}_v}{\partial n} = -16\tilde{y}_v$.
Here $\tilde{y}_v$, t are nondimentionalised quantities.

Numerical observations:

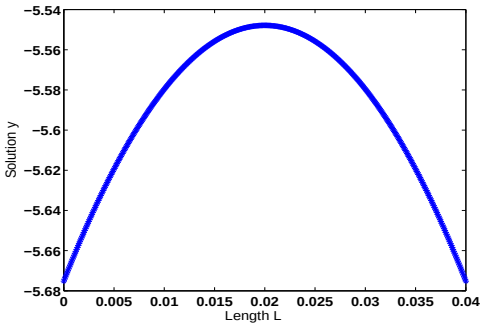
The numerical observations for the problem (4.32) are plotted in the Figure 4.1.

4.4.2 Isothermal drying at constant air flow

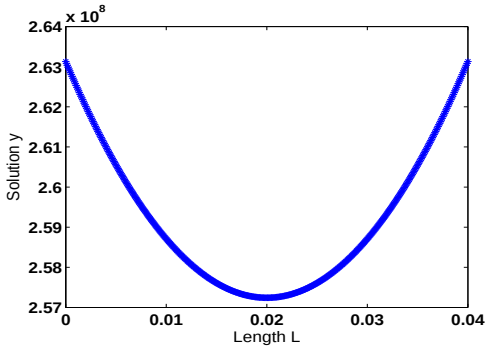
Drying is a heat and mass transfer process involved in the porous media. We reduce the problem by Perré [36] or see Chapter 2, to a scalar equation with some assumptions by considering the drying under constant temperature and constant air flow. The scalar problem consists of 19 variables. The main variable is S and the other 18 variables are given by the equations of state. For the complete set of equations, see Chapter 2. By substituting the equations of state into the partial differential equation, we write the PDE in terms one variable S. Then the partial differential equation take the following form

$$\frac{\partial \theta(S)}{\partial t} = \begin{cases} \frac{\partial}{\partial x}\left[b(S)\frac{\partial g(S)}{\partial x}\right] & : \text{ if } S \geq S_{crit} \\ \frac{\partial}{\partial x}\left[a(S)\frac{\partial f(S)}{\partial x} + c(S)\frac{\partial h(S)}{\partial x}\right] & : \text{ if } S < S_{crit} \end{cases} \tag{4.33}$$

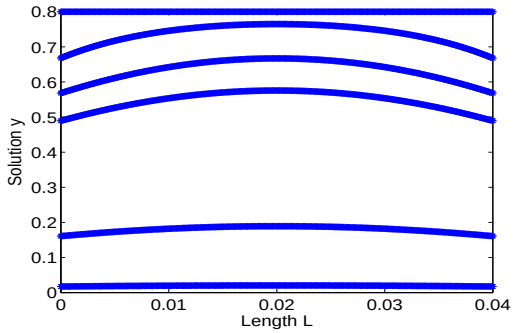
with initial condition $S(x,0)=S_0 \geq 0$. The flux at the boundary is a nonlinear function of S which is denoted by $r(S)$. Here we take the fluxes as symmetric. In general, they may not be symmetric in many other problems. But the proof of the Theorem 4.2 holds also if



(a) loss of positivity, 500 nodes



(b) loss of stability, 500 nodes



(c) saturation at different times satisfying positivity condition, 500 nodes

Figure 4.1: Numerical observations for the problem (4.32).

the fluxes are not symmetric. The boundary condition is given by $J \cdot \bar{n} = r(S)$ where J is the boundary flux. The problem under consideration satisfies the following conditions
(i) $\theta(S)$ is a strictly increasing function with $\theta(0) = 0$,
(ii) $f, g, h,$ are differentiable functions,
(iii) $a(S), b(S), c(S) \geq 0$ for all S.
Then the solution S is positivity preserving and L_∞ stable which follows from the Theorem 4.2.

Physically $f(S)$ denotes the mass fraction of vapour, $g(S)$ denotes the liquid pressure which is the difference of total pressure and capillary pressure and $h(S)$ denotes the total pressure. The functions $\theta(S)$, $f(S)$, $g(S)$, $h(S)$ are strictly increasing and continuous which are plotted in Figures 4.2, 4.3, 4.4 and 4.5 respectively in the next pages. The range of the solution is [0,1], since the saturation S always lies in between 0 and 1. If S is 1, then the porous medium is completely filled with water. When S reduces to zero, the drying is complete.

The function $\theta(S)$ is given by

$$\theta(S) = \begin{cases} (A-B)S + B & : \quad if \quad S \geq S_{crit} \\ A_1 S^3 + B_1 S^2 + C_1 S & : \quad if \quad S < S_{crit} \end{cases}$$

A=798.4, B= 0.013811, A_1=7.1867, B_1=-7.8168, C_1=799.0301.

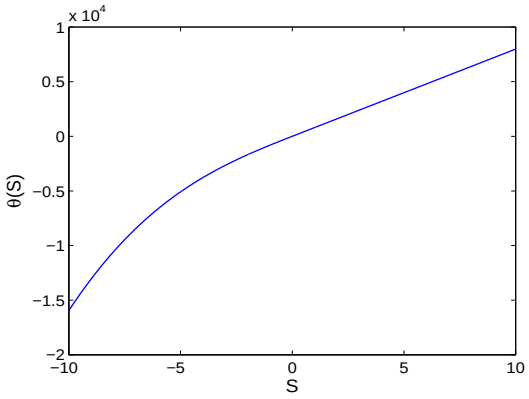


Figure 4.2: $\theta(S)$.

The function $f(S)$ is given by

$$f(S) = \frac{A_2 S^2 + B_2 S}{A_2 S^2 + B_2 S + C_2}, \text{ for } 0 \leq S \leq S_{crit}. \tag{4.34}$$

A_2=-8.98345, B_2=0.78762, C_2=1.161031.

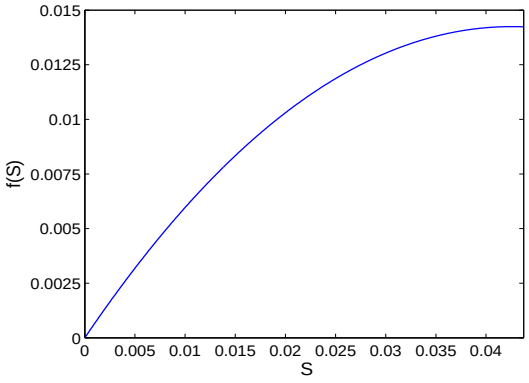


Figure 4.3: $f(S)$.

$$g(S) = 100000 - k_1 e^{(8.4057)10^{-0.3476(1.5968S-0.07)}}, \text{ for } S_{crit} \leq S \leq 0.8. \tag{4.35}$$

k_1=2.9040.

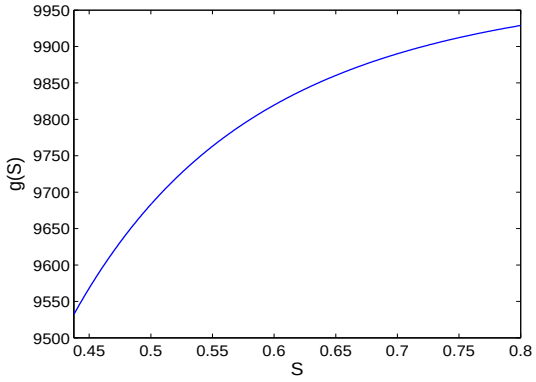


Figure 4.4: $g(S)$.

The function $h(S)$ is given by

$$h(S) = \lambda_4 + \lambda_5 S + \lambda_6 S^2, \text{ for } 0 \leq S \leq S_{crit}. \tag{4.36}$$

λ_4=97666.1, λ_5=106479.18628, $\lambda_6 = -1214471.176150$,

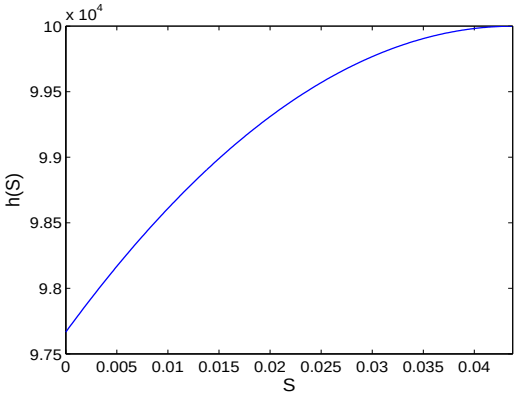


Figure 4.5: $h(S)$.

Thus the above problem (4.33) satisfies the conditions of the Theorem (4.2), hence according to Theorem (4.2) it follows that the solution of the problem (4.33) preserves positivity and L_∞ stable with a semi-implicit finite difference discretization.

4.5 Extending to fully implicit scheme

The above analysis of positivity preservation for the quasilinear parabolic problem (4.19) with a semi-implicit discretization can be easily extended to a fully implicit discretization.

4.5.1 Fully implicit discretization

Let h be the space step size, $\triangle t$ be the time step size and N+1 be the number of nodal points i.e., $h = \frac{L}{N}$. Let $x_0, x_1, ...x_N$ be the nodes in [0,L] and $x_{i+1} - x_i$=h. Then, the corresponding fully implicit discretized form of the equation (4.19) is given by

$$\begin{aligned}\frac{\theta(u_i^{n+1}) - \theta(u_i^n)}{\tau} = &\sum_{j=1}^{k}\left[a_j(u_{i+1}^{n+1})\left(\frac{f_j(u_{i+1}^{n+1}) - f_j(u_i^{n+1})}{h^2}\right)\right] \\ &-\sum_{j=1}^{k}\left[a_j(u_i^{n+1})\left(\frac{f_j(u_i^{n+1}) - f_j(u_{i-1}^{n+1})}{h^2}\right)\right],\end{aligned} \tag{4.37}$$

for $1 < i < N-1$. Near the boundaries at left and right ends i.e., at i=0 and i=N, we have

$$\begin{aligned}\theta(u_0^{n+1}) &= \theta(u_0^n) + \frac{\tau}{h^2}\left(\sum_{j=1}^{k}\left[a_j(u_1^{n+1})\left(f_j(u_1^{n+1}) - f_j(u_0^{n+1})\right)\right] - hr(u_0)\right), \\ \theta(u_N^{n+1}) &= \theta(u_N^n) + \frac{\tau t}{h^2}\left(-hr(u_N) - \sum_{j=1}^{k}\left[a_j(u_N^{n+1})\left(f_j(u_N^{n+1}) - f_j(u_{N-1}^{n+1})\right)\right]\right).\end{aligned} \tag{4.38}$$

Theorem 4.3 *Consider the problem (4.19). Theorem (4.2) holds also with a fully implicit discretization.*

Proof: Consider the fully implicit finite difference scheme as introduced above 4.37, 4.38. We put $\mathbf{u} = \theta(\mathbf{v})$ and following the similar steps from the Theorem 4.2, the matrix

$\mathbf{A} = (a_{ij})$ is given by

$$a_{ij} = \begin{cases} 1 + \lambda \left(\sum_{j=1}^{n} \tilde{a}_j(v_{i+1}^{n+1}) \tilde{f}_j'(\eta_{i+1}^{n+1}) + \sum_{j=1}^{n} \tilde{a}_j(v_i^{n+1}) \tilde{f}_j'(\eta_i^{n+1}) \right) & : \quad i = j \\ -\lambda \left(\sum_{j=1}^{n} \tilde{a}_j(v_{i+1}^{n+1}) \tilde{f}_j'(\eta_{i+1}^{n+1}) \right) & : \quad i = j - 1 \\ -\lambda \left(\sum_{j=1}^{n} \tilde{a}_j(v_i^{n+1}) \tilde{f}_j'(\eta_i^{n+1}) \right) & : \quad i = j + 1 \\ 1 + \lambda \left(\sum_{j=1}^{n} \tilde{a}_j(v_1^{n+1}) \tilde{f}_j'(\eta_1^{n+1}) \right) & : \quad i = j = 0 \\ 1 + \lambda \left(\sum_{j=1}^{n} \tilde{a}_j(v_N^{n+1}) \tilde{f}_j'(\eta_N^{n+1}) \right) & : \quad i = j = N \\ 0 & : \quad \text{otherwise.} \end{cases}$$

□

Since $a_j(\mathbf{v})$ is always non-negative for any $\mathbf{v}$, we have $a_j(\mathbf{v^{n+1}}) \geq 0$ and therefore $\mathbf{A}$ is a strictly dominant tridiagonal matrix. Hence from Lemma 4.1, it follows that $\mathbf{A^{-1}} \geq 0$, and following the similar steps as in the Theorem 4.2, it implies the positivity of $\mathbf{v}$, which in turn implies the positivity of $\mathbf{u}$.

4.6 Coupled problem

In this section we only present the coupled drying problem at isothermal conditions. The positivity analysis for this problem is very complicated and we leave it for further work. The complication is due to the fact that the matrix is no longer a strictly tri-diagonal matrix. The matrix will be a block tri-diagonal matrix or a full matrix which is not a tri-diagonal matrix. It is not easy to prove that the inverse of such a matrices is positive, as we did in the scalar case. The coupled system comprises of two primary variables and 19 secondary variables. For the complete set of equations, See Chapter 2.

$$\frac{\partial[\psi S \rho_l + \psi(1-S)\rho_v]}{\partial t} = \frac{\partial}{\partial x}(\rho_g D_{eff} \frac{\partial}{\partial x} y_v + \rho_l K \frac{k_l}{\mu_l} \frac{\partial}{\partial x} P_l + \frac{K}{\mu_g} \rho_v k_g \frac{\partial}{\partial x} P_g), \tag{4.39}$$

$$\frac{\partial[\psi(1-S)\rho_a]}{\partial t} = \frac{\partial}{\partial x}(\rho_g D_{eff} \frac{\partial}{\partial x} y_a + \frac{K}{\mu_g} \rho_a k_g \frac{\partial}{\partial x} P_g). \tag{4.40}$$

We replace the equations of state in the main equations and here we present the equations only in terms of primary variables S and ρ_a.

$$\frac{\partial \theta_1(S)}{\partial t} = \frac{\partial}{\partial x} \left[a1(S, \rho_a) \frac{\partial f1(S, \rho_a)}{\partial x} + b1(S) \frac{\partial g(S)}{\partial x} + c1(S) \frac{\partial h1(S, \rho_a)}{\partial x} \right], \tag{4.41}$$

$$\frac{\partial \theta_2(S,\rho_a)}{\partial t} = \frac{\partial}{\partial x}\left[a1(S,\rho_a)\frac{\partial f2(S,\rho_a)}{\partial x} + c2(S,\rho_a)\frac{\partial h1(S,\rho_a)}{\partial x}\right]. \tag{4.42}$$

Where $\theta_1(S)$ and $\theta_2(S,\rho_a)$ are change in the water content and the air content respectively. $f1(S,\rho_a)$ is mass fraction of vapour, $g(S)$ is the capillary pressure, $h1(S,\rho_a)$ is the total pressure or gas pressure, $f2(S,\rho_a)$ is the mass fraction of air which is 1- $f1(S,\rho_a)$. $a1, b1, c1, c2$ are the diffusive coefficients.
The functions in the equations 4.41 and 4.42 are defined below.

1. $\theta_1(S)$ is given by

$$\theta_1(S) = \begin{cases} (A-B)S + B & : \quad if \quad S \geq S_{crit} \\ A_1S^3 + B_1S^2 + C_1S & : \quad if \quad S < S_{crit} \end{cases}$$

   where A, B, A_1, B_1, C_1 are constants.

2. Mass fraction of vapour $f1(S,\rho_a)$ is given by

$$f1(S,\rho_a) = \begin{cases} \frac{A_5}{A_5+\rho_a} & : \quad if \quad S \geq S_{crit} \\ \frac{k_4S^2+k_5S}{k_4S^2+k_5S+\rho_a} & : \quad if \quad S < S_{crit} \end{cases}$$

   where $A_5, k_4, k_5,$ are constants.

3. Capillary pressure $g1(S)$ is given by

$$g(S) = -k_6e^{(8.4057)10^{-k_0S}}$$

   where k_6 and k_0 are constants.

4. Total pressure $h1(S,\rho_a)$ is given by

$$h1(S,\rho_a) = \begin{cases} A_4 + k_a\rho_a & : \quad if \quad S \geq S_{crit} \\ A_2S^2 + B_2S + k_a\rho_a & : \quad if \quad S < S_{crit} \end{cases}$$

   where A_4, k_a, A_2, B_2 are constants.

5. $\theta_2(S,\rho_a)$ is given by

$$\theta_2(S,\rho_a) = \psi(1-S)\rho_a$$

   where ψ is a constant called porosity.

6. Mass fraction of air $f2(S,\rho_a$ is given by

$$f2(S,\rho_a) = \begin{cases} \frac{\rho_a}{A_5+\rho_a} & : \quad if \quad S \geq S_{crit} \\ \frac{\rho_a}{k_4S^2+k_5S+\rho_a} & : \quad if \quad S < S_{crit} \end{cases}$$

   $f2(S,\rho_a) = 1 - f1(S,\rho_a)$ where A_5, k_4, k_5 are constants.

The diffusion coefficients are defined as follows

1. $a1(S, \rho_a)$ is defined by

$$a1(S,\rho_a) = \begin{cases} D1\frac{A_5+\rho_a}{A_4+k_a\rho_a}(1+2S^3-3S^2) & : \quad if \quad S \geq S_{crit} \\ D1\frac{k_4S^2+k_5S+\rho_a}{A_2S^2+B_2S+k_a\rho_a} & : \quad if \ S < S_{crit} \end{cases}$$

   where $D_1, A_5, A_4, k_a, A_2, B_2$ are constants.

2. $b1(S)$ is given by

$$b1(S) = \begin{cases} k_7S^3 & : \quad if \quad S \geq S_{crit} \\ 0 & : \quad if \ S < S_{crit} \end{cases}$$

   where k_7 is a constant.

3. $c1(S)$ is given by

$$c1(S) = \begin{cases} k_7S^3 + k_8A_5(1+2S^3-3S^2) & : \quad if \quad S \geq S_{crit} \\ k_8(k_4S^2+k_5S) & : \quad if \ S < S_{crit} \end{cases}$$

   where k_7, k_8, k_4, k_5 are constants.

4. $c2(S, \rho_a)$is given by

$$c2(S,\rho_a) = \begin{cases} k_8\rho_aA_5(1+2S^3-3S^2) & : \quad if \quad S \geq S_{crit} \\ k_8\rho_a & : \quad if \ S < S_{crit} \end{cases}$$

   where k_8, A_5 are constants.

One can obtain the expressions for constants by simple algebraic manipulations from the complete set of equations and material constants which are given in Chapter 2. We have just stated the problem here to show the difficulty in analysis for such a complicated problem.

Chapter 5

Numerical simulation in the two dimensional case

In this chapter, the numerical approach in solving a scalar as well as coupled quasilinear parabolic partial differential equations and the corresponding numerical results in two space dimensions are presented. Here we consider the dimensional splitting approach for the numerical computations. The dimensional splitting technique can be used to break down the two dimensional problems into easier one dimensional parts. In the first section of this chapter, we give an introduction to the concept of operator splitting. In Section 5.2, we describe the dimensional splitting approach for solving multidimensional problems. Then two kinds of symmetric splitting methods of Strang are presented. In the final section, we give the numerical results for a scalar and coupled quasilinear parabolic partial differential equations in two dimensions using dimensional splitting.

5.1 Operator splitting

To motivate the concept of operator splitting, let us consider the following ordinary differential equation

$$\frac{\partial u}{\partial t} = (a+b)u, \tag{5.1}$$

with initial condition $u(0) = u_0$ and $a, b \in \mathbb{R}$. The exact solution of this problem is given by

$$u(t) = e^{t(a+b)}u_0 = e^{ta}e^{tb}u_0. \tag{5.2}$$

Using operator splitting, the solution of the initial value problem (5.1) can also be obtained by first solving $\frac{\partial u}{\partial t} = bu$ with u_0 as the initial condition which has the solution given by $e^{tb}u_0$. Then, solve $\frac{\partial u}{\partial t} = au$ with respect to initial condition $e^{tb}u_0$. It can be seen in the following two steps

Step 1 Solve

$$\begin{aligned} \frac{\partial u}{\partial t} &= bu, \\ \text{with initial condition: } u(0) &= u_0. \end{aligned} \tag{5.3}$$

The solution to the problem (5.3) is given by $u = e^{tb}u_0$ which will become the initial condition for the next step.
Step 2 Solve

$$\begin{aligned} \frac{\partial u}{\partial t} &= au, \\ \text{with initial condition: } u(0) &= e^{tb}u_0. \end{aligned} \tag{5.4}$$

The solution of (5.4) is given by $u = e^{ta}e^{tb}u_0$. The solution $u(t)$ thus obtained from step 2 is the solution of the problem (5.1). This approach of obtaining the solution in two steps is called operator splitting.

5.1.1 Extention to vectors and operators

The above interpretation of operator splitting, however does not extend to vector systems. Let $\mathbf{u} : [0, T] \times \Omega \to \mathbb{R}$, $\Omega \subseteq \mathbb{R}^m$. Now, let us consider the following system of ordinary differential equations

$$\frac{\partial \mathbf{u}}{dt} = (\mathbf{A} + \mathbf{B})\mathbf{u}, \tag{5.5}$$

with initial condition $\mathbf{u}(t_0) = \mathbf{u_0}$, where $\mathbf{u} \in \mathbb{R}^m$ is a vector, $\mathbf{A}$, $\mathbf{B} \in \mathbb{R}^{m\times m}$ are scalar matrices. The exact solution of the problem (5.5) is given by $\mathbf{u} = e^{(\mathbf{A}+\mathbf{B})\mathbf{t}}\mathbf{u_0}$. Using operator splitting, by proceeding in two steps as explained for the problem (5.1), the solution of (5.5) is approximated by $\mathbf{u} = e^{\mathbf{At}}e^{\mathbf{Bt}}\mathbf{u_0}$. Here we can see $e^{(\mathbf{A}+\mathbf{B})\mathbf{t}}\mathbf{u_0} \neq e^{\mathbf{At}}e^{\mathbf{Bt}}\mathbf{u_0}$, unless the matrices $\mathbf{A}$ and $\mathbf{B}$ commute.

Let us extend the study of splitting to the concept of operators and space dimensions. Consider the following linear partial differential equation

$$\frac{\partial \mathbf{u}}{\partial t} = \Delta\mathbf{u} = (\Delta_1 + \Delta_2)\mathbf{u}. \tag{5.6}$$

Here $\Delta = \Delta_1 + \Delta_2$ is a spatial differential operator. As a particular case, $\Delta_1 = \frac{\partial^2}{\partial x^2}$ and $\Delta_2 = \frac{\partial^2}{\partial y^2}$. The solution of the linear partial differential equation (5.6) can also be written in the following exponential form

$$\mathbf{u}(x, y, t) = e^{t\Delta}\mathbf{u}(x, y, 0),$$

where Δ is independent of t. The exponential of the operator Δ can be interpreted using the Taylor's series expansion of $\mathbf{u}$ in terms of t,

$$\mathbf{u}(x,y,t) = \mathbf{u}(x,y,0) + t\Delta\mathbf{u}(x,y,0) + \frac{t^2}{2}\Delta^2\mathbf{u}(x,y,0) + \ldots\ldots\ldots = e^{t\Delta}\mathbf{u}(x,y,0).$$

Again we can observe, $\mathbf{u}(x,y,t) = e^{t(\Delta_1+\Delta_2)}\mathbf{u}(x,y,0) \neq e^{t\Delta_1}e^{t\Delta_2}\mathbf{u}(x,y,0)$, unless the operators Δ_1 and Δ_2 commute.

Splitting error

Consider the Taylor's series expansions of $e^{t(\Delta_1+\Delta_2)}\mathbf{u}(x,y,0)$ and $e^{t\Delta_1}e^{t\Delta_2}\mathbf{u}(x,y,0)$, then we get

$$e^{t(\Delta_1+\Delta_2)}\mathbf{u}(x,y,0) = (\mathbf{I}+t(\Delta_1+\Delta_2)+\frac{t^2}{2}(\Delta_1^2+\Delta_1\Delta_2+\Delta_2\Delta_1+\Delta_2^2)+\ldots\ldots\ldots)\mathbf{u}(x,y,0), \quad (5.7)$$

and

$$\begin{aligned} e^{t\Delta_1}e^{t\Delta_2}\mathbf{u}(x,y,t) &= (\mathbf{I}+t\Delta_1 + \frac{t^2}{2}\Delta_1^2 + \ldots\ldots\ldots)(\mathbf{I} + t\Delta_2 + \frac{t^2}{2}\Delta_2^2 + \ldots\ldots\ldots)\mathbf{u}(x,y,0) \\ &= (\mathbf{I} + t(\Delta_1+\Delta_2) + \frac{t^2}{2}(\Delta_1^2 + 2\Delta_1\Delta_2 + \Delta_2^2) + \ldots\ldots\ldots)\mathbf{u}(x,y,0). \end{aligned} \quad (5.8)$$

Substracting (5.8) from (5.7), we get

$$\{e^{t(\Delta_1+\Delta_2)} - e^{t\Delta_1}e^{t\Delta_2}\}\mathbf{u}(x,y,0) = [\frac{t^2}{2}(\Delta_2\Delta_1 - \Delta_1\Delta_2) + O(t^3)]\mathbf{u}(x,y,0). \quad (5.9)$$

The difference between the two expressions is $\mathrm{O}(t^2)$, unless $\Delta_1\Delta_2 = \Delta_2\Delta_1$. Obviously this splitting is a first order process unless Δ_1 and Δ_2 commute. The error is often called as splitting error. It follows that for commuting operators the splitting is exact, it leaves no error. However the factorization almost works when t is small, hence, we can replace t by a small time increment δt to obtain the solution

$$\mathbf{u}(x,y,\delta t) = e^{\delta t(\Delta_1+\Delta_2)}\mathbf{u}(x,y,0) = [e^{\delta t\Delta_1}e^{\delta t\Delta_2} + \frac{\delta t^2}{2}(\Delta_2\Delta_1 - \Delta_1\Delta_2) + O(\delta t^3)]\mathbf{u}(x,y,0).$$

To obtain a numerical method, we discretize the spatial operators Δ_1 and Δ_2, neglect the local error terms and then use the resulting scheme from time step to time step. Thus we get the numerical scheme

$$\mathbf{u}^{n+1} = e^{\delta t\Delta_{1,\delta}}e^{\delta t\Delta_{2,\delta}}\mathbf{u}^n,$$

where $\Delta_{1,\delta}$ and $\Delta_{2,\delta}$ are discrete numerical approximations of Δ_1 and Δ_2. In literature this technique is often called as method of fractional steps or operator splitting or dimensional splitting. The primary advantage of this technique is, if each of the operators are stable,

then the composition is stable. If the operators $\Delta_{1,\delta}$ and $\Delta_{2,\delta}$ satisfy the von Neumann conditions, then the combined scheme is stable since

$$\|e^{\delta t \Delta_{i,\delta}}\| \leq 1 + c_i \delta t, \qquad i = 1, 2$$
$$\|\mathbf{u}^{n+1}\| \leq \|e^{\delta t \Delta_{1,\delta}}\| \|e^{\delta t \Delta_{2,\delta}}\| \|\mathbf{u}^n\| \leq (1 + c\delta t)\|\mathbf{u}^n\|.$$

With operator splitting, the local error is $O(\delta t^2)$ unless the operators Δ_1 and Δ_2 commute. If the operators commute then the local error is $O(\delta t^3)$.

5.2 Dimensional splitting

The dimensional splitting technique is a numerical method to extend the one dimensional case to two or higher dimensions. The primary advantage of dimensional splitting is the efficiency of the numerical scheme. The application of standard implicit schemes to any time dependent partial differential equations results in solving the linear system of equations of the form $\mathbf{Az} = \mathbf{b}$, where $\mathbf{A}$ is any matrix, $\mathbf{z}$ is the unknown vector and $\mathbf{b}$ is the right hand side vector. The matrix $\mathbf{A}$ is a tridiagonal matrix for a scalar parabolic problem in one dimension, and it is no longer a tridiagonal matrix in two and higher dimensions. Thus the size of the matrix $\mathbf{A}$ increases with dimension of any scalar or system of partial differential equations. Hence, it is worthwhile to note that the efficiency of inverting the matrix $\mathbf{A}$ must be increased, when moving to higher dimensions. The inversion of a large matrix in higher dimensions at each time step of the numerical process is costly and this can be overcome by using splitting methods. Here dimensional splitting can be used to break the two dimensional case into easier one dimensional parts. These methods take less memory and more efficient. These schemes are motivated by ADI scheme of Peaceman and Rachford [31].

Let $\Omega = [a, b] \times [c, d]$, $a, b, c, d > 0$ be any two dimensional domain and $u : [0, T] \times \Omega \to \mathbb{R}$. Consider the following partial differential equation

$$\frac{\partial \theta(u)}{\partial t} = \frac{\partial}{\partial x}\left[\sum_{j=1}^{k}\left(a_j(u)\frac{\partial f_j(u)}{\partial x}\right)\right] + \frac{\partial}{\partial y}\left[\sum_{j=1}^{k}\left(b_j(u)\frac{\partial g_j(u)}{\partial y}\right)\right] \text{ on } [0, T] \times \Omega, \tag{5.10}$$

with initial condition: $u(t_0, x, y) = u_0$. Using dimensional splitting, the problem (5.10) can be solved in two steps. In each step a one-dimensional problem is solved.
Step 1: Solve

$$\frac{\partial \theta(u^*)}{\partial t} = \frac{\partial}{\partial x}\left[\sum_{j=1}^{k}\left(a_j(u^*)\frac{\partial f_j(u^*)}{\partial x}\right)\right] \text{ on } [0, T] \times \Omega, \tag{5.11}$$

with initial condition $u^*(t_0, x, y) = u_0$, let the solution to the step 1 at time t_n is $u^*(t_n, x, y)$
step 2: Solve

$$\frac{\partial \theta(u)}{\partial t} = \frac{\partial}{\partial y}\left[\sum_{j=1}^{k}\left(b_j(u)\frac{\partial g_j(u)}{\partial y}\right)\right] \text{ on } [0, T] \times \Omega, \tag{5.12}$$

with initial condition $u(t_0) = u^*(t_n)$. The solution $u(t_n)$ obtained from step 2 is the approximate solution of the problem (5.10). The same procedure can be applied to any boundary value problem.

5.2.1 Discretization in two dimensions

In the present work, we use a cell centered finite volume scheme for discretization. For the two dimensional case, we consider only implicit time discretization. A square of side length L is considered as the two dimensional domain. Let N be the number of cells or control volumes in each direction, then the total number of cells in the complete two dimensional domain is N^2. We take $h = \frac{L}{N}$ as the fixed space step size in each direction. The area of each control colume or cell is h^2. The area of the boundary cells is $\frac{h^2}{2}$, and the area of the corner cells is $\frac{h^2}{4}$. For the treatment of boundary control volumes in different ways, see Patankar [30]. Applying a cell centered finite volume scheme to (5.10) with implicit time discretization results in solving a nonlinear system of equations with a large matrix of size $N^2 \times N^2$, whereas, using dimensional splitting it is enough to deal with a matrix of size $N \times N$ in each direction. Here we do not present the cell centered finite volume discretization for a two dimensional problem, for details see Patankar [30] or Versteeg and Malalasekera [46]. Figure 5.1 represents a two dimensional control volume mesh illustrating the dimensioinal splitting approach. We first solve the solution of a given problem in x-direction (step 1) which are shown in green lines in Figure 5.1, then we solve in y-direction (step 2) which are shown in yellow lines. The discretization in the one dimensional case is given in Section 3.1.

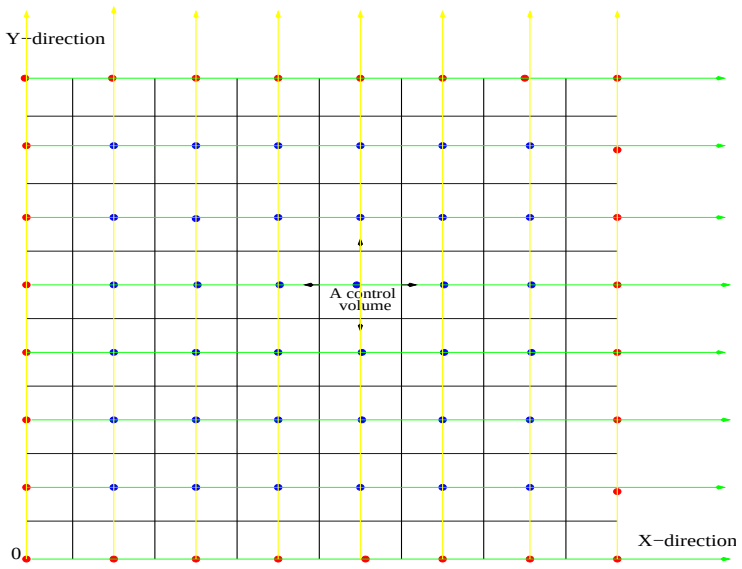


Figure 5.1: A control volume mesh with dimensional splitting.

The main advantages of these methods are

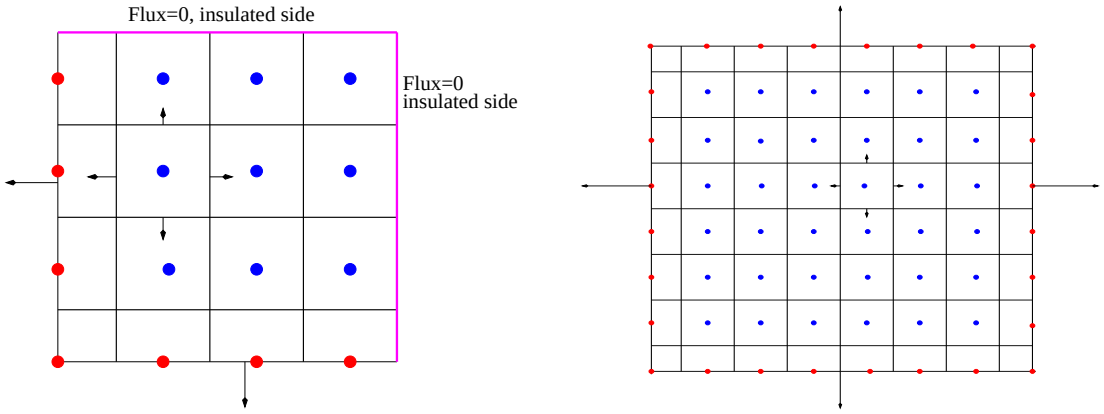
1. one dimensional time step restrictions for two dimensional problem,

2. solving matrices of smaller size than those matrices in the case of direct two dimensional discretization.

Off course the dimensional splitting is restricted to structured meshes only. For multi-dimensional PDEs, the dimensional splitting works in such a way that all computations become effectively one dimensional. For this reason these methods are also known as locally one dimensional.

5.2.2 The domain reduction using symmetry

The right hand Figure 5.2 represents the complete two dimensional discretized domain. For the problems with symmetric solutions, it is enough to solve the domain given by left hand side in Figure 5.2 by insulating two ends of the boundary i.e., by making the flux equal to zero at the insulated boundary ends. The reduced domain is one fourth of the complete domain, and therefore an essential gain is attained in the computational cost. We considered the domain represented by Figure 5.2(a) in our numerical computations.



(a) a square of side length $\frac{L}{2}$ (b) a square of side length L

Figure 5.2: A two dimensional mesh with isolated ends to make use of symmetry and the right figure is a complete domain.

5.3 Splitting methods

Here we present two kinds of splitting procedures of Strang for a two dimensional problem. To explain these procedures, we consider the following linear partial differential equation

$$\frac{\partial u}{\partial t} = \triangle u. \tag{5.13}$$

Here $\triangle = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. Split $\triangle = \triangle_1 + \triangle_2$ where $\triangle_1 = \frac{\partial^2}{\partial x^2}$ and $\triangle_2 = \frac{\partial^2}{\partial y^2}$. We use the following notation in order to explain the two kinds of splitting procedures which are used in the present numerical simulation. The solution to (5.13) is given by $u_{n+1} = \left(e^{\tau(\triangle_1+\triangle_2)}\right) u_n$. We denote $u_{n+1} = u(t_{n+1})$ and $\tau = t_{n+1} - t_n$. Using dimensional splitting, the solution is approximated by $u_{n+1} = \left(e^{\tau\triangle_1} e^{\tau\triangle_2}\right) u_n$. The notation can be understood in the way that $e^{\tau\triangle_2} u_n$ indicates step 1 which is (5.11). Let u^* be the solution to this step 1. Then $e^{\tau\triangle_1} u^*$ is the step 2 given by (5.12) which gives the solution to (5.13). With this notation, the two kinds of splitting procedures are given below.

Method 1

The following splitting procedure is suggested by Strang,

$$u_{n+1} \approx \left(e^{\tau\triangle_1} e^{\tau\triangle_2}\right) u_n. \tag{5.14}$$

The point of attaining symmetric solutions is important in various problems like image processing, drying technology etc., see Barash and Kimmel [2]. As we mentioned earlier, if the operators are not commutative, the order of changing the splitting not only effects the solution, importantly it loses the symmetry of the problem. Though this might not be a severe problem in solving linear problems, but the loss of symmetry in the solution is severe in the case of nonlinear problems. So, we alter the order of splitting suggested by Strang [41] and get

$$u_{n+1} \approx \left(e^{\tau\triangle_2} e^{\tau\triangle_1}\right) u_n. \tag{5.15}$$

Thus the solution of (5.13) is approximated by

$$u_{n+1} = \frac{1}{2}\left(e^{\tau\triangle_1} e^{\tau\triangle_2} + e^{\tau\triangle_2} e^{\tau\triangle_1}\right) u_n. \tag{5.16}$$

Method 2

A second order splitting is given by

$$u_{n+1} \approx \left(e^{\frac{\tau}{2}\triangle_1} e^{\frac{\tau}{2}\triangle_2}\right)\left(e^{\frac{\tau}{2}\triangle_2} e^{\frac{\tau}{2}\triangle_1}\right) u_n = \left(e^{\frac{\tau}{2}\triangle_1} e^{\tau\triangle_2} e^{\frac{\tau}{2}\triangle_1}\right) u_n. \tag{5.17}$$

Interchanging the order of splitting after each step will lead to symmetry and better accuracy. This idea of splitting has been proposed by Strang [42] and Marchuk [25]. To attain a better symmetry, the solution of (5.13) is given by

$$u_{n+1} = \frac{\left(e^{\frac{\tau}{2}\triangle_1} e^{\tau\triangle_2} e^{\frac{\tau}{2}\triangle_1}\right) u_n + \left(e^{\frac{\tau}{2}\triangle_2} e^{\tau\triangle_1} e^{\frac{\tau}{2}\triangle_2}\right) u_n}{2}. \tag{5.18}$$

Though splitting by the second method gives better accuracy, it is expensive because it needs to evaluate the solution by one time step more than the first method.

5.4 Numerical results

In this section, we present the numerical observations in two space dimensions for a scalar and coupled quasilinear parabolic equations. These problems govern the drying process. At each time step, we apply Newton's method in linearizing the nonlinear system of equations and then the linear system of equations are solved by BiCGSTAB with ILU preconditioning.

Scalar problem

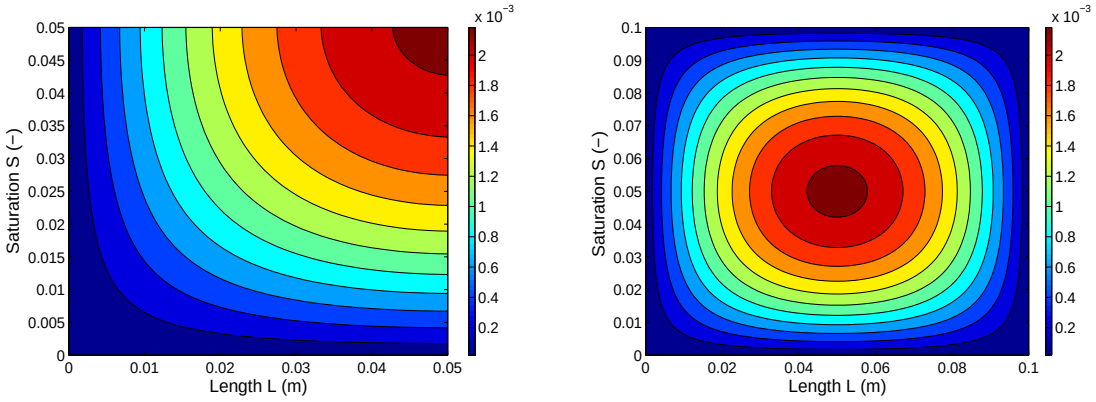
The governing partial differential equation is given by equation (2.1). The initial and boundary conditions are given in Section 3.3.1. The primary variable is S and the complete equations of state are presented in Section 3.3.2.

Coupled problem

The coupled system of quasilinear parabolic partial differential equations are given by equations (2.1) and (2.2). The primary variables are S and ρ_a. For the initial, boundary conditions and the complete set of equations of state, see Section 3.4.

5.4.1 Simulation results: scalar problem

We consider a concrete material of square of length L=0.1m. Figure 5.3(a) represents the domain used for numerical computations. It is only one fourth of the total square of side length 0.1m, with two sides being insulated. Figure 5.3(b) represents the solution for the complete concrete material of length 0.1m. These profiles are drawn at a drying time 5.11 days.



(a) a square of side length $\frac{L}{2}$ (b) a square of side length L

Figure 5.3: Saturation profiles.

The following table represent the CPU times taken by the two splitting methods given in

Section 5.3. The programming has been done in languages C and C++. These CPU times are calculated on a Linux machine of 1GB ram and 1.6GHz Processor at the end of real drying time of 5.11 days.

Method.	CPU time (min)
1	67.84
2	101.05

Table 5.1: Efficiency table using Strang's splittings.

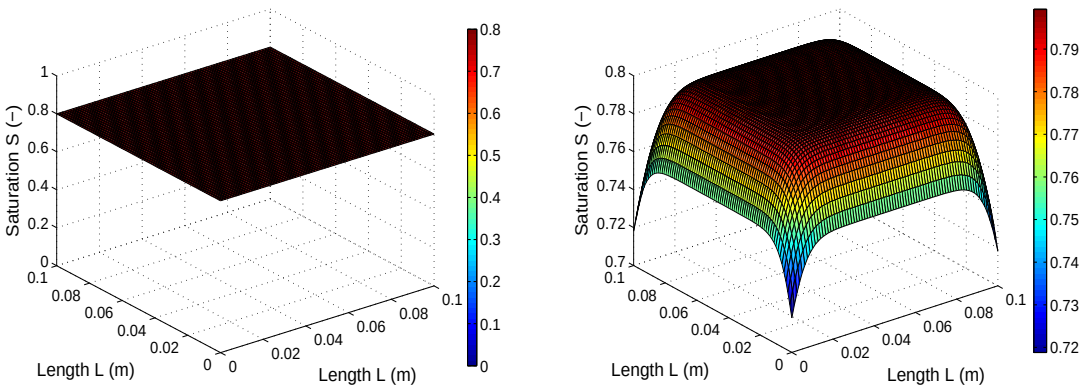
Here we observe that the first method is more efficient than the second method and the accuracy is also as good as the second method. For the scalar problem we stop the simulation at a real drying time of 5.11 days. With these observations we now proceed to the coupled problem which govern the exact physics of the drying problem at isothermal conditions and we use the first method of splitting.

5.4.2 Simulation results: coupled problem

For the coupled problem also, we consider a concrete material of square of length 0.1m. The number of cells are 80×80. The initial average saturation is 0.8 and the initial air density is 1.16103 $\frac{kg}{m^3}$.

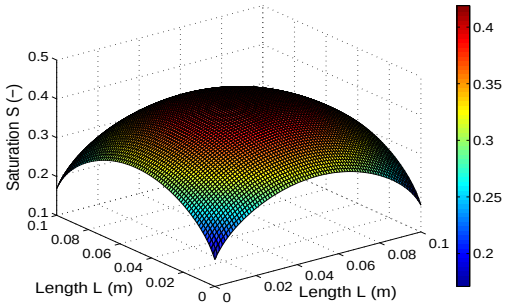
Saturation profiles at different drying times

The following sequence of figures represent saturation profiles at different drying times.

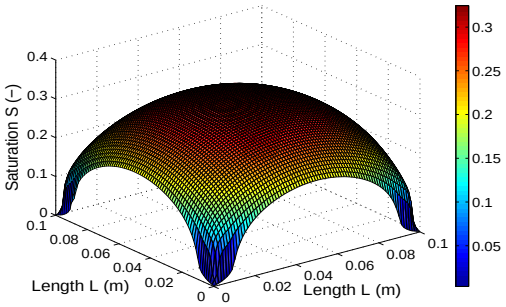


(a) initial saturation (b) at 18 min

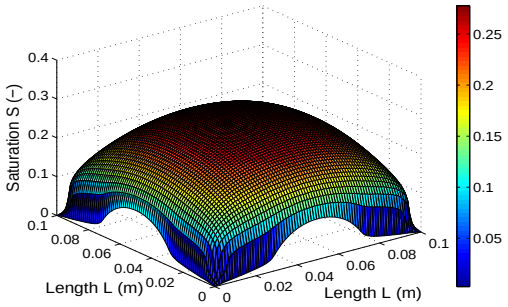
Figure 5.4: Saturation profiles at different drying times.



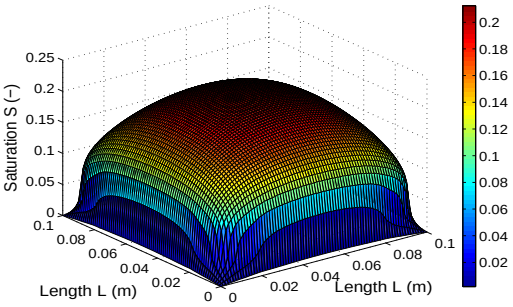
(c) at 14.45 hours



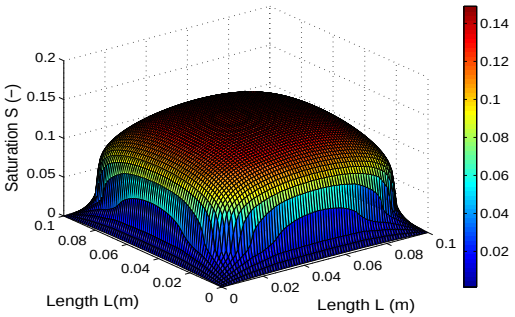
(d) at 17.69 hours



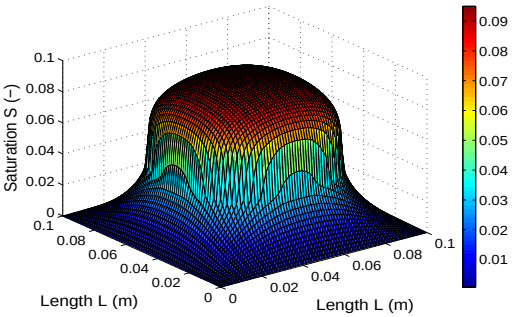
(e) at 19.62 hours



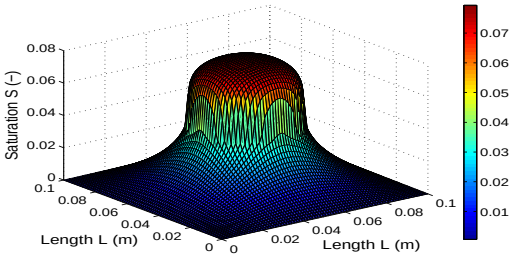
(f) at 23.56 hours



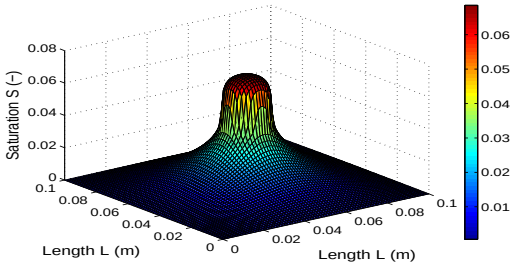
(g) at 1.43 days



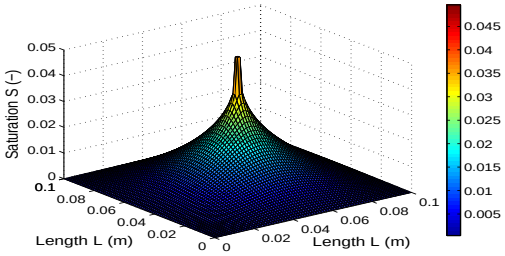
(h) at 3.46 days



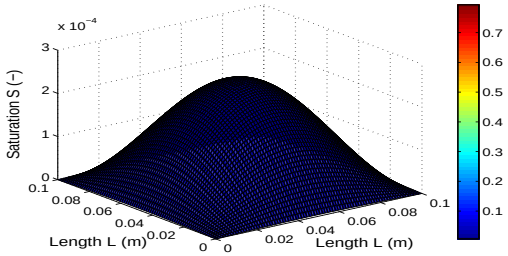
(i) at 5.32 days



(j) at 6.89 days

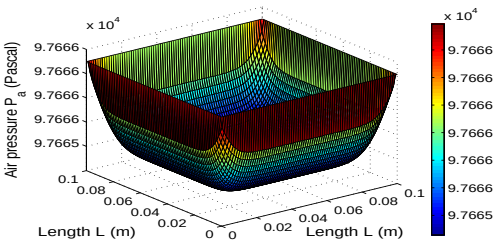


(k) at 7.60 days

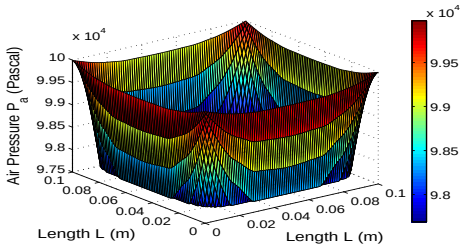


(l) at 13.02 days

Air pressure profiles at different drying times

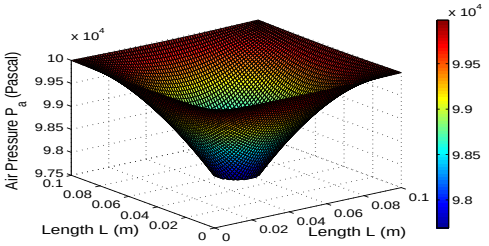


(a) at 18 min

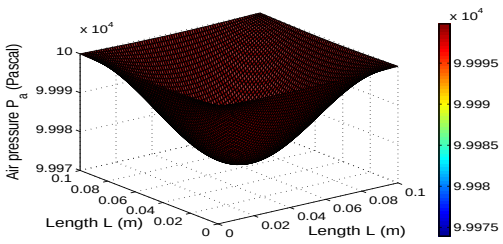


(b) at 1.43 days

Figure 5.5: Air Pressure at different drying times.



(i) at 6.89 days



(j) at 13.02 days

With our numerical simulations in two dimensions using dimensional splitting approach, we attain the physical trends of drying with good drying rates in an efficient way. We do not present here the profiles of all the other variables, but the trends are similar as in the one dimensional case. The drying time required to reduce the saturation from an average initial saturation of 0.8 to 10^{-4} for a concrete material of square side length 0.1m is 13.02 days. The CPU time consumed is 12.81 hours. The coding has been written in C programming language and the CPU time is calculated on a Linux machine with 1GB Ram and 1.6GHz Processor. Through the sequence of figures in Figure 5.4, it can be noticed that the saturation decreases with increasing time. The air pressure in the porous media increases with increasing time and it reaches the atmoshpheric gas pressure at the end of the drying process, which can be observed in the sequence of figures in Figure 5.5. The coupled problem is highly stiff as we have noticed in the one dimensional case. Also, we have observed similar time step reductions in the two dimensional case. During the first drying state, we take the time step size as 180 and in later states we reduce the time step size to 5. During the intermediate state of drying, the problem becomes highly stiff in local regions where a change in drying state occur. Figures 5.4 (d), (e), (f), (g), (h), (i), (j), (k) represent the saturation profiles during the intermediate state of drying. In these figures, we can see the steep gradients at local regions at which the change in drying state causes rapid reduction or increase in the variables. Through dimensional splitting, the stability is maintained with the same time step restrictions as in the one dimensional case and this approach is efficient than the direct discretization.

Chapter 6

Time stepping strategies

In this chapter, we present some efficient time stepping strategies for solving a time dependent nonlinear parabolic partial as well as ordinary differential equations in one dimensional case. Given a time dependent partial differential equation, discretization in space leads to a semi-discrete system of ODEs. To solve these system of ODEs, there exist several time integration methods which can be explicit and implicit Runge-Kutta methods, linearly implicit methods such as Rosenbrock methods, W-methods etc. In this chapter we study the drying problem using linearly implicit time stepping methods. And we investigate suitable time stepping strategies in order to increase the efficiency of the numerical scheme in solving the equations governing the drying problem at isothermal conditions. For this purpose, we take the method of lines approach. In the first section of this chapter, we discuss about linearly implicit methods with variable time step size control for solving ordinary differential equations. In the next section, we describe a partitioning strategy and the corresponding numerical observations for a scalar quasilinear parabolic problem will be presented. In the final section of this chapter, a local time stepping method is described and the numerical observations for a coupled quasilinear parabolic problem are presented.

6.1 Linearly implicit methods

Linearly implicit methods have proven succesful in solving many stiff ODE, DAE and PDE problems efficiently. The primary advantage of these methods is, only linear system of algebraic equations have to be solved. In the case of an implicit scheme, the implementation of an efficient nonlinear solver in solving the system of nonlinear algebraic equations is the main problem. Rosenbrock put forth an important consideration to derive a stable formulas by working the Jacobian matrix directly into the integration formula. It means that linearly implicit methods avoid Newton iterations, which are required in general while solving any nonlinear PDEs or ODEs. Moreover these methods with embedding provides a cheap local error estimator which is needed for time step control. In this section we present two types of linear multistep methods which are W-methods and Rosenbrock methods. More details on linearly implicit methods can be found in Hairer and Wanner [18], Deuflhard

and Bornemann [14], Hundsdorfer and Verwer [20].

6.1.1 W-method

Let us consider the following ODE problem

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}(t, \mathbf{u}) \tag{6.1}$$

in the time interval $[t_0, t_{end}]$. We denote $\tau_n = t_{n+1} - t_n$ as the time step size at n^{th} discrete time interval in the sequence $t_0 < t_1 < t_2 < .....t_N = t_{end}$ and $\mathbf{u}^n$ denotes the numerical solution at time t_n. At the n^{th} integration step, a q-stage W-method of order p has the following form

$$(\mathbf{I} - \tau_n \gamma \mathbf{J_c})\mathbf{k}_j = f(t_n + \tau_n a_j, \mathbf{u^n} + \tau_n \sum_{l=1}^{j-1} b_{lj}\mathbf{k}_l) + \sum_{l=1}^{j-1} c_{lj}\mathbf{k}_l, \text{ for } j = 1, 2, ...q, \tag{6.2}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \tau_n \sum_{l=1}^{q} d_l \mathbf{k}_l. \tag{6.3}$$

A W-method with embedding of order $\hat{p} \neq p$ has the form

$$\hat{\mathbf{u}}^{n+1} = \mathbf{u}^n + \tau_n \sum_{l=1}^{q} \hat{d}_l \mathbf{k}_l. \tag{6.4}$$

The coefficients $\gamma, a_j, b_{jk}, c_{jk}, d_j$ and $\hat{d}_j$ are chosen such that the local error of $\mathbf{u}$ is of order τ_n^{p+1}, and the local error of $\hat{\mathbf{u}}$ is of order $\tau_n^{\hat{p}+1}$. These orders are independent of the Jacobian matrix $\mathbf{J_c}$. We use a 3 stage W-method, with orders p=2 and $\hat{p} = 1$. The corresponding coefficients in the method are given by

$\gamma = 1 - \frac{\sqrt{2}}{2}$,
$a_1 = 0, a_2 = 1, a_3 = 1,\ b_{12} = 1, b_{13} = 1, b_{23} = 0,$
$c_{12} = -2 - \sqrt{2}, c_{13} = -1, c_{23} = -1 + \sqrt{2},$
$d_1 = 1, d_2 = \frac{1}{2} - \frac{\sqrt{2}}{2}, d_3 = \frac{1}{2},$
$\hat{d}_1 = \frac{9}{10} - \frac{\sqrt{2}}{20}, \hat{d}_2 = \frac{9}{20} - \frac{11}{20}\sqrt{2}, \hat{d}_2 = \frac{11}{20} + \frac{\sqrt{2}}{20}.$

Variable time step control

In many applications, ODE problems are solved by integrating with variable time step sizes. Users have to specify a prescribed tolerance and specific norm for calculating the error, and then the code automatically adjusts the time step τ to the local variation in the computed solution to meet a certain local error criterion in the specified norm. This approach leads to smaller time steps in the regions of rapid variation and larger in the slow

variation regions. This result to an efficient numerical computation.

A W-method with embedding provides a cheap local error estimator which is needed for the time step control. By comparing the solution with two methods of order p and $\hat{p}$, we get an estimate for the local error. After the n^{th} integration step the value

$$err = \|\mathbf{u}^n - \hat{\mathbf{u}}^n\|_\infty, \tag{6.5}$$

gives an estimate for the local temporal error. We take the maximum norm because it aims at errors below the given tolerance at all spatial grid points. After obtaining the solutions $\mathbf{u}$ and $\hat{\mathbf{u}}$, there arises two cases. If $err <=$ tolerance, we accept the time step and continue the integration from t_n to t_{n+1}. If $err >$ tolerance, we reject the time step and the time step is repeated with a smaller step size τ_{new}. The new time step size τ_{new} is given by

$$\tau_{new} = \zeta \tau_{old} \left(\frac{tol}{err} \right)^{1/p}, \tag{6.6}$$

where tol is the desired tolerance specified by the user and $0 < \zeta < 1$ is the safety factor to make the estimate conservative which avoids the repeated time step rejections. This (6.6) type of time step selection is standard, we refer to Shampine [40].

6.1.2 Rosenbrock method

These methods are linearly implicit which are of Runge-Kutta type methods for solving stiff ODEs. These methods are named after Rosenbrock, who has introduced the methods of this kind. For more details of these methods, we refer to Lang [23] and Verwer [20]. There exist many Rosenbrock methods in the literature, a q-stage Rosenbrock method for solving the ODE system (6.1) is given by

$$\mathbf{k}_i = \tau_n f(\mathbf{u}^n + \sum_{j=1}^{i-1} \alpha_{ij} \mathbf{k}_j) + \tau_n \mathbf{J_c} \sum_{j=1}^{i} \gamma \mathbf{k}_j, \text{ for } i = 1, 2, ...q, \tag{6.7}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \tau_n \sum_{l=1}^{q} d_l \mathbf{k}_l. \tag{6.8}$$

The coefficients d_l, α_{ij} and γ are chosen to attain a desired order of stability and consistency. In our numerical computation, we use a two stage Rosenbrock method known as ROS2 which has the following form

$$\begin{aligned} (\mathbf{I} - \tau_n \gamma \mathbf{J_c}) \mathbf{k}_1 &= \tau_n f(\mathbf{u}^n), \\ (\mathbf{I} - \tau_n \gamma \mathbf{J_c}) \mathbf{k}_2 &= \tau_n f(\mathbf{u}^n + \mathbf{k}_1) - 2\mathbf{k}_1, \\ \mathbf{u}^{n+1} &= \mathbf{u}^n + d_1 \mathbf{k}_1 + d_2 \mathbf{k}_2. \end{aligned} \tag{6.9}$$

Here $d_1 = \frac{3}{2}$ and $d_2 = \frac{1}{2}$. The method (6.9) is of order 2 for any choice of γ. The method is A-stable if $\gamma \geq \frac{1}{4}$ and L-stable if $\gamma = 1 \pm \frac{\sqrt{2}}{2}$. In our computational work, we take

$\gamma = 1 - \frac{\sqrt{2}}{2}$ which gives L-stability. At each time step, these methods solve a system of linear algebraic equations with the matrix $(\mathbf{I} - \tau_n \gamma \mathbf{J_c})$.

Variable time step control

We use the embedded first order Rosenbrock method to estimate the local temporal error. The first order Rosenbrock method is given by

$$\hat{\mathbf{u}}^{n+1} = \mathbf{u}^n + \mathbf{k}_1. \tag{6.10}$$

The local error at n^{th} integration step is estimated by $err = \|\mathbf{u}^n - \hat{\mathbf{u}}^n\|_\infty$ and the time step size for $(n+1)^{\text{th}}$ integration step can be estimated by equation (6.6).

6.2 Governing equations and discretization

We consider two problems which are a scalar and a coupled quasilinear time dependent parabolic partial differential equations. We consider the method of lines approach where the space and time discretizations are done seperately. First we discretize in space, which leads to a semi-discrete DAE system. Then we use linearly implicit methods to integrate the ODE system in time. In our computational work, we consider an embedded W-method of order 3 and a 2-stage Rosenbrock method for time integration. The scalar and coupled problems are given as follows.

6.2.1 Scalar parabolic problem

The governing partial differential equation with the initial and boundary conditions are given in Section 3.3 and the equations of state can be found in 3.3.2. The semi-discrete ODE system obtained after space discretization is given by equations (3.28), (3.29) and (3.30) in Section 3.5.1 with $a_1(S, \rho_a) = a_1(S)$. For the scalar problem, we take the following expressions for P_c, k_l and k_g as a numerical test case, instead of those given in Section 3.3.2.

$$P_c = 40\sigma e^{(8.4057)10^{-0.3476X}}, \tag{6.11}$$

$$k_l = S^3 \text{ if } X \geq X_{crit} \text{ else } k_l = 0, \tag{6.12}$$

$$k_g = 1 + (2S - 3)S^2 \text{ if } X \geq X_{crit} \text{ else } k_g = 1. \tag{6.13}$$

6.2.2 Coupled parabolic problem

The governing partial differential equations for the coupled problem with the initial and boundary conditions are given in Section 3.4.1. The equations of state are given in 3.3.2 and by equation (3.21). This governing system represent the isothermal drying of porous

media. The DAE system of equations obtained after discretization in space are given by equations (3.28), (3.29), (3.30), (3.31) and (3.32) in Section 3.5.1.

Solution domain

For both the scalar and coupled problems we consider the drying of a concrete material of length L=0.04m. To make use of symmetry, the right hand side boundary is insulated by making the flux equal to zero at this boundary. Hence it is sufficient to perform numerical simulations for length L=0.02m with one end of the boundary being insulated. The space step size is given by h=$\frac{0.02}{N}$, N being the number of cells. The construction of Jacobian matrix has to be done in great care, as the drying problem under consideration is highly nonlinear and also due to many nonlinear equations of state with switching between states of drying. In such cases, computation of analytical Jacobain is quite difficult. So we compute the Jacobian matrix numerically using finite differences which is explained in Section 3.3.6.

6.3 Partitioning method

In some cases, stiffness exist at some specific regions rather than being stiff at the entire domain, which means only a part of the components of the semi-discrete ODE system is stiff. Several examples of reaction-diffusion systems, diffusion systems exist which are locally stiff. For instance in the scalar and coupled drying problems which are mentioned in Section 6.2, the problem becomes highly stiff at some local regions during the intermediate state of the drying process. The high stiffness occurs locally at the regions where a node changes its state from the first drying state to the second drying state. The local temporal error is high in such regions using large time steps. Due to these high local temporal errors, the time steps are rejected, and we proceed to the next time level using a reduced time step size estimated by equation (6.6) for the entire domain.

Construction of the Jacobian matrix and solving large system of linear equations make the code expensive in CPU time using an implicit scheme. On the other hand, though explicit schemes do not lead to system of nonlinear algebraic equations but these schemes demand a small time step size for stability conditions. Partitioning exploit the fact that stiffness is only affecting a part of the system in such a way that only this part is solved by a stable implicit scheme while the remaining nonstiff part can be solved by an explicit scheme. For interested reader on partitioning type methods, we refer to [8, 10, 15, 19, 28, 48].

Explicit-Implicit Switching

W-methods are appropriate choice for partitioning. A W-method (6.3) reduces to an explicit Runge-Kutta scheme if the Jacobian $\mathbf{J_c}$=0 in equation (6.2). Therefore an easy switching between an explicit and an L-stable implicit method is possible by a change of

the matrix $\mathbf{J_c}$. We identify the local regions where the local temporal error is greater than the prescribed tolerance by using equation (6.5). Then, consider an implicit scheme in the regions where this error is high and an explicit scheme in the remaining parts. For instance, consider a one dimensional domain of length L as shown in the following figure.

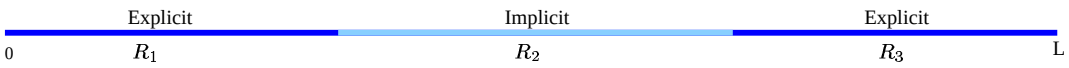


Figure 6.1: A one dimensional domain with partitioning.

Let R_2 be the region where the local error is more than the given tolerance, then we use an implicit scheme only in R_2 and an explicit scheme in R_1 and R_2. Thus, this approach results in solving a small system of linear algebraic equations and therefore an essential gain is attained in the computational cost.

6.3.1 Numerical observations using partitioning strategy

Scalar parabolic problem

We use a fully implicit scheme during the first and second drying states and partitioning (explicit and implicit) during the intermediate state. Here we consider W-method for time discretization. Figures 6.2(a) and 6.2(b) represent the saturation profile and the corresponding local temporal error at a drying time of 24.39 hours during the intermediate state of drying. Figure 6.2(a) represents the solution of a concrete material of length 0.02m, with one end of the boundary being insulated. The solution of the complete domain of length 0.04m can be seen in Figure 6.3. In Figure 6.2(a), a change of drying state occurs at cell number 60. The local error is high only in a neighbourhood of cell number 60. This can be seen in Figure 6.2(a). The time step is rejected due to this local maximum error at cell number 60 and redo the computation with a smaller time size at all spatial grid points using an implicit scheme. We partition the time discretization in such a way that we consider an implicit scheme in the neighbourhood of a region where the drying state changes and an explict scheme in the rest of the domain using the reduced time step sizes. For instance in Figure 6.2(b), we use an implicit scheme only in a neighbourhood of the cell number 60. Figure 6.3 presents the comparison of saturation profiles with and without partitioning at time 3.05 days. We take tol=10^{-5} during the first and intermediate state of drying and tol=10^{-6} during the second drying state.

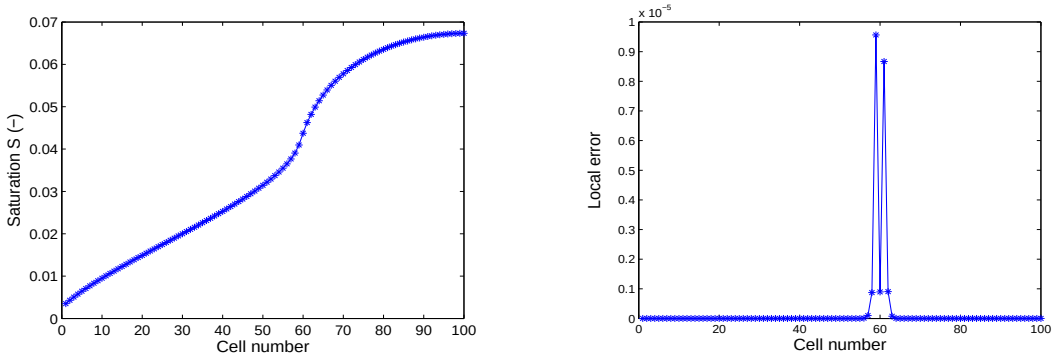


Figure 6.2: Saturation profile at 24.39 hours and the corresponding local temporal error.

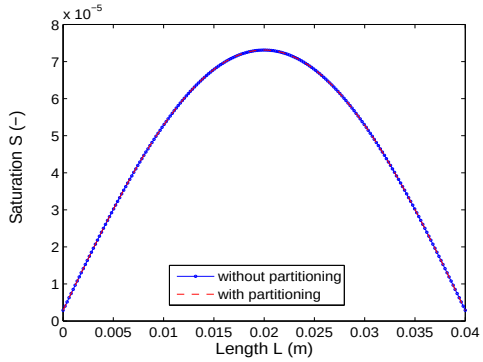


Figure 6.3: Comparison of solution with and without partitioning.

Efficiency

The following table gives the CPU time in minutes using a W-method with partitioning and without partitioning. These CPU times are calculated at the end of a real drying time of 3.05 days.

Cells	CPU time (min) without partitioning	CPU time (min) with partitioning
100	20.85	15.98
150	77.52	47.18
200	203.35	91.15

Table 6.1: Partitioning efficiency.

Loss of positivity

It is important to notice that the solution loses positivity with larger time step sizes during the second state of drying process. For this purpose, the tolerance *tol* should be reduced during the second drying state in order to reduce the time step sizes. For instance by using tol=10^{-5}, the solution loses positivity and then diverges at a real time of 4.29 days as shown in the following figure. This is the reason to take tol=10^{-6} during the second drying state.

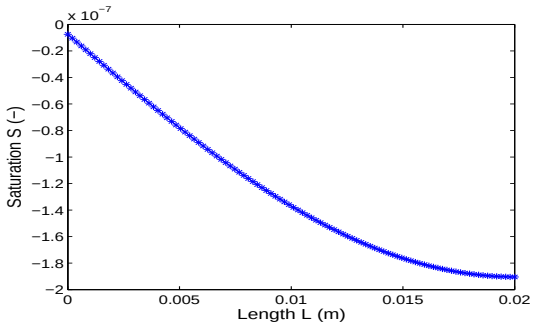


Figure 6.4: Loss of positivity.

Coupled problem

We have noticed that the partitioning is not efficient for the coupled problem. This is because there are no local regions where the scheme is stable with explicit schemes. The problem is highly stiff at first, second and intermediate stages of the drying process. Figure 6.5 presents the instability of the solution with an explicit W-method. The accepted time step sizes are very small using a tolerance of tol=10^{-3}. A further decrease of tolerance leads to very small time step sizes. Hence an explicit scheme is not efficient in the case of coupled problem. This motivates us towards a local time stepping strategy for this case.

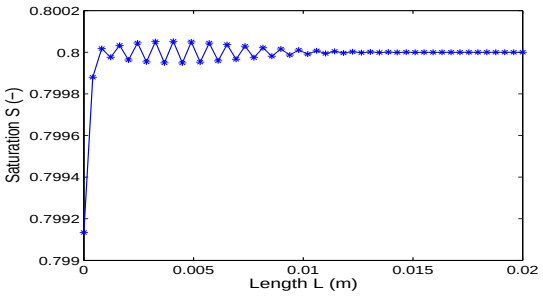


Figure 6.5: Instability with an explicit scheme.

6.4 Local time stepping

Local time stepping schemes are restricted by a local stability condition instead of the general global stability condition. The idea of local time stepping is to exploit the largely differing time scales during the numerical integration. In solving the complex systems of partial as well as ordinary differential equations, during many cases the problem becomes highly stiff only in a certain local region. An integration scheme for solving ordinary differential equation is limited by the fastest changing component, whereas, the local time stepping strategy employ an inherent time step size for each subsystem. In literature these methods are also known as multirate methods. The main goal of the local time stepping is to use larger time step sizes in the nonstiff and less stiff regions, and smaller time step sizes in the highly stiff regions. In our computational work, we use a local time stepping strategy suggested by Savcenco et al. [39] where the time step size at a particular grid point is determined by the local temporal variation of the solution. We apply this strategy for solving the quasilinear time dependent coupled parabolic problem given in Section 6.2. More information about these local time stepping procedures can be found in [3, 11, 12, 17, 39, 43].

6.4.1 A single time slab

Let R_0=Ω be the complete spatial domain and [0,T_{end}] be a given time interval. Divide [0,T_{end}] into finite set of subintervals $0 = t_0 < t_1 < t_2 .... < t_n = T_{end}$. The length of a single time slab is given by τ_i=$t_{i+1} - t_i$. Figure 6.6 represents a typical single time slab of length τ_n. Here we explain the processing of this time slab.

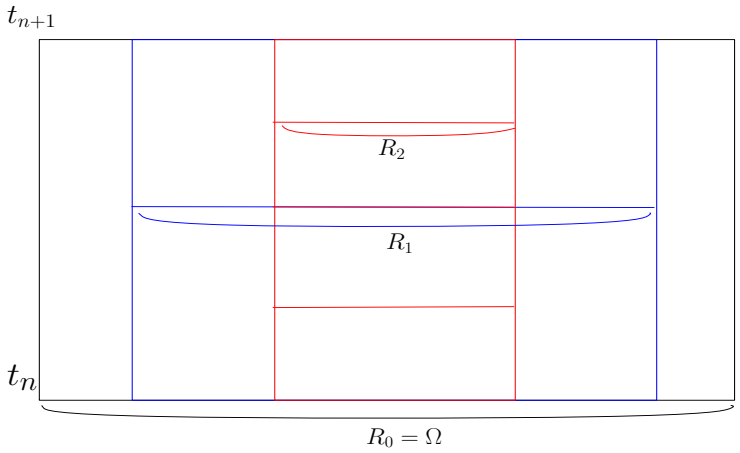


Figure 6.6: Typical time slab.

The solution of a given problem is known at the current time level t_n and we wish to find the solution at the next time level t_{n+1}. At first, we calculate the solution at the time

level t_{n+1} with a time step size of τ_n=$t_{n+1} - t_n$. Then, we determine the local temporal error at the time level t_{n+1} using the equation (6.5). In the usual process, we reject the time step if the local temporal error is greater than the prescribed tolerance and redo the computations with a reduced time step size given by equation (6.6).

By using the local time stepping strategy we do not reject the time step, instead we proceed in the following way. Identify the local region where the temporal error is greater than the prescribed tolerance. For instance in Figure 6.6, this region is $R_1 \subset R_0 = \Omega$. Then, we go back to the time level t_n and recalculate the solution only in the region R_1 with a time step size of $\frac{\tau_n}{2}$ instead of τ_n. We call the processing of this region R_1 as the first level of refinement. Thus in R_1, we reach the time level t_{n+1} in two time steps. Now we find the region in R_1 where the local error is greater than the given tolerance. This region is represented by $R_2 \subset R_1$ in Figure 6.6. Again we go back to the previous time level t_n and we recalculate the solution only in this small region R_2 with a time step size of $\frac{\tau_n}{4}$ i.e., in R_2 we reach the time level t_{n+1} in four time steps. We continue this process until there is no error at all spatial grid points at the time level t_{n+1}. Thus for m levels of refinement, we have m local regions $R_m \subset R_{m-1} \subset ..... \subset R_0 = \Omega$ and the time step size in each R_i is given by $\frac{\tau_n}{2^i}$. By this process, the time steps are rejected only at some local grid points instead of rejecting the time step at all the spatial grid points.

6.4.2 Interface boundary conditions

In the processing of a time slab using local time steps, it is required to deal with artificial boundary nodes. These nodes are shown by red circles in the Figure 6.7. As the solution at time levels t_n and t_{n+1} is known, a linear or quadratic interpolation is used to obtain the values at the internal interface boundary nodes during every successive level of refinement.

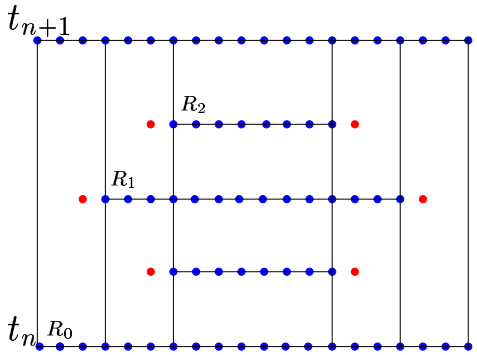


Figure 6.7: Interface boundaries.

6.4.3 Length of a time slab

The important factor is chosing an appropriate length of a time slab. A proper estimate for the length of the time slab is needed to make the local time stepping strategy more efficient. For this purpose, we make use of the information available from the last completed time slab to estimate the length of the next time slab. Let the length of the last completed time slab be τ_n=$t_{n+1} - t_n$. We store the information at the regions of last accepted local time steps in all the levels of refinement during the last completed time slab. This region is represented in the shaded part of the Figure 6.8.

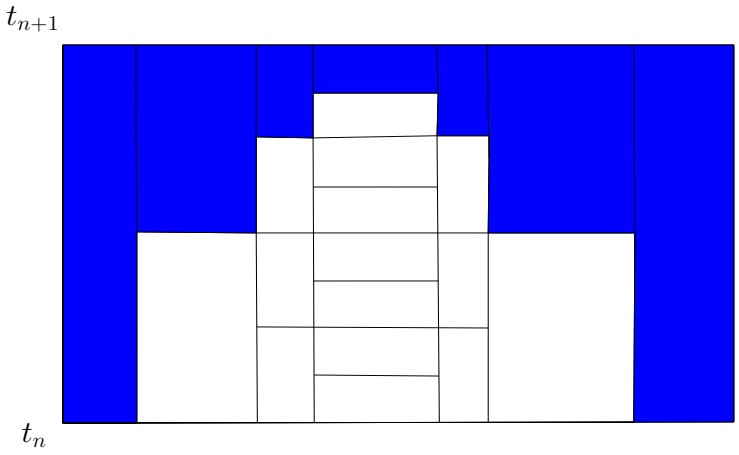


Figure 6.8: Last accepted local time steps.

Suppose the n^{th} slab is completed with m levels of refinements i.e., we have m regions $R_m \subset R_{m-1} \subset ... \subset R_0 = \Omega$. The length of the time step size in each R_i is given by $\frac{\tau_n}{2^i}$, for i=0,1...,m. We store the local error which is estimated by equation (6.5) at last accepted local time steps in each R_i which is the shaded part in Figure 6.8 and denote this error by err_i. Then, we estimate a time step for all points in $R_i \backslash R_{i+1}$ based on the local time steps $\frac{\tau_n}{2^{i+1}}$ and error err_i. This estimate is given by

$$\tau_{n+1}^i = \zeta \frac{\tau_n}{2^{i+1}} \left(\frac{tol}{err_i} \right)^{1/p}. \tag{6.14}$$

The equation (6.14) gives an estimate of the time step size for all the grid points in $R_i \backslash R_{i+1}$. We estimate these time step sizes for each local region. The minimum of all these time step sizes is denoted by $\tau_{n+1}^{\min}$. This optimal time step size τ_{n+1}^{min} serves as the time step size to keep the local error less than the tolerance at all grid points in order to proceed in time from t_n to $t_n + \tau_{n+1}^{\min}$ in a single rate. This minimal time step is given by

$$\tau_{n+1}^{\min} = \min(\tau_{n+1}^0, \tau_{n+1}^1, ...). \tag{6.15}$$

Let L_n denote the number of levels of refinement for the last completed time slab and L_{n+1} denote the estimated levels of refinement for the next time slab. It can be understood that the length of the new time slab is then given by

$$\tau_{n+1} = 2^{L_{n+1}} \tau_{n+1}^{\min}. \tag{6.16}$$

The next task is to find an estimate for L_{n+1}. A large choice of L_{n+1} may cause instabilities and a small choice of L_{n+1} leads to poor efficiency. The estimation of L_{n+1} is based on the amount of work required to cover a unit of time. Let the n^{th} time slab of length τ_n=$t_{n+1} - t_n$ is finished with m levels of refinement i.e, we have m local regions $\Omega = R_0 \subset R_1 \subset R_2 .... \subset R_m$. The number of nodes in each R_i be N_i, for i=0,1,2,...m with total number of nodes $N_0 = N$ and the time step size in each R_i is $\frac{\tau}{2^i}$.

The amount of work done for one time step at N spatial grid points is proportional to N^r, $r \geq 1$. Thus the amount of work done in processing the entire time slab per unit time is approximated by $W = \frac{1}{\tau_n}(N_0^r + 2N_1^r + ... + 2^m N_m^r)$. Then we go back to the time level t_n and estimate the amount of work done if we have started with a smaller time slab of length $\frac{1}{2^{m_0}}\tau_n$, instead of τ_n. In this case, the levels of refinement are estimated to be $m - m_0$. Denote the amount of work done using this smaller time slab by W_1. Naturally if W_1 is less than W, the choice of this smaller time slab of length $\frac{1}{2^{m_0}}\tau_n$ will be efficient with less levels of refinement than the original slab of length τ_n. The value of such W_1 attains its minimum for

$$L_* = \max\{i : \text{the number of nodes in } R_i > \rho N\}. \tag{6.17}$$

The value of ρ is $(\frac{1}{2})^{1/r}$, $r \geq 1$. For complete details about calculating the amount of work done W, W_1 and the derivation of the estimate (6.17), see Savcenco et al. [39]. Now there exist two cases

1. if L_*>0, a smaller time slab might be efficient. Then the estimated levels of refinement for the new time slab should be reduced by L_*, i.e. $L_{n+1} = L_n - L_*$.

2. if L_*=0, a larger time slab might be efficient.

If we are in case 2, we can double the length of the next time slab by increasing the estimated levels of refinement by 1, i.e., L_{n+1}=$L_n + 1$. For this purpose, we go back to the time level t_n and approximate the amount of work done per unit time if we have started with a larger time slab of length $2\tau_n$ instead of τ_n. Denote the work done using $2\tau_n$ by W_2. If $W_2 < W$, then it implies that the larger time slab with length $2\tau_n$ would be more efficient than the original time slab of length τ_n. Therefore we need an estimate to decide whether $W_2 > W$ or $W_2 < W$. It is estimated in the following way.

Again we use the information from the last computed time slab with length τ_n=$t_{n+1} - t_n$. Let $E = \mathbf{u}^{n+1} - \hat{\mathbf{u}}^{n+1}$ be the difference in the two solutions calculated using Rosenbrock methods given by (6.9), (6.10) and err be the maximum norm of E. The vector E is of length N, where N is the total number of grid points. The value of E is approximately

equal to τ_n^p, where p is the order of the method. Hence we would expect an error of $2^p E_n$ if we would have started with a time step size of $2\tau_n$. Therefore, at time level t_{n+1}, we would also find the local temporal error with a tolerance of $\frac{tol}{2^p}$ instead of tol. Since we use a 2 stage Rosenbrock method, here we take p=2. Consider the following set

$$E_d = \{i : |E_i| > \frac{tol}{4}, i = 1, 2, ...N\}, \tag{6.18}$$

where E_i gives the error at each node i and E_d gives the number of nodes at which the local error is greater than $\frac{tol}{4}$. Consider the following inequality

$$E_d < \rho N. \tag{6.19}$$

If (6.19) is satisfied, then it implies that $W_2<W$, which follows from (6.17). And therefore, we can conclude that it would be more efficient if the length of the time slab is doubled.

After doing all these approximations, the estimated levels of refinement for the next time slab L_{n+1} can be given by

$$L_{n+1} = \begin{cases} L_n + 1 & : \text{ if (6.19) is satisfied} \\ L_n - L_* & : \text{ otherwise.} \end{cases} \tag{6.20}$$

The value of L_* is given by (6.17). Therefore the length of the next time slab can be calculated by equation (6.16). All this process should be carried out for time slab to time slab until the final time is reached. At each region, great care should be taken to evaluate the interface boundaries and at each local region the Jacobian matrix is constructed using finite differences.

6.4.4 Numerical observations

We present the numerical observations for the coupled problem with local time stepping using ROS2 for time integration. The length of the concrete sample is 0.04m and for numerical simulations we take a length 0.02m by insulating a boundary end. Figure 6.9 represents the accepted time step sizes using ROS2 without local time stepping. From this figure, it can be noticed that there are time step reductions in the intermediate state of the drying process. Figures 6.10(a),(b) and (c) represent the saturation profiles during the intermediate state of drying at 28.51, 37.46 and 46.60 hours respectively. From these figures we notice that time step reductions occur, due to a high local temporal error only in a small region where the change of drying state occurs. For instance at 28.51 hours in Figure 6.10(a), a change of drying state occur at cell number 30 and the corresponding error is high only in a region of small neighbourhood of cell 30. Similar trends can be observed in figures 6.10(b) and (c) at 37.46 hours and 46.60 hours near cell numbers 50 and 70 respectively. In the usual process, the time step size is reduced at all spatial grid points due to this local high errors. But with local time stepping smaller time steps are chosen only in the regions of high errors. Figure 6.11 present the comparison of average saturation with and without local time stepping. Tables 6.2 and 6.3 represent the efficiency with local time stepping method calculated at the end of the real drying time of 5.2 days.

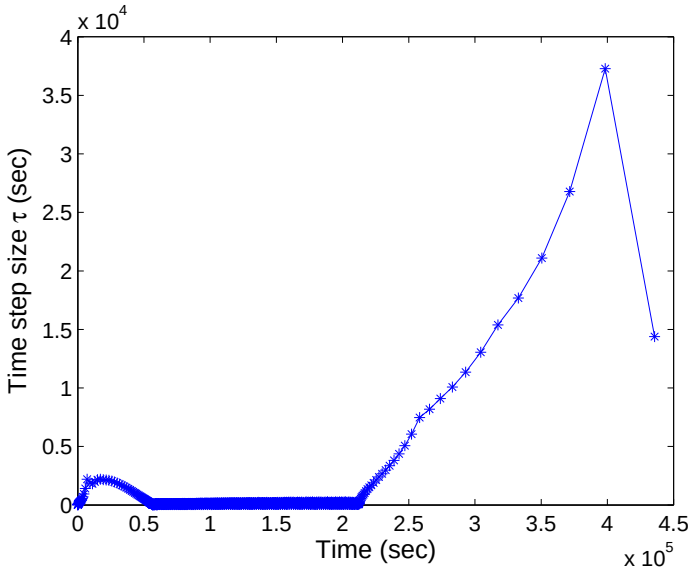
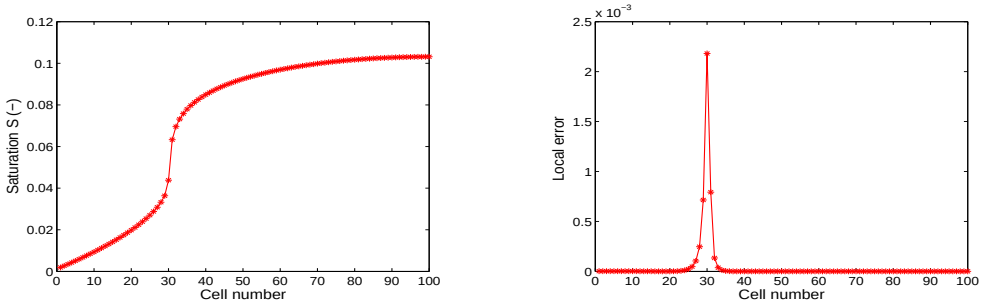
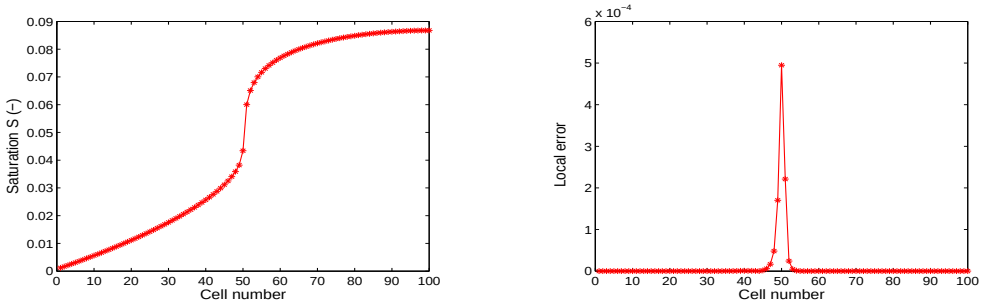


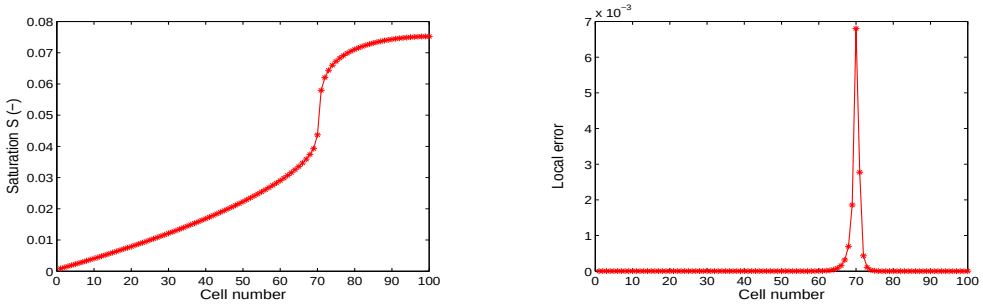
Figure 6.9: Accepted time step sizes with ROS2, tol=10^{-4}.



(a) at 25.51 hours



(c) at 37.46 hours



(d) at 46.60 hours

Figure 6.10: Saturation profiles and the corresponding local errors.

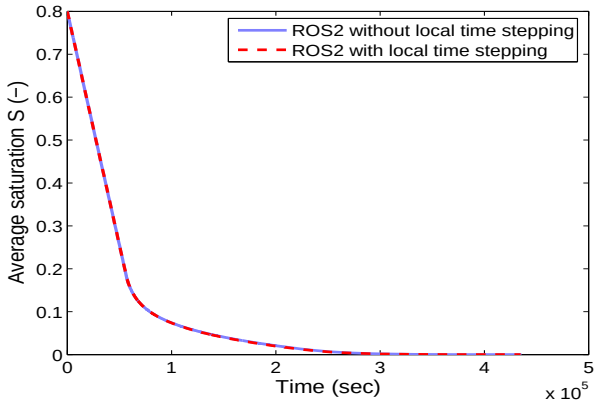


Figure 6.11: Average saturation profile.

Cells	CPU time (min) without local time steps	CPU time (min) with local time steps
100	19.62	16.51
150	69.96	65.78
200	175.78	163.38

Table 6.2: Efficiency using local time stepping, CPU time in minutes.

Another case

The following table illustrates the CPU times for the coupled problem by taking P_c, k_l and k_g given by equations (6.11), (6.12) and (6.13).

Cells	CPU time (min) without local time steps	CPU time (min) with local time steps
50	12.38	5.68
100	166.91	32.94

Table 6.3: Efficiency using local time stepping.

From these tables, we can observe the gain in the computational cost using local time steps. In certain cases, these local time stepping methods are stronger than partitioning methods because partitioning is more efficient only if there exist a local region where an explicit scheme is stable. On the other hand local time stepping methods can be applied to highly stiff problems using implicit local time steps determined according to the local variation in the solution though these schemes are slightly nonconservative at the grid interfaces.

Chapter 7

Summary

The main aim of the work was to investigate efficient numerical methods for solving the equations governing the isothermal drying of porous media. Drying is a complex process of heat and mass transfer which involves the removal of water in pores by evaporation. We considered a continuous drying model of Perré [36] at isothermal conditions i.e., drying at constant temperature. An isotropic porous medium (concrete) was considered. The governing equations are a quasilinear coupled system of stiff parabolic partial differential equations.

The difficult factors which influence numerical simulations in this field are nonlinearity, strong coupling, the necessity of large real time computations and many dependent variables defined by just as many equations of state. Moreover the definitions of all the variables change according to the drying state. They are solution dependent. This change of drying states leads to a sudden rapid increase or decrease in variables near a grid point at which a change in drying state takes place. In a neighbourhood of that grid point, we observed that the problem becomes highly stiff. Due to the complexity of the problem, we therefore reduced the model into two problems, a scalar and a coupled quasilinear parabolic problem. The scalar problem consists of a primary variable saturation S and 18 dependent variables whereas the coupled problem consists of two primary variables, saturation S and air density ρ_a with 19 secondary variables.

At first we considered the one dimensional case for the numerical simulation. A cell centered finite volume scheme has been implemented for the scalar problem with explicit, semi-implicit and implicit time discretizations and the corresponding efficiencies were compared. The Jacobian matrix was constructed numerically using finite differences. The linear system of equations with a tridiagonal matrix resulting from the scalar problem was solved by the Thomas algorithm. An algorithm for solving the scalar quasilinear parabolic problem has been presented. The convergence of the solution is attained with refinements of the mesh. For the coupled quasilinear parabolic problem, we found that explicit schemes are not stable with sufficiently large time steps. Hence an algorithm has been developed to solve the complex coupled system with fully implicit time discretization. Through our

numerical simulation, the physical trends of the drying process were depicted properly. The profiles of average moisture, air density and the local behaviour of saturation, vapour, air and gas pressures were presented. Here we used fixed time step sizes, a large time step size was considered in the first drying state and a small time step size after that. We observed a drastic reduction in time step size after the first drying state even using an implicit time discretization. With large time step sizes, the Jacobian matrix becomes ill-conditioned and the Newton method diverges. Further we noticed negative solutions using large time steps during the final stages of the drying process. As a next step, we considered the method of lines approach where the space and time discretizations are done seperately. A fifth order implicit time integration method, the so called Radau5 method which uses simplified Newton iterations with variable time step size control has been used. We found that the variable time stepping procedure using the Radau5 method is more efficient than the fixed time step size using an implicit Euler method. We also observed that these time step reductions occur during the intermediate state of the drying process at which the definitions of the variables change according to the drying state.

Also the time step size is to be restricted in order to attain positivity preservation. Positive solutions are important in a wide range of practical problems. Especially, in the current problem of drying many variables such as saturation and densities cannot become negative. Hence a positivity analysis has been done for a scalar parabolic problem in one space dimension. We first consider a quasilinear parabolic initial value problem. Then by using the properties of M-matrices, the positivity of the solution was proved using a semi-implicit finite difference scheme. Next, the analysis has been extended to a quasilinear parabolic boundary value problem and finally to a particular case of a quasilinear parabolic problem. We show that the reduced scalar drying problem is an application for the analysis of the extended quasilinear parabolic case. However, in the present work the analysis could not be extended to the coupled case.

For the two dimensional case, we considered the dimensional splitting approach, because the time step restrictions become severe when moving to higher dimensions. Especially we observed that these time step reductions are severe during the intermediate state of drying. Also the efficiency decreases by solving a large system of nonlinear algebraic equations with reduced time step sizes, when moving to higher dimensions. Using dimensional splitting, we broke the two dimensional problem into one dimensional parts. Two kinds of symmetric splitting techniques of Strang were implemented and the efficiency was compared. A two dimensional spatial domain (square) was reduced to one fourth of the original domain by making use of the symmetry with two sides being insulted. We considered the Newton method to linearize the nonlinear system of equations. A BiCGSTAB with ILU preconditioner was used to solve the resulting linear system of equations. The dimensional splitting approach is efficient and the symmetry in the solution is maintained perfectly. The physical trends of the drying process are attained properly. The saturation and the air pressure profiles for concrete were presented at different drying times.

Final aim of the study was to investigate efficient time stepping strategies to increase the efficieny in the computational cost in one space dimension. For this we chose the MOL approach. We considered linearly implicit methods for integrating the semi-discrete system in time. The linearly implicit methods are efficient in solving many stiff ODEs and DAEs because these methods avoid nonlinear Newton iterations. We used variable time step sizes based on a time step selection criterion. In the current work, a three stage W-method and a two stage Rosenbrock method ROS2 were implemented for time integration. We first considered a partitioning strategy through which an explicit scheme is used in nonstiff regions and an implicit scheme in stiff regions. These regions are recognised by a local temporal error estimator, calculated by the difference between two solutions of two different orders. The partitioning strategy was first implemented to a quasilinear scalar parabolic problem. The local temporal error is high only in a small neighbourhood of the region where the change in drying state takes place. This causes the reduction of the time step size for the entire domain using an implicit scheme. We partitioned in such a way that an implicit scheme is used in a small neighbourhood of a region where a change in drying state occurs and an explicit scheme in the rest of the domain. This approach results in solving small linear systems and therefore we attained good efficiency in the computational cost for a scalar problem. Also positivity is maintained by decreasing the tolerance during the second drying state which in turn decreases the time step size. However for the coupled drying problem, partitioning is not found to be efficient because the problem is highly stiff in all parts of the domain and at all stages of the drying process. For the coupled parabolic case, a local time stepping method has been implemented. We have considered different time steps in different parts of the domain based on local temporal variation. The total time interval was divided adaptively into finite subintervals which are called time slabs. In each time slab, we used local time steps based on a local error criterion. We considered smaller time steps in highly stiff regions and larger time steps in mildly stiff and nonstiff regions. A linear interpolation has been used to approximate the internal interface boundaries. The local time stepping strategy is efficient in the computational cost for the coupled parabolic drying problem.

Bibliography

[1] U.M. Asher, S. J. Ruuth, and B.T.R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM J. Numer. Anal.*, 32:797–823, 1995.

[2] D. Barash and R. Kimmel. An accurate operator splitting scheme for nonlinear diffusion filtering. *Lecture notes in computer science, Springer Berlin*, 2160:281–300, 2001.

[3] A. Bartel and M. Gunther. A multirate W-method for electrical networks in state-space formulation. *J. of Computational and App. Mathematics*, 147:411–425, 2002.

[4] T.A. Bickart. An efficient solution process for implicit Runge-Kutta methods. *SIAM J. Numer. Anal.*, 14:1022–1027, 1977.

[5] R.B. Bird, W.E. Steward, and E.N. Lightfoot. *Transport phenomena.* JohnWiley and Sons, New York, 1960.

[6] E.K. Blum. *Numerical analysis and computation, theory and practice.* Addison-Wesley Publishing Company, 1972.

[7] N. Boukadida and S.B. Nasrallah. Two dimensional heat and mass transfer during convective drying of porous media. *Drying technology*, 13(3):661–694, 1995.

[8] J. Bruder, K. Strehmel, and R. Weiner. Partitioned adaptive Runge-Kutta methods for the solution of nonstiff and stiff systems. *Num. Math*, 52/6:621–638, 1998.

[9] J.C. Butcher. On the implementation of implicit Runge-Kutta methods. *BIT*, 17:237–240, 1976.

[10] M. Büttner, B. A. Schmitt, and R. Weiner. W-methods with automatic partitioning by krylov techniques for large stiff systems. *SIAM J. Numer. Anal.*, 32:260–284, 1995.

[11] C. Lubich C. Engstler. Multirate extrapolation methods for differential equations with different time scales. *Computing*, 58:173–185, 1997.

[12] C. Dawson and R. Kirby. High resolution schemes for conservation laws with locally varying time steps. *SIAM J. Sci. Comput*, 22:2526–2281, 2001.

[13] H.A. Van der Vorst. A fast and smoothly converging variant of bi-cg for the solution of non-symmetric linear systems. *SIAM J. Sci. Stat. Computat.*, 13:631–645, 1992.

[14] P. Deuflhard. *Newton methods for nonlinear problems-Affine Invariance and Adaptive algorithms.* Springer, New York, 2004.

[15] J. Frank, W. Hundsdorfer, and J.G. Verwer. On the stability of implicit-explicit linear multistep methods. *Appl. Numer. Mathematics*, 25:193–205, 1997.

[16] C. Fyhr and A. Rasmuson. Numerical simulation of superheated steam drying of wood chips. *Proceedings of the 9th international drying symposium, Australia*, 94:727–734, 1994.

[17] C. Gear and D. Wells. Multirate linear multistep methods. *BIT*, 24:1996, 1984.

[18] Hairer and Wanner. *Solving ordinary differential equations II, stiff and differential-algebraic equations.* Springer-Verlag, Berlin, 1996.

[19] W. Heineken and G. Warnecke. Partitioning methods for reaction-diffusion problems. *Applied numerical mathematics*, 56:981–1000, 2006.

[20] W. Hundsdorfer and J.G. Verwer. *Numerical solution of Time-Dependent Advection-Diffusion-Reaction Equations.* Springer, Berlin, 2003.

[21] C.T. Kelley. *Iterative methods for linear and nonlinear equations.* SIAM, Philadelphia, 1995.

[22] A. Kurganov and D. Levy. A third-order semidiscrete central scheme for conservation laws and convection-diffusion equations. *SIAM J. Sci. Comput.*, 22:1461–1488, 2000.

[23] J. Lang. Adaptive multilevel solutions of nonlinear parabolic pde systems. theory, algorithm, and applications. *Lecture notes in computational science and engineering, springer*, 16, 2000.

[24] R. J. LeVeque. *Finite volume methods for hyperbolic problems.* Cambridge University Press, 2002.

[25] G. I. Marchuk. On the theory of splitting-up method. *SYNSPADE, Ed. B. Hubbard, Procs. of the second symposium on the numerical solution of partial differential equations, Academic press*, 1970.

[26] K.W. Morton and D.F. Mayers. *Numerical solution of partial differential equations.* Cambridge University Press, England, 1994.

[27] C. Moyne and P. Perré. Processes related to drying: Part I, theoretical model. *Drying technology*, 9(5):1135–1152, 1991.

[28] L. Noels, L. Stainier, and J.P. Ponthot. Combined implicit/explicit time-integration algorithms for the numerical simulation of sheet metal forming. *J. Comput. Appl. Math.*, 168:331–339, 2004.

[29] J. M. Ortega. *Numerical analysis, a second course.* SIAM, philadelphia, 1990.

[30] S. V. Patankar. *Numerical heat transfer and fluid flow.* Longman scientific and technical, 1995.

[31] D.W. Peaceman and H.H. Rachford. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math.*, 3:28–41, 1955.

[32] P. Perré, M. Moser, and M. Martin. Advances in transport phenomena during convective drying with superheated steam and moist air. *Int. J. of heat and mass transfer*, 36:2725–2746, 1993.

[33] P. Perré and I. Turner. A synopsis of the strategies and efficient resolution techniques used for modelling and numerically simulating the drying process. In *Mathematical Modelling and Numerical Techniques in Drying Technology*, Marcel Dekker, pages 1–82, New York, 1996.

[34] P. Perré and I. Turner. The use of macroscopic equations to simulate heat and mass transfer in porous media. In *Mathematical Modelling and Numerical Techniques in Drying Technology*, Marcel Dekker, pages 83–156, New York, 1996.

[35] P. Perré and I. Turner. Microwave drying of softwood in an oversized waveguide. *AIChE Journal*, 43(10):2579–2595, 1997.

[36] P. Perré and I. Turner. A 3-d version of transpore: a comprehensive heat and mass transfer computational model for simulating the drying of porous media. *Int. J. of heat and mass transfer*, 42:4501–4521, 1999.

[37] P. Perré and I. Turner. A generic heat and mass transfer computational model for understanding and visualising the drying of porous media. *Drying technology*, 17:1273–1289, 1999.

[38] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes in C++.* Cambridge university press, England, 2002.

[39] V. Savcenco, W.H. Hundsdorfer, and J.G. Verwer. A multirate time stepping strategy for parabolic pde. *Report MAS-E0516*, 2005.

[40] L. F. Shampine. *Numerical solution of ordinary differential equations.* Chapman and Hall Mathematics, New York, 1994.

[41] G. Strang. Accurate partial difference methods I: linear cauchy problems. *Arch. Rat. Mech. Anal*, 12:392–402, 1963.

[42] G. Strang. On the construction and comparison of difference schemes. *SIAM J. of Sc. Comp*, 5:506–517, 1968.

[43] H. Tang and G. Warnecke. A class of high resolution schemes for hyperbolic conservation laws and convection-diffusion equations with varying time and space grids. *SIAM J. of Sc. Comp.*, 4:1415–1431, 2005.

[44] I. Turner. *The modelling of combined microwave and convective drying of a wet porous material.* PhD thesis, University of Queensland, 1991.

[45] I. Turner and P. Perré. The use of implicit flux limiting schemes in the simulation of the drying process. *Applied mathematical modelling*, 25:513–540, 2001.

[46] H.K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: The finite volume method.* Longman Scientific and Technical, England, 1995.

[47] J. Weickert, B.M.T.H. Romeny, and M.A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on image processing*, 13:303–319, 1993.

[48] R. Weiner, M. Arnold, P. Rentrop, and K. Strehmel. Partitioning strategies in Runge-Kutta type methods. *IMA J. Num. Anal.*, 13:303–319, 1993.

[49] S. Whitaker. Heat and mass transfer in granular porous media. In A.S. (Eds.) Hemisphere Publ. Corp. I, Mujumdar, editor, *Advances in drying*, Marcel Dekker, pages 23–61, Washington DC, 1980.

[50] S. Whitaker. Simultaneous heat, mass and momentem transfer in porous media: A theory of drying. *Advances in heat transfer*, 31:1–104, 1998.

[51] S. Whitaker. *The method of volume averaging.* Kluwer academic publishers, Netherlands, 1999.

[52] G. Windisch. *M - matrices in Numerical Analysic.* Teubner - Texte Zur Mathematik, Leipzig, 1989.