A robust implicit scheme for two-phase flow in porous media

Anna Kvashchuk

Thesis for the degree of Master of Science
in Applied and Computational Mathematics

Department of Mathematics
University of Bergen
Norway

November 2015

Acknowledgments

First of all I would like to express my deep gratitude to my supervisor Adrian Florin Radu for his help, inspiration and positivity about the project. It was a great pleasure to work under his wise guidance.

I also would like to thank my husband Sergey Alyaev who challenged me a lot regarding this work. This thesis would not be so tough and at the same time, satisfying without him. I am grateful to him for his support, patience and numerous hints and advices.

I would like to thank the Department of Mathematics and NUPUS group for the valuable, fascinating and just lovely trip to the USA where we visited great universities and meet awesome people. I also want to thank Alexander Vasiliev for establishing the agreement with Saratov State University that gave me an opportunity to study at the University of Bergen.

I am thankful to all my dear friends who helped me going through it. I thank Georgy Ivanov and Anastasia Lisenkova for their help with improving my English language grammar, Anna Varzina and Alexey Tochin for useful conversations during the breaks and moral support. I am also deeply grateful to my family whose love and support I felt despite the long distance.

Anna Kvashchuk
November, 2015

Abstract

In this thesis we present a new implicit scheme for the numerical simulation of two-phase flow in porous media. Linear finite elements are considered for the spatial discretization. The scheme is based on the iterative IMPES approach and treats the capillary pressure term implicitly to ensure stability. Under assumption of smoothness of the capillary pressure and the phase mobility curves, we were able to prove convergence theorem for the scheme. Two dimensional numerical simulations furthermore verify the convergence.

To illustrate the potential of the new scheme we compare its computational efficiency to our implementation of two other common approaches to the problem: IMPES and the fully implicit formulation solved by Newton's method. The advantage of our scheme over IMPES is improved stability for larger time-step. At the same time, it is cheaper in terms of computational costs and memory requirements compared to the Newton method.

Contents

Chapter 1

Introduction

Multiphase flow in porous media is a problem that appears in many fields of knowledge and has numerous applications. Whether you study enhanced oil recovery, CO_2 storage, groundwater flows or nuclear waste management you have to deal with multiphase systems. That is why industries are interested in efficient methods for modeling multiphase flow. The numerical methods come into focus as in most cases the analytical solution cannot be obtained without making significant simplifications that may lead to an unphysical behavior of the model.

From the mathematical point of view, multiphase flow in porous media can be represented as a system of coupled nonlinear partial differential equations (PDEs). The nonlinearity of the system makes development and implementation of robust numerical schemes a challenging task. Moreover, this kind of problems usually are formulated for a big and complex structured domain which also brings difficulties.

One way to deal with the complex domain geometry is to use the Finite Elements Method (FEM) [5] for space discretization. FEM is a commonly used powerful technique for solving systems of PDEs in complex domains as it naturally can be extended to flexible discretizations. Another advantage of FEM is a relatively easy handling of boundary conditions and a solid theoretical base that gives it high reliability. Therefore, in this thesis we are using FEM as a basis for the considered numerical schemes.

The numerical approaches that help to handle nonlinearities in a multiphase flow system have been developed and improved over the last decades [3, 7]. In this thesis we study two of the most frequently used classes of numerical schemes for time integration of the two-phase flow model. The most popular one is called the IMplicit Pressure Explicit Saturation method, or IMPES [2, 16, 9]. The main feature of this method is elimination of nonlinearities by taking advantage of the form of the equations. However, the explicit treatment of the saturation equation results in the restrictions on the time-step [10, 11] which makes the scheme relatively slow. The alternative to IMPES is the fully implicit scheme [13, 22, 14, 23] which does not have any restrictions on the time step. The system arising from applying the fully implicit scheme is nonlinear and one needs an efficient algorithm for solving it. A common approach is Newton's method [18, 21] which has quadratic convergence. This comes at a price of a costly computation of derivatives at each iteration. Additionally, the mentioned convergence properties demand that the initial guess is sufficiently close to the true solution which, in its turn, may imply additional requirements on the step size.

In this thesis we present a new implicit scheme specially tailored for the two-phase flow model and compare it with the two approaches mentioned above. We base it on the iterative IMPES approach which is a straightforward way to make a solver for the implicit scheme. However, our early numerical studies showed that using the naive iterative IMPES without stabilization as a solver does not bring any improvements over using IMPES as a semi-explicit scheme. The difficulties arise from nonlinear coefficients, especially the capillary pressure term. In order to approximate the gradient of the capillary pressure function which appears in the saturation equation we use a linear expression involving the saturation at the current and the previous iterations. This made the scheme stable and more efficient compared with the standard IMPES. In [20] the authors independently developed a similar approximation for the capillary pressure function for the two-phase flow model written for the pressure potentials and also rigorously proved the convergence. In contrast with our approach, they use the cell-centered finite difference method for the space discretization. Another approach is presented in [28], where the authors developed a new linearization scheme for the nonlinear system arising after the finite volume discretization of the two-phase flow model.

The new scheme developed in this thesis preserves efficiency in treatment of nonlinearities and implementation simplicity of IMPES while relaxing the time step condition common for explicit methods. At the same time, it does not involve computation of derivatives, which brings it advantages over Newton's method. What is more, the linear systems to be solved at each iteration step for the new scheme are much better conditioned compared with the one resulting for Newton's method. The rigorous convergence proof for the proposed scheme is also presented.

The rest of the thesis is organized as follows. In Chapter 2 we provide an overview of the equations that govern the two-phase flow and derive the mathematical model for the averaged pressure formulation. Chapter 3 is devoted to an overview of numerical methods for the introduced model. We give a short overview of the Finite Elements Method, derive the weak form of the problem and discuss different types of time discretization, both explicit and implicit. The new implicit scheme is presented in Section 3.3.2 and its convergence is rigorously proven in Section 3.3.3. In Chapter 4 we illustrate the convergence of the new method with several numerical simulations. The comparison of the new scheme with IMPES and Newton's iteration for implicit scheme is demonstrated in Section 4.2. The conclusion and final remarks are made in Chapter 5.

Chapter 2

Mathematical Model

In order to derive a good mathematical model one should study in depth the physics behind the modelled process. Porous media has been a subject of research for a quite a long time now and main constitutive relations were developed in many works [3, 26, 7]. In this chapter we present derivations inspired by the book [26] and follow their notation.

The aim of this chapter is to construct a mathematical model of two-phase flow. In this thesis we work with the averaged pressure formulation of the two-phase flow model and develop various numerical methods for it. This valuable formulation is widely used in practice.

We start with the fundamental equations that govern single-phase flow in porous media and basic physical properties of the flow and porous media itself. Then we introduce a general mathematical model for two-phase flow, from which after some algebraic manipulations and a change of the primary variables we obtain the averaged pressure formulation.

2.1 Mass Conservation

The principle of mass conservation is a basic concept in many fields, including fluid dynamics. This statement is crucial for deriving a mathematical model that describes fluid flow in porous media. The idea is to consider an arbitrary domain Ω with a boundary Γ (see Fig. 2.1) and observe how mass changes inside this volume over time. The change of mass in a particular region is equal to the amount of mass that goes in or out of Ω trough the boundary Γ, plus any mass added or subtracted with possible sink or source term. However, in order to write this principal in mathematical equation we first need to introduce various physical quantities that would be used in our model.

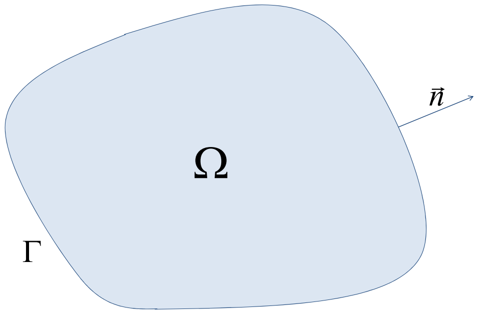

Figure 2.1: Example of an arbitrary domain Ω.

2.1.1 Physical Properties of Porous Media

A porous medium is a common name for a large group of materials and domains. Groundwater aquifers, oil reservoirs, human skin and wood, are examples of porous media. A common property of all these materials is a special structure where part of the domain is covered by a solid skeleton, also refereed to as the matrix, and the rest consists of pores filled with fluids. When space is filled with one or several fluids, we call it single-phase or multi-phase flow respectively. The fluids can be gases or liquids, or both. In this thesis we consider porous media filled with oil and water.

A porous medium has a complex geometry and cannot be described point-wise because each single point in space may contain only solid or only fluid. That is why in this work we use the common approach where instead of a single point we consider the *Representative Elementary Volume (REV)*. The REV is the smallest possible volume which contains a representative amount of void and solid such that we can define the mean (macro) properties with it. The size of the REV should be restricted so that properties of the medium are still local. Figure (2.2) shows one way of choosing the size of REV by examining the void fraction. If the size of REV is too small there will be random oscillations in the void fraction function, however with growth of REV's size an equilibrium is reached which means that the best size of REV (V1 on the Figure 2.2) is found.

The volume of voids in REV divided by its total volume is called *porosity*. Porosity can be a function of time or space.

2.1.2 Mass Conservation Equation for Single-Phase Flow

We can now derive the equation of mass conservation for an arbitrary domain Ω (Figure 2.1).

The change of the mass in the domain can only be caused by the mass flux through the boundaries and by mass sources or sinks within the domain. In mathematical terms

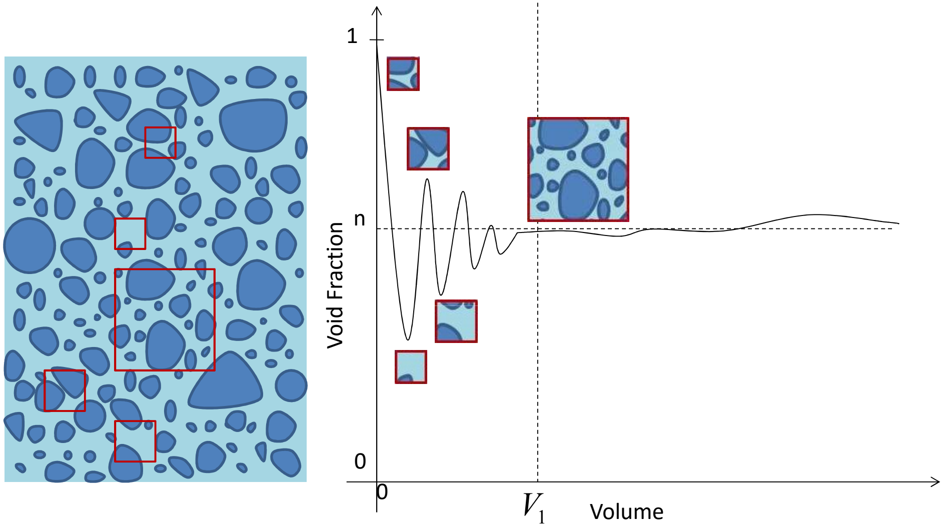

Figure 2.2: Schematic representation of the relationship between the void fraction and the REV volume (modified after [3]).

we can write this statement as:

$$\frac{\partial}{\partial t}\int_{\Omega}\eta dV = -\int_{\partial\Omega}\boldsymbol{f}\cdot\boldsymbol{n}ds + \int_{\Omega}rdV, \tag{2.1}$$

where η is mass per volume, $\boldsymbol{f}$ represents the mass flux vector, $\boldsymbol{n}$ - the outer normal vector and r is any source or sink term within the volume. The units of r are mass per volume per time.

If there are no sources or sinks ($r = 0$), or if r represents the external forces, then m is locally preserved and we call equation (2.1) a *conservation law*. Otherwise, if r includes internal changes (like chemical reactions) then it is called *mass balance* or *transport equation*.

In the case of single-phase flow, the following relations hold:

$$\eta = \rho\phi, \quad \boldsymbol{f} = \rho\boldsymbol{u}, \quad r = F, \tag{2.2}$$

where ρ is the density of the fluid, ϕ is the porosity, $\boldsymbol{u}$ is the volumetric flux vector and F represents the source or sink of mass term. The *volumetric flux* is a volumetric flow rate per area, its dimension is $[LT^{-1}]$. The volumetric flux represents the fluid volume going through a column in a unit time.

For all time-invariant domains Ω the Leibniz integral rule and the Gauss theorem yield

$$\int_{\Omega}\Big(\frac{\partial\phi\rho}{\partial t} + \bigtriangledown\cdot\rho\boldsymbol{u} - F\Big)dV = 0. \tag{2.3}$$

Under the conditions of sufficient smoothness of the functions in (2.3) one can obtain the differential form of the mass conservation equation:

$$\frac{\partial \phi \rho}{\partial t} + \bigtriangledown \cdot \rho \boldsymbol{u} = F. \tag{2.4}$$

If the fluid density depends on time, pressure, temperature etc. the fluid is called *compressible*. Otherwise, it is called *incompressible*. In this thesis we assume that the fluids we work with are incompressible. Therefore we rewrite the equation (2.4) as follows:

$$\frac{\partial \phi}{\partial t} + \bigtriangledown \cdot \boldsymbol{u} = \frac{F}{\rho}. \tag{2.5}$$

2.1.3 Mass Conservation Equation for Two-Phase Flow

In order to derive two-phase flow extension of the mass conservation equation we first have to introduce new quantities.

The modeling reservoir contains two fluids: oil and water, which brings us to the fluid property called the *fluid phase saturation*, S_α. It is a dimensionless quantity defined as the fraction of pore space occupied by fluid α in REV. It is clear that $0 \leq S_\alpha \leq 1$, and the sum of all fluid saturations in multi-phase flow will be equal to 1, $\sum_\alpha S_\alpha = 1$.

We will also use the common assumption that our fluids are *immiscible* (there is no mass exchange between the fluids) which is usually the case for oil and water.

In that case, the equation (2.4) should hold for each fluid phase α:

$$\frac{\partial \rho_\alpha \phi S_\alpha}{\partial t} + \bigtriangledown \cdot \rho_\alpha \boldsymbol{u_\alpha} = F_\alpha. \tag{2.6}$$

Here ρ_α is the density and $\boldsymbol{u_\alpha}$ is the volumetric flux of the fluid α. Note that the flux of each fluid is different. In the next section we present the equations for the fluxes $\boldsymbol{u_\alpha}$.

In the case of incompressible flow and solid (porosity ϕ does not change with time) we can write the equation above as follows,

$$\phi \frac{\partial S_\alpha}{\partial t} + \bigtriangledown \cdot \boldsymbol{u_\alpha} = \frac{F_\alpha}{\rho_\alpha}. \tag{2.7}$$

Having considered mass conservation equations let us take a look at how the flux can be approximated.

2.2 Darcy's Law

Darcy's law is a fundamental constitutive equation that describes the fluid flow through porous media. It was experimentally derived by the French engineer Henric Darcy in the middle of the 19th century, and to this day it forms the basis of the mathematical modeling of flow in porous media. Darcy's law has many extensions. In this section we present the original formulation for the hydraulic head, as well as two extensions - Darcy's law for single-phase and two-phase flow models.

2.2.1 Darcy's Law for the Hydraulic Head

To formulate Darcy's law we first have to introduce some quantities. One important quantity that we have already mentioned is a *hydraulic head* h, its dimension is length, or $[L]$. It shows the direction of the flow: ground water flows from regions with higher hydraulic conductivity to regions with lower values of h. The hydraulic head represents the total energy of the water. In this thesis we consider *laminar flow* meaning that the velocity of the flow is sufficiently small and we can neglect the kinetic energy. This means that the total energy, which in general is the sum of kinetic and potential energy, is represented only by the potential energy. The potential energy itself is a sum of pressure potential and gravitational potential inside the aquifer which can be expressed as follows:

$$m\rho h = pV + mgz. \tag{2.8}$$

As a consequence, we obtain a useful expression for the hydraulic head in terms of the pressure:

$$h = \frac{p}{\rho g} + z. \tag{2.9}$$

The second quantity is *hydraulic conductivity* κ. In general case, the hydraulic conductivity is a tensor with dimension $[LT^{-1}]$. It is a function of both the porous medium and the fluid, and it indicates how easily the fluid flows through the material. It can be expressed as

$$\kappa = \frac{k\rho g}{\mu}, \tag{2.10}$$

where μ is viscosity, ρ fluid density, g gravitational acceleration and k is a very important property of the porous medium called *intrinsic permeability*, or just *permeability*. It measures the ability of fluid to flow through porous media. It has dimension $[L^2]$, however, the derived units called *Darcy* or *milliDarcy* are usually used, $(1Darcy \sim 10^{-12}m^2)$. In general case permeability is a space dependent tensorial quantitiy. In certain cases there can be simplifications.

Darcy's law presents a relation between all these quantities. In the differential form it says:

$$\boldsymbol{u} = -\boldsymbol{\kappa} \bigtriangledown \boldsymbol{h}, \tag{2.11}$$

where $\boldsymbol{u}$ is the volumetric flow rate, κ is the hydraulic conductivity and h is the hydraulic head.

2.2.2 Darcy's Law for Single-Phase Flow

In our work we use the pressure formulation of Darcy's law which can be easily obtained from (2.11) using the expression for the hydraulic conductivity (2.10) and the hydraulic head (2.9):

$$\boldsymbol{u} = -\frac{\boldsymbol{k}}{\mu}(\bigtriangledown p + \rho g \bigtriangledown z) = -\frac{\boldsymbol{k}}{\mu}(\bigtriangledown p - \rho \boldsymbol{g}), \tag{2.12}$$

where $\boldsymbol{g} = -g\boldsymbol{e_z} = (0,\, 0,\, -g)^T$ is the gravitational acceleration vector.

2.2.3 Darcy's Law for Two-Phase Flow

Darcy's law in the form (2.12) is used to model single-phase flow when all the pores are filled with one fluid and the whole pore space in available for this fluid to flow. To model two-phase flow we need to deal with a system where part of the pores is already occupied with one fluid, which obstructs the flow of the second fluid. This implies lower permeability for both fluids. That is why we need to introduce the *relative permeability*, $k_{r,\alpha} = k_{r,\alpha}(S_\alpha)$, which is different for each phase α. In the general case it is anisotropic. There exists various models based on experimental data that parameterize the relative permeability. It is most commonly approximated as a scalar nonlinear function of the saturation. In this thesis we used van Genuchten parametrization, see Appendix A, (5).

We can now formulate Darcy's law for the multi-phase flow as an extension of equation (2.12) as follows:

$$\boldsymbol{u}_\alpha = -\frac{\boldsymbol{k}_{r,\alpha}\boldsymbol{k}}{\mu_\alpha}(\bigtriangledown p_\alpha - \rho_\alpha \boldsymbol{g}). \tag{2.13}$$

We will now introduce a new function, called the *phase mobility*, $\boldsymbol{\lambda_\alpha}$, which is defined as the ratio of the relative permeability function to the phase viscosity, $\lambda_\alpha = \dfrac{\boldsymbol{k}_{r,\alpha}}{\mu_\alpha}$. With this definition Darcy's law may be written as follows:

$$\boldsymbol{u}_\alpha = -\boldsymbol{\lambda_\alpha k}(\bigtriangledown p_\alpha - \rho_\alpha \boldsymbol{g}). \tag{2.14}$$

Later on we will refer to the equation (2.14) as Darcy's law for two-phase flow.

2.3 Governing Equations and Common Simplifications of the Two-Phase Flow Model

The two main equations on which our model is based are Darcy's law for two-phase flow and the mass conservation equations for each fluid phase. We have already introduced both of them, so we can write the mass conservation law (2.4) and Darcy's law for two-phase flow (2.14) for each phase:

$$\begin{aligned}
\boldsymbol{u}_w &= -\boldsymbol{\lambda_w k}(\bigtriangledown p_w - \rho_w \boldsymbol{g}),\\
\boldsymbol{u}_n &= -\boldsymbol{\lambda_n k}(\bigtriangledown p_n - \rho_n \boldsymbol{g}),\\
\phi\frac{\partial S_w}{\partial t} &+ \bigtriangledown \cdot \boldsymbol{u_w} = \frac{F_w}{\rho_w},\\
\phi\frac{\partial S_n}{\partial t} &+ \bigtriangledown \cdot \boldsymbol{u_n} = \frac{F_n}{\rho_n}.
\end{aligned} \tag{2.15}$$

To derive this system of equation we made a simplifying assumption of: we assumed incompressibility of fluids and solid matrix (ϕ, ρ_w, ρ_n are constants). Also, as before, we assumed immiscible and non-diffusive fluids.

Let us mention that even under the assumptions above, system (2.15) is not closed. For example, in 2D case we have six equations with eight unknowns (u_w^i, u_n^i, p_w, p_n, S_w, S_n). After adding the equation for the sum of two saturations $S_w + S_n = 1$ we still miss one equation.

To close the system we have to investigate carefully the relation between pressures p_n and p_w. Note that the pressure on each side of the fluid-fluid interface may be different because of the inter-facial tension between the two phases. The fluid acts differently in contact with the solid part of the reservoir. Figure (2.3) shows the differences in the contact angle of water and oil with the surface. The fluid that is preferentially attracted by the solid is called the *wetting fluid*. The contact angle of such fluid with the solid is less then 90°, $\theta < 90°$. The other fluid is referred to as the *non-wetting fluid*. In this thesis we model the reservoir that contains oil and water, and water in this situation is a wetting fluid, while oil is non-wetting.

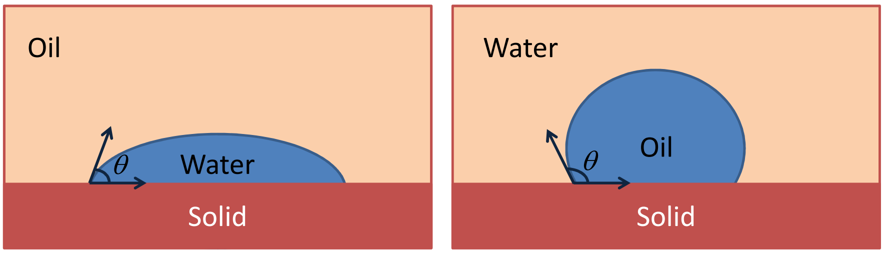

Figure 2.3: The contact angle between the wetting fluid (water) and a solid (a), and between the non-wetting fluid (oil) and a solid.

We define the difference between the phase pressures as the *capillary pressure*:

$$p_c = p_n - p_w. \tag{2.16}$$

Generally, the function p_c is chosen based on laboratory experiments. As laboratory measurements can be taken only when an equilibrium is reached, it is usually parameterized as an algebraic function of the phase saturation, $p_c = p_c(S_w)$ [3]. Including the capillary pressure makes the model more realistic, even though the parametrization of the capillary function is a complicated problem itself and requires deep studying. There are several types of parametrization, for example the van Genuchten parametrization, see Appendix 5.

After adding appropriate boundary and initial conditions we get a closed system of equations that describe two-phase flow:

$$\begin{aligned}
&\boldsymbol{u}_w = -\boldsymbol{\lambda_w}\boldsymbol{k}(\bigtriangledown p_w - \rho_w \boldsymbol{g}),\\
&\boldsymbol{u}_n = -\boldsymbol{\lambda_n}\boldsymbol{k}(\bigtriangledown p_n - \rho_n \boldsymbol{g}),\\
&\phi\frac{\partial S_w}{\partial t} + \bigtriangledown\cdot\boldsymbol{u_w} = \frac{F_w}{\rho_w},\\
&\phi\frac{\partial S_n}{\partial t} + \bigtriangledown\cdot\boldsymbol{u_n} = \frac{F_n}{\rho_n},\\
&S_w + S_n = 1,\\
&p_n - p_w = p_c(S_w),\\
&S_w^0 = S_w(x,t_0), \quad \overline{p}^0 = \overline{p}(x,t_0),\\
&S_w|_{\partial\Omega} = S_w^\Gamma, \quad \overline{p}|_{\partial\Omega} = \overline{p}^\Gamma.
\end{aligned} \tag{2.17}$$

2.4 Two-Phase Flow Model in Averaged Pressure Formulation

There are different ways to rewrite the system (2.17). In this thesis we work with the averaged pressure formulation which is a commonly used practical reformulation of the equations (2.17), where the averaged pressure and the saturation of the wetting fluid phase are used as the primary variables.

We consider a new function called the *averaged pressure*, which we define as

$$\overline{p} = \frac{p_n + p_w}{2}. \tag{2.18}$$

In order to express p_n and p_w in terms of the averaged pressure and the capillary pressure we can use (2.16) and (2.18):

$$p_w = \overline{p} - \frac{1}{2}p_c, \quad p_n = \overline{p} + \frac{1}{2}p_c. \tag{2.19}$$

Summing up two mass conservation equations from the system (2.15) results in following equation:

$$\phi\frac{\partial(S_w + S_n)}{\partial t} + \bigtriangledown\cdot(\boldsymbol{u_w} + \boldsymbol{u_n}) = \frac{F_w}{\rho_w} + \frac{F_n}{\rho_n}. \tag{2.20}$$

Since the sum of the saturations is equal to one, $S_w + S_n = 1$, the first term in the equation above is zero.

Substituting (2.19) into (2.15) and summing up the first two equations of the fluxes, we get the following expression for the total flux $\boldsymbol{u_\Sigma} = \boldsymbol{u_w} + \boldsymbol{u_n}$

$$\begin{aligned}\boldsymbol{u_\Sigma} &= \boldsymbol{u_w} + \boldsymbol{u_n} = -\boldsymbol{\lambda_w k} \bigtriangledown (\overline{p} - \frac{1}{2}p_c - \rho_w \boldsymbol{g}) - \boldsymbol{\lambda_n k} \bigtriangledown (\overline{p} + \frac{1}{2}p_c - \rho_n \boldsymbol{g}) \\ &= -\boldsymbol{k}(\boldsymbol{\lambda_w} + \boldsymbol{\lambda_n}) \bigtriangledown \overline{p} - \boldsymbol{k}\frac{\boldsymbol{\lambda_n} - \boldsymbol{\lambda_w}}{2} \bigtriangledown p_c + (\boldsymbol{\lambda_w}\rho_w + \boldsymbol{\lambda_n}\rho_n)\boldsymbol{k} \bigtriangledown \boldsymbol{g}.\end{aligned} \tag{2.21}$$

Let us now introduce the *total mobility* function, $\boldsymbol{\lambda_\Sigma}$:

$$\boldsymbol{\lambda_\Sigma} = \boldsymbol{\lambda_w} + \boldsymbol{\lambda_n}. \tag{2.22}$$

Combining equations (2.22), (2.21) and (2.20) and neglecting the gravity term (as we will model these equations in a 2D domain), we get a new equation which we will later refer to as the **pressure equation**:

$$-\bigtriangledown \cdot \boldsymbol{k}(\boldsymbol{\lambda_\Sigma} \bigtriangledown \overline{p} + \frac{\boldsymbol{\lambda_n} - \boldsymbol{\lambda_w}}{2} \bigtriangledown p_c) = \sum_{\alpha=n,w} \frac{F_\alpha}{\rho_\alpha} \tag{2.23}$$

In this work we use a common approach to identify unknown parameter functions as functions of S_w. Therefore in our model we combine the pressure equation with the mass conservation equation for the water phase and later on refer to it as **saturation equation**:

$$\phi\frac{\partial S_w}{\partial t} - \bigtriangledown \cdot \big(\boldsymbol{\lambda_w k} \bigtriangledown (\overline{p} - \frac{1}{2}p_c)\big) = \frac{F_w}{\rho_w}. \tag{2.24}$$

To ensure that the solution of our final system of equations is unique and that the problem is well-posed, we add initial conditions, appropriate boundary conditions and a parametrization for the functions λ_n, λ_w and p_c.

Finally, the two-phase flow model in the averaged pressure formulation takes the form:

$$\begin{aligned}&-\bigtriangledown \cdot \boldsymbol{k}(\boldsymbol{\lambda_\Sigma} \bigtriangledown \overline{p} + \frac{\boldsymbol{\lambda_n} - \boldsymbol{\lambda_w}}{2} \bigtriangledown p_c) = \sum_{\alpha=n,w} \frac{F_\alpha}{\rho_\alpha}, \\ &\phi\frac{\partial S_w}{\partial t} - \bigtriangledown \cdot \big(\boldsymbol{\lambda_w k} \bigtriangledown (\overline{p} - \frac{1}{2}p_c)\big) = \frac{F_w}{\rho_w}, \\ &S_w^0 = S_w(x, t_0), \quad \overline{p}^0 = \overline{p}(x, t_0), \\ &S_w|_{\partial\Omega} = S_w^\Gamma, \quad \overline{p}|_{\partial\Omega} = \overline{p}^\Gamma.\end{aligned} \tag{2.25}$$

Chapter 3

Numerical Modeling

The system (2.25) derived in the previous chapter cannot be solved analytically for general cases. Moreover, nonlinearities make numerical simulation extra challenging.

In this chapter we present the numerical schemes we use to solve the system (2.25). First, we give a short overview of the finite element method. Second, we examine when implicit and explicit methods can be applied, whether we can combine them, and how one can treat nonlinearities in implicit solutions. We briefly present the standard approaches (IMPES (Section 3.2) and Newton's method (Section 3.3.1)) and propose a new implicit scheme (see Section 3.3.2). In Section 3.3.3 we prove rigorously that the new scheme is global convergent. Only a relatively mild constraint on the time step size is required.

3.1 The Finite Element Method

The finite element method (FEM) is a powerful and commonly used technique for finding numerical solutions of partial differential equations. While it is difficult to give a specific date when FEM was invented, it is clear that this method was derived from the finite differences method in the 1950's [30]. As a computational method it originated in the engineering literature and the name of the finite element method appeared first in [8].

The finite element method has a solid theoretical foundation based on Sobolev space theory which brings it several advantages. The theoretical base gives it reliability, makes it easier to work with general boundary conditions, complex domain geometry, variable material properties, etc. In many cases exact error estimates for finite element solutions can be obtained [5].

First we discuss the space discretization, examine the model problem and then show how we applied the FEM technique to our model problem.

3.1.1 Discretization in Space

Oil reservoirs are 3D structures with highly complex geometry. However, most permeable formations have a horizontal extent that is much greater then the vertical extent. As a consequence, the flow is mostly horizontal and it is usually the case that the number of dimensions can be reduced to two.

In this thesis we solve the system (2.25) in a 2D domain. The domain is divided into a finite number of small sub-domains, also called elements. One can choose different types of elements. In this thesis we work with triangles. We cover Ω with a set $\mathcal{T}_h = T_1, ..., T_m$ of non-overlapping triangles T_i, such that $\Omega = \underset{T \in \mathcal{T}_h}{\cup T} = T_1 \cup T_2 \cup ... \cup T_m$ and no vertex of one triangle lies on the edge of another triangle. This process is called *triangulation*. Figure (3.1) presents an example of triangulation of a domain with complex structure.

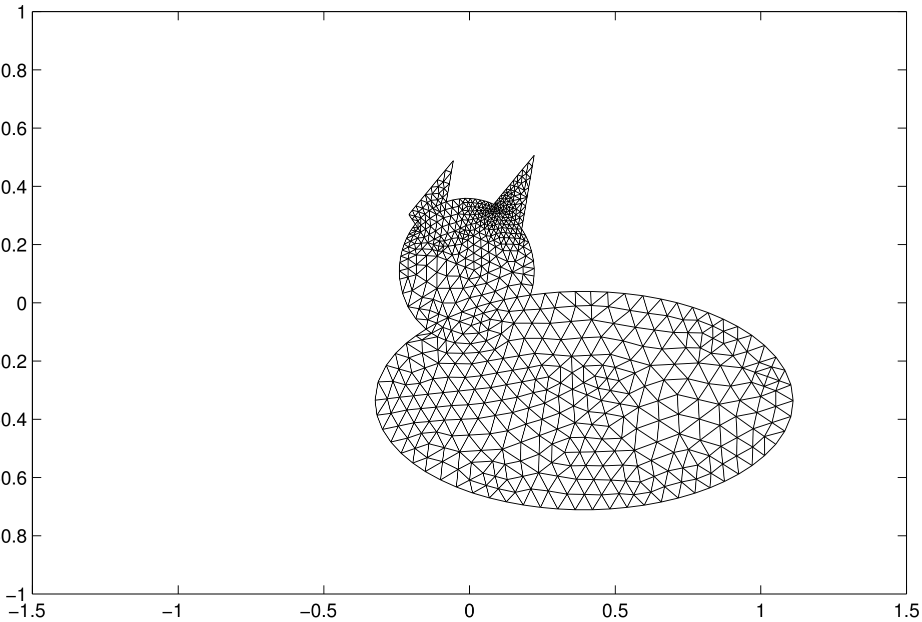

Figure 3.1: Example of a triangular mesh on a complex structured domain.

3.1.2 Variational Formulation of the Model Problem

Let us start by considering a model problem where we solve an elliptic PDE with zero boundary conditions. This model problem corresponds to the pressure equation (2.23) in a simpler form (as if we have no capillary pressure). We derive a variational formulation of this problem and provide guidance on applying the finite element method to solving this problem numerically. For the case of the variational formulation we search for a weak solution. This solution and its derivative should be square integrable [5]. The space of such functions is called H^1.

Model problem. Find $u \in H^1$ such that

$$- \bigtriangledown \cdot (a \bigtriangledown u(\boldsymbol{x})) = f(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \Omega, \tag{3.1}$$

$$u(\boldsymbol{x})|_{\partial\Omega} = 0, \tag{3.2}$$

where Ω is a bounded open domain in R^2 with a boundary $\partial\Omega$, a is a given weight function (a symmetric positive definite matrix), and $f(\boldsymbol{x})$ is also a known function.

In order to derive the variational formulation we multiply both sides of (3.1) by a test function $v(\boldsymbol{x}) \in H^1$ and integrate them over the whole domain Ω. From now on, we write a, u, f, v instead of $a(\boldsymbol{x}), u(\boldsymbol{x}), f(\boldsymbol{x}), v(\boldsymbol{x})$ for conciseness:

$$- \int_\Omega \bigtriangledown \cdot (a \bigtriangledown u)\, v\, dx = \int_\Omega f v\, dx.$$

Integrating by parts and using the zero boundary condition (3.2) we arrive at

$$\int_\Omega a \bigtriangledown u \cdot \bigtriangledown v\, dx = \int_\Omega f v\, dx.$$

This is the variational formulation of the model problem.

It is easy to see that $\int_\Omega a \bigtriangledown u \cdot \bigtriangledown v\, dx$ is a bilinear form. Let us introduce a new notation for it

$$a(u, v) \equiv \int_\Omega a \bigtriangledown u \cdot \bigtriangledown v\, dx, \tag{3.3}$$

and a linear functional

$$l(v) \equiv \langle f, v \rangle_{L_2} \equiv \int_\Omega f\, v\, dx, \tag{3.4}$$

where $\langle . , . \rangle_{L_2}$ is the L_2-inner product.

Now we can construct the finite element method (FEM) for (3.1)-(3.2).

At the first step of FEM we define a finite-dimensional space $V_h \in H^1$ of piecewise linear continuous functions $V_h = \{v : v \text{ is continuous on } \Omega, \text{ linear on each } T_i, v = 0 \text{ on } \Gamma\}$. The basis functions $\phi_j \in V_h, j = 1, \ldots, M$ then are defined for each node $N_i, i = 1, \ldots, M$ of $\mathcal{T}_h$, excluding nodes on the boundary:

$$\phi_j(N_i) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \qquad i, i = 1, \ldots, M.$$

Any function $v \in V_h$ now has a representation through these basis functions

$$v(\boldsymbol{x}) = \sum_{i=1}^{M} \alpha_i \phi_i(\boldsymbol{x}), \quad \alpha_i = v(N_i), \text{ for } x \in \Omega \cup \Gamma.$$

We can now formulate the finite element method as follows.

FEM Formulation. Find $u_h \in V_h$ such that

$$a(u_h, v_h) = l(v_h) \quad v_h \in V_h. \tag{3.5}$$

The unknown function is represented as a combination of basis functions:

$$u_h(\boldsymbol{x}) = \sum_{i=1}^{M} \alpha_i \phi_i(\boldsymbol{x}), \quad \alpha_i = u_h(N_i). \tag{3.6}$$

Then we substitute this representation into (3.5), use the basis function ϕ_i as the test function v_h and get:

$$\sum_{j=1}^{M} \alpha_j \, a(\phi_j, \phi_i) = l(\phi_i), \quad i = 1, \ldots, M. \tag{3.7}$$

This can be written as a system of linear equations

$$A\xi = b, \tag{3.8}$$

where $\xi = (\alpha_1, \ldots, \alpha_M)^T$, $b = (l(\phi_1), \ldots, l(\phi_M))^T$ and the matrix

$$A = \sum_{K \in \mathcal{T}_h} \int_K a \bigtriangledown \phi_i \cdot \bigtriangledown \phi_j \, dx \tag{3.9}$$

is called the *weighted stiffness matrix*. To calculate this matrix we only need to compute the gradient of the linear basis functions that are non-zero on the triangle. Each basis function is equal to one in one node of the triangular K and zero in two others. There are numerous books dedicated to the finite element method, where one can find a detailed description of the matrix assembling, see e.g. [5].

By solving the system (3.8) we can easily find the unknown function u_h. There are various works where the existence and uniqueness solution is proven, for example [18]. Here we will only mention that the Lax-Milgram theorem provides the list of requirements that must be satisfied for existence and uniqueness of the solution and the book [18] contains a prove that our model problem satisfies this theorem conditions.

3.2 The IMPES method

Note that the equation (2.24) depends on time and space. It means that we need not only space discretization, but also discretization in time.

We begin with the most common solution procedure the IMplicit Pressure Explicit Saturation method (IMPES)[26]. As it says in the name of the scheme, we solve the saturation equation explicitly. In other words, we use the standard finite-differences approximation for the time derivative

$$\frac{\partial S_w}{\partial t} = \frac{S_w^{n+1} - S_w^n}{\Delta t}, \tag{3.10}$$

where $\Delta t = t^{n+1} - t^n$ and $0 = t^0 \leq t^1 \leq \cdots \leq t^N = T$. The initial information is available from the initial conditions.

All other coefficients depending on S_w are computed using the value of the saturation found at the previous time step. This scheme is a well-known fixed step solver called the *forward* or the *explicit Euler method.* Now we can write the finite element approximation of the equations (2.25) in terms of solving it with the IMPES method.

Find $\overline{p}_h^{n+1}, S_h^{n+1} \in V_h \; \forall \upsilon_h \in V_h$ such that

$$-\langle \nabla \cdot \left(\lambda_\Sigma(S_h^n) \bigtriangledown \overline{p}_h^{n+1} + \frac{\lambda_n(S_h^n) - \lambda_w(S_h^n)}{2} \bigtriangledown p_c(S_h^n) \right), \upsilon_h \rangle = \langle F_{pr}, \upsilon_h \rangle, \tag{3.11}$$

$$\langle \phi \frac{S_h^{n+1} - S_h^n}{\Delta t} - \nabla \cdot \left(\lambda_w(S_h^n) \bigtriangledown (\overline{p}^{n+1} - \frac{1}{2} p_c(S_h^n)) \right), \upsilon_h \rangle = \langle F_{sat}, \upsilon_h \rangle, \tag{3.12}$$

where $F_{pr} = \sum_{\alpha=n,w} \dfrac{F_\alpha}{\rho_\alpha}$ and $F_{sat} = \dfrac{F_w}{\rho_w}$.

First of all, we need to write the variational formulation for our system (2.25) in the same way as we did for the model problem.

We again multiply the equations in the system by the test function, integrate them over the domain Ω and apply the divergence theorem. Hence, for the pressure equation we obtain

$$\int_\Omega \left(\lambda_\Sigma(S_h^n) \bigtriangledown \overline{p}_h^{n+1} + \frac{\lambda_n(S_h^n) - \lambda_w(S_h^n)}{2} \bigtriangledown p_c(S_h^n) \right) \cdot \nabla \upsilon_h \, dx = \int_\Omega F_{pr} \upsilon_h \, dx.$$

Let us denote $\lambda_{dif} = \lambda_n - \lambda_w$, then

$$\int_\Omega \lambda_\Sigma(S_h^n) \bigtriangledown \overline{p}_h^{n+1} \cdot \nabla \upsilon_h \, dx = \int_\Omega F_{pr} \upsilon_h \, dx - \int_\Omega \frac{\lambda_{dif}(S_h^n)}{2} \bigtriangledown p_c(S_h^n) \cdot \nabla \upsilon_h. \tag{3.13}$$

The same procedure should be applied to the saturation equation. Note that the test function for the saturation equation may belong to a different space than the test function for the pressure equation. And what is more important, the saturation function itself may not belong to the same class as the pressure function.

Nevertheless, we used the same approximation for the saturation equation:

$$\int_\Omega \phi \frac{S_h^{n+1} - S_h^n}{\Delta t} \upsilon_h \, dx + \int_\Omega \left(\lambda_w(S_h^n) \bigtriangledown (\overline{p}_h^{n+1} - \frac{1}{2} p_c(S_h^n)) \right) \cdot \bigtriangledown \upsilon_h \, dx = \int_\Omega F_{sat} \upsilon_h \, dx,$$

$$\begin{aligned} &\frac{\phi}{\Delta t} \int_\Omega S_h^{n+1} \upsilon_h \, dx - \frac{\phi}{\Delta t} \int_\Omega S_h^n \upsilon_h \, dx + \int_\Omega \lambda_w(S_h^n) \bigtriangledown \overline{p}_h^{n+1} \cdot \bigtriangledown \upsilon_h \, dx \\ &- \int_\Omega \frac{1}{2} \lambda_w(S_h^n) \bigtriangledown P_c(S_h^n) \cdot \bigtriangledown \upsilon_h \, dx = \int_\Omega F_{sat} \upsilon_h \, dx, \end{aligned}$$

$$\begin{aligned} &\frac{\phi}{\Delta t} \int_\Omega S_h^{n+1} \upsilon_h \, dx = \frac{\phi}{\Delta t} \int_\Omega S_h^n \upsilon_h \, dx - \int_\Omega \lambda_w(S_h^n) \bigtriangledown \overline{p}_h^{n+1} \cdot \bigtriangledown \upsilon_h \, dx \\ &+ \int_\Omega \frac{1}{2} \lambda_w(S_h^n) \bigtriangledown P_c(S_h^n) \cdot \bigtriangledown \upsilon_h \, dx + \int_\Omega F_{sat} \upsilon_h \, dx. \end{aligned} \tag{3.14}$$

Let us represent the unknown functions as combinations of the basis functions:

$$\overline{p}_h^{n+1}(\boldsymbol{x}) = \sum_{i=1}^{M} \xi_i \phi_i(\boldsymbol{x}), \quad \xi_i = p_h^{n+1}(x_i), \tag{3.15}$$

$$S_h^{n+1}(\boldsymbol{x}) = \sum_{i=1}^{M} \eta_i \phi_i(\boldsymbol{x}), \quad \eta_i = S_h^{n+1}(x_i). \tag{3.16}$$

We use the corresponding basis function ϕ_i as the test function υ_h.

The phase mobility and the capillary pressure functions are approximated by their average values on each triangle.

In the numerical implementation we work with the equations (3.13)-(3.14) element-wise. It means that all the values are first calculated over each triangle and then combined together to get a linear system similar to the one we obtained for the model problem.

Let us write the equations (3.13)-(3.14) for each element $K \in \mathcal{T}_h$:

$$\int_K \xi_i \lambda_\Sigma^0 \bigtriangledown \phi_i \cdot \bigtriangledown \phi_j \, dx = \int_K F_{pr} \phi_i \, dx - \int_K \frac{\lambda_{dif}^0}{2} p_{c,i} \bigtriangledown \phi_i \cdot \bigtriangledown \phi_j, \tag{3.17}$$

$$\begin{aligned} &\phi \int_K \eta_i^{n+1} \phi_i \phi_j \, dx = \phi \int_K \eta_i^n \phi_i \phi_j \, dx - \Delta t \int_K \lambda_w^0 x i_i \lambda_\Sigma^0 \bigtriangledown \phi_i \cdot \bigtriangledown \phi_j \, dx \\ &+ \Delta t \int_K \frac{1}{2} \lambda_w^0 p_{c,i} \bigtriangledown \phi_i \cdot \bigtriangledown \phi_j + \Delta t \int_K F_{sat} \phi_i \, dx. \end{aligned} \tag{3.18}$$

In spite of the complicated form, the equations above represent a system of linear equations. The last step in obtaining the system is to compute the stiffness matrix

$$A = \sum_{K \in \mathcal{T}_h} \int_K \bigtriangledown \phi_i \bigtriangledown \phi_j \, dx \tag{3.19}$$

and the matrix

$$B = \sum_{K \in \mathcal{T}_h} \int_K \phi_i \phi_j \, dx. \tag{3.20}$$

We already know how to deal with the stiffness matrix (3.9) from the model problem, and in order to calculate the matrix B we use the following formula [12]:

$$\iint_A N_1^i N_2^j N_3^k dA = \frac{i!j!k!(2A)}{(i+j+k+2)!}. \tag{3.21}$$

The boundary conditions (BCs) are another important aspect. There are several types of boundary conditions and of their combinations. In this thesis we consider two types of BCs, Dirichlet's:

$$u(\boldsymbol{x}) = u_D(\boldsymbol{x}) \; on \; \Gamma, \tag{3.22}$$

and Neumann's $\dfrac{\partial u}{\partial n}$ on Γ. We implemented both types of boundary conditions, however, this is not the main part of our research, so we just mention that we used well known techniques for treating boundary conditions, see e.g. [18].

3.3 Fully Implicit Scheme

We work with the two-phase flow model (2.25) that consists of two coupled nonlinear partial differential equations. The nonlinearity causes numerical difficulties for solving them.

One common approach to solve the system (2.25), IMPES, was already presented in the previous section. This type of time discretization provides a good tool for finding the numerical solution of the system as it eliminates nonlinearities in the equations. However, an explicit solving of the saturation equation causes stability problems and imposes restrictions on the size of the time-step.

Another approach is the fully implicit scheme. In this scheme the equations are coupled and simultaneously solved at each time step; all the terms are treated implicitly, including the capillary pressure. An advantage of this method is the unconditional stability, but it cannot help in dealing with nonlinearities.

The nonlinear fully discrete variational formulation of our system (2.25) at the time t_n is written as follows.

Find $\overline{p}_h^{n+1}, S_h^{n+1} \in V_h$ such that the following equations are satisfied $\forall \upsilon_h \in V_h$:

$$-\langle \nabla \cdot \left(\lambda_\Sigma(S_h^{n+1}) \nabla \overline{p}_h^{n+1} + \frac{\lambda_n(S_h^{n+1}) - \lambda_w(S_h^{n+1})}{2} \nabla p_c(S_h^{n+1}) \right), \upsilon_h \rangle = \langle F_{pr}, \upsilon_h \rangle, \tag{3.23}$$

$$\langle \phi \frac{S_h^{n+1} - S_h^n}{\Delta t} - \nabla \cdot \left(\lambda_w(S_h^{n+1}) \nabla (\overline{p}^{n+1} - \frac{1}{2} p_c(S_h^{n+1})) \right), \upsilon_h \rangle = \langle F_{sat}, \upsilon_h \rangle. \tag{3.24}$$

As $\upsilon_h \in V_h$ it means that υ_h is equal to zero on a boundary of the domain, which means that after applying the divergence theorem the equations above become:

$$\langle \left(\lambda_\Sigma(S_h^{n+1}) \nabla \overline{p}_h^{n+1} + \frac{\lambda_n(S_h^{n+1}) - \lambda_w(S_h^{n+1})}{2} \nabla p_c(S_h^{n+1}) \right), \nabla \upsilon_h \rangle = \langle F_{pr}, \upsilon_h \rangle, \tag{3.25}$$

$$\langle \phi \frac{S_h^{n+1} - S_h^n}{\Delta t}, \upsilon_h \rangle + \langle \left(\lambda_w(S_h^{n+1}) \nabla (\overline{p}_h^{n+1} - \frac{1}{2} p_c(S_h^{n+1})) \right), \nabla \upsilon_h \rangle = \langle F_{sat}, \upsilon_h \rangle. \tag{3.26}$$

The equations (3.25)-(3.26) are still coupled nonlinear equations and in order to solve them numerically one should think about a method that is suitable for solving such equations. One of the possibilities is Newton method (see Section 3.3.1) where the pressure and the saturation equations are coupled and solved as one at each iteration. The benefit of this method is the stability and the second order of convergence. However, in terms of the computational cost and the memory requirements it is an expensive method because it requires Jacobian matrix computation at each iteration. Another concern is how close the initial guess is to the true solution, which results in the restrictions on the time-step.

Thus, it is natural to come up with an IMPES-based solver that would maintain cheap computational costs of IMPES while relaxing its stability constraints. Such improved versions of the classical IMPES scheme were presented in several works, see [19, 20, 25, 6]. One of them is the iterative IMPES where equations are split and solved at each iteration using the IMPES method. This approach has a serious disadvantage. Decoupling the pressure and the saturation equations implies an explicit treatment of the capillary pressure which results in additional restrictions on the time step. The authors of [20] developed a new iterative IMPES scheme where they use a linear approximation of the capillary pressure function at the current iteration. The authors proved that such scheme applied to two-phase incompressible flow written for the pressure potentials is stable. For the spacial discretization they used the cell-centered finite difference method.

We also gain an understanding of implicit treatment of the capillary pressure function as with explicit solving we cannot guarantee the convergence of the method. That is why we developed a new implicit scheme that is based on the iterative IMPES scheme but handles the capillary pressure implicitly. A detailed explanation of the new implicit scheme is provided in Section 3.3.2 and Section 3.3.3 presents a rigorous proof of its convergence.

3.3.1 Newton's Method

Newton's method (also called the Newton–Raphson method) is one of the most common and powerful techniques to find a numerical approximation for the roots (or zeros) for systems of nonlinear equations. The method has the second order of convergence, see [29].

We want to apply Newton's method for finding the numerical solution of the discretized nonlinear system resulting from (3.25)-(3.26) and compare its performance with IMPES and our new implicit scheme.

First, we introduce Newton's method for multiple variables. Assume we are looking for the solution of the system

$$\boldsymbol{K}(\boldsymbol{u}) = \boldsymbol{f}. \tag{3.27}$$

For this system we define the residual as

$$\boldsymbol{r}(\boldsymbol{u}) = \boldsymbol{K}(\boldsymbol{u}) - \boldsymbol{f}. \tag{3.28}$$

In other words, we are solving the equation $\boldsymbol{r}(\boldsymbol{u}) = 0$. In order to obtain the classical Newton's method for this equation we follow derivation from [29]. Let $\boldsymbol{\xi}$ be a zero of the function $\boldsymbol{r}$ which is differentiable in the neighbourhood $N(\boldsymbol{\xi})$. Then the Taylor expansion of $\boldsymbol{r}$ about $\boldsymbol{u}^0 \in N(\boldsymbol{\xi})$ is

$$\boldsymbol{r}(\boldsymbol{\xi}) = 0 = \boldsymbol{r}(\boldsymbol{u}^0+(\boldsymbol{\xi}-\boldsymbol{u}^0)) = \boldsymbol{r}(\boldsymbol{u}^0)+D\boldsymbol{r}(\boldsymbol{u}^0)(\boldsymbol{\xi}-\boldsymbol{u}^0)+D^2\boldsymbol{r}(\boldsymbol{u}^0)\frac{(\boldsymbol{\xi}-\boldsymbol{u}^0)^2}{2!}+\dots \tag{3.29}$$

If we neglect terms with the second order derivative and higher, we get an estimation for the root of the equation (3.28):

$$\overline{\boldsymbol{\xi}} \approx \boldsymbol{u}^0 - J^{-1}\boldsymbol{r}(\boldsymbol{u}^0)\boldsymbol{r}(\boldsymbol{u}^0), \tag{3.30}$$

where J is the Jacobian matrix of the residual and can be computed as:

$$J\boldsymbol{r}(\boldsymbol{u}^0) = \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{u}_j} = \begin{bmatrix} \frac{\partial \boldsymbol{r}_1(\boldsymbol{u})}{\partial \boldsymbol{u}_1} & \frac{\partial \boldsymbol{r}_1(\boldsymbol{u})}{\partial \boldsymbol{u}_2} & \cdots & \frac{\partial \boldsymbol{r}_1(\boldsymbol{u})}{\partial \boldsymbol{u}_n} \\ \frac{\partial \boldsymbol{r}_2(\boldsymbol{u})}{\partial \boldsymbol{u}_1} & \frac{\partial \boldsymbol{r}_2(\boldsymbol{u})}{\partial \boldsymbol{u}_2} & \cdots & \frac{\partial \boldsymbol{r}_2(\boldsymbol{u})}{\partial \boldsymbol{u}_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial \boldsymbol{r}_n(\boldsymbol{u})}{\partial \boldsymbol{u}_1} & \frac{\partial \boldsymbol{r}_n(\boldsymbol{u})}{\partial \boldsymbol{u}_2} & \cdots & \frac{\partial \boldsymbol{r}_n(\boldsymbol{u})}{\partial \boldsymbol{u}_n} \end{bmatrix}_{\boldsymbol{u}=\boldsymbol{u}^0}. \tag{3.31}$$

The approximation $\overline{\boldsymbol{\xi}}$ should be close to the unknown root, however, it still needs corrections. This brings us to the iteration process:

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n - J^{-1}\boldsymbol{r}(\boldsymbol{u}_n)\boldsymbol{r}(\boldsymbol{u}_n). \tag{3.32}$$

The equation (3.32) is the *Newton iteration formula.*

In order to apply Newton's method to the problem (3.25)-(3.26) we first need to write it in the matrix form (3.27). As we want to solve the system (3.25)-(3.26) with the help of the FEM, we need to derive FEM formulation similarly as we get it for IMPES in section (3.2). The matrix form of the problem (3.25)-(3.26) then can be written as:

$$\begin{bmatrix} \lambda_\Sigma(S_h^{n+1})A & 0 & \dfrac{\lambda_{dif}(S_h^{n+1})}{2}A \\ \Delta t\lambda_w(S_h^{n+1})A & \phi B & -\dfrac{\Delta t}{2}\lambda_w(S_h^{n+1})A \end{bmatrix} \times \begin{bmatrix} \overline{p}^{n+1} \\ S_h^{n+1} \\ p_c(S_h^{n+1}) \end{bmatrix} = \begin{bmatrix} \langle F_{pr}, \upsilon_h\rangle \\ \langle F_{sat}, \upsilon_h\rangle + \phi B S_h^n \end{bmatrix}, \tag{3.33}$$

where A and B are the same as is (3.19)-(3.20).

Thus, in our case K, u and f from (3.32) are written as follows:

$$\boldsymbol{K} = \begin{bmatrix} \lambda_\Sigma(S_h^{n+1})A & 0 & \dfrac{\lambda_{dif}(S_h^{n+1})}{2}A \\ \Delta t\lambda_w(S_h^{n+1})A & \phi B & -\dfrac{\Delta t}{2}\lambda_w(S_h^{n+1})A \end{bmatrix}, \tag{3.34}$$

$$\boldsymbol{u} = \begin{bmatrix} \overline{p}^{n+1} \\ S_h^{n+1} \\ p_c(S_h^{n+1}) \end{bmatrix}, \tag{3.35}$$

$$\boldsymbol{f} = \begin{bmatrix} \langle F_{pr}, \upsilon_h\rangle \\ \langle F_{sat}, \upsilon_h\rangle + \phi B S_h^n \end{bmatrix}. \tag{3.36}$$

To compute the Jacobian matrix $J(\boldsymbol{u}^n)$ we first write the system (3.33) for each triangle K. When computing the derivatives we exploit the sparsity of the system. It means that we compute the derivative on each triangle separately and then add the computed values to the proper positions in the final Jacobian matrix. We use the following approximation for the derivatives:

$$J = \begin{bmatrix} \dfrac{\partial \boldsymbol{r}_1^i}{\partial \overline{p}^j} & \dfrac{\partial \boldsymbol{r}_1^i}{\partial S_w^j} \\ \dfrac{\partial \boldsymbol{r}_2^i}{\partial \overline{p}^j} & \dfrac{\partial \boldsymbol{r}_2^i}{\partial S_w^j} \end{bmatrix}, \tag{3.37}$$

$$\frac{\partial \boldsymbol{r}_l^i}{\partial \overline{p}^j} = \frac{\boldsymbol{r}_l^i(\overline{p} + \varepsilon e_j, S_w) - \boldsymbol{r}_l^i(\overline{p}, S_w)}{\varepsilon}, \quad \forall i, j, l \tag{3.38}$$

where e_j is a vector with zeros in all components except of j's which is equal to one $e_j = (0, 0, \dots, 1, 0, \dots 0)^T$. The length of this vector is six which corresponds to the number of unknowns on each triangle (three unknown pressure values and three saturation values). This approximation looks like the standard finite-difference approximation for the derivatives, however it represents a discrete functional derivative with respect to model variables. The stopping criteria for the Newton iteration process is $||\boldsymbol{u}^{n+1} - \boldsymbol{u}^n||_{L_2} \leq \delta$.

Later in this thesis we will refer to the Newton's implementation of the implicit method as simply Newton's method.

3.3.2 The New Implicit Scheme

In the averaged pressure formulation of the two-phase flow model (2.25) the capillary pressure appears under the gradient, but it still depends on the unknown $S_h^{n+1,i+1}$. We approximate it as follows:

$$\bigtriangledown p_c(S_h^{n+1,i+1}) \sim p_c'(S_h^{n+1,i}) \bigtriangledown S_h^{n+1,i+1}. \tag{3.39}$$

Assume that we know the discrete solution at the fixed time t_n. Then in order to find a solution at the next time step we start the iteration process. The iterations start with the solution at the previous time step, i.e. $S_h^{n+1,0} = S_h^n$.

Then the new iteration scheme to solve (2.25) reads as follows.

Let S_h^n and $S_h^{n+1,i}$ be given. Find $\overline{p}_h^{n+1,i+1}, S_h^{n+1,i+1}$ such that

$$\langle \left(\lambda_\Sigma(S_h^{n+1,i}) \bigtriangledown \overline{p}_h^{n+1,i+1} + \frac{\lambda_n(S_h^{n+1,i}) - \lambda_w(S_h^{n+1,i})}{2} \bigtriangledown p_c(S_h^{n+1,i}) \right), \bigtriangledown \upsilon_h \rangle = \langle F_{pr}, \upsilon_h \rangle, \tag{3.40}$$

$$\langle \phi \frac{S_h^{n+1,i+1} - S_h^n}{\Delta t}, \upsilon_h \rangle + \langle \left(\lambda_w(S_h^{n+1,i}) \bigtriangledown (\overline{p}_h^{n+1,i+1} - \frac{1}{2} p_c(S_h^{n+1,i+1})) \right), \bigtriangledown \upsilon_h \rangle = \langle F_{sat}, \upsilon_h \rangle. \tag{3.41}$$

The stopping criterion is $||(S_h^{n+1,i} - S_h^{n+1,i+1})|| \leq \varepsilon$, $||(p_h^{n+1,i} - p_h^{n+1,i+1})|| \leq \varepsilon$.

3.3.3 Proof of Convergence of the New Implicit Iteration Scheme

In this section we present the convergence proof that was inspired by [24, 27]

In order to prove that the scheme above converges, we need to make the following assumptions.

(A1) The functions λ_w, λ_n, $\lambda_\Sigma = \lambda_w + \lambda_n$, $\lambda_{dif} = \lambda_w - \lambda_n$, p_h^{n+1}, p_c, $p_c' = \dfrac{dp_c(S_w)}{dS_w}$ and $\bigtriangledown p_c$ are Lipschitz continuous in the domain Ω (which also means that all of them are bounded in this domain).

(A2) p_c is a decreasing function, and, as a consequence, $p_c' \leq 0$.

From now on we will use the following notation:

$$e_p^{i+1} = p_h^{n+1} - p_h^{n+1,i+1}, \quad e_s^{i+1} = S_h^{n+1} - S_h^{n+1,i+1}. \tag{3.42}$$

Theorem 3.3.1. *Under the assumptions (A1)-(A2) the scheme (3.40)-(3.41) converges linearly when the time step satisfies (3.51).*

Proof. Subtracting (3.40) from (3.25), and (3.41) from (3.26) we get

$$\langle \lambda_\Sigma(S_h^{n+1}) \nabla \overline{p}_h^{n+1} - \lambda_\Sigma(S_h^{n+1,i}) \nabla \overline{p}_h^{n+1,i+1}, \nabla \upsilon_h \rangle + \Big\langle \frac{\lambda_n(S_h^{n+1}) - \lambda_w(S_h^{n+1})}{2} \nabla p_c(S_h^{n+1})$$
$$- \frac{\lambda_n(S_h^{n+1,i}) - \lambda_w(S_h^{n+1,i})}{2} \nabla p_c(S_h^{n+1,i}), \nabla \upsilon_h \Big\rangle = 0,$$

$$\frac{\phi}{\Delta t} \langle e_s^{i+1}, \upsilon_h \rangle + \langle \left(\lambda_w(S_h^{n+1}) \nabla \overline{p}_h^{n+1} - \lambda_w(S_h^{n+1,i}) \nabla \overline{p}_h^{n+1,i+1}\right), \nabla \upsilon_h \rangle$$
$$+ \frac{1}{2} \langle \lambda_w(S_h^{n+1,i}) \nabla p_c(S_h^{n+1,i+1}) - \lambda_w(S_h^{n+1}) \nabla p_c(S_h^{n+1}), \nabla \upsilon_h \rangle = 0.$$

We can rewrite the first equation as

$$\langle \lambda_\Sigma(S_h^{n+1}) \nabla \overline{p}_h^{n+1} \mp \lambda_\Sigma(S_h^{n+1,i}) \nabla \overline{p}_h^{n+1} - \lambda_\Sigma(S_h^{n+1,i}) \nabla \overline{p}_h^{n+1,i+1}, \nabla \upsilon_h \rangle$$
$$+ \Big\langle \frac{\lambda_n(S_h^{n+1}) - \lambda_w(S_h^{n+1})}{2} \nabla p_c(S_h^{n+1}) \mp \frac{\lambda_n(S_h^{n+1,i}) - \lambda_w(S_h^{n+1,i})}{2} \nabla p_c(S_h^{n+1})$$
$$- \frac{\lambda_n(S_h^{n+1,i}) - \lambda_w(S_h^{n+1,i})}{2} \nabla p_c(S_h^{n+1,i}), \nabla \upsilon_h \Big\rangle = 0,$$

which is further equivalent to

$$\langle (\lambda_\Sigma(S_h^{n+1}) - \lambda_\Sigma(S_h^{n+1,i})) \nabla \overline{p}_h^{n+1}, \nabla \upsilon_h \rangle + \langle \lambda_\Sigma(S_h^{n+1,i}) \nabla e_p^{i+1}, \nabla \upsilon_h \rangle$$
$$+ \Big\langle \left(\frac{\lambda_n(S_h^{n+1}) - \lambda_w(S_h^{n+1})}{2} - \frac{\lambda_n(S_h^{n+1,i}) - \lambda_w(S_h^{n+1,i})}{2} \right) \nabla p_c(S_h^{n+1}), \nabla \upsilon_h \Big\rangle$$
$$+ \Big\langle \frac{\lambda_n(S_h^{n+1,i}) - \lambda_w(S_h^{n+1,i})}{2} \nabla \left(p_c(S_h^{n+1}) - p_c(S_h^{n+1,i})\right), \nabla \upsilon_h \Big\rangle = 0.$$

Let us test the equation above with $\upsilon_h = e_p^{i+1}$.

$$\lambda_\Sigma^0 || \nabla e_p^{i+1} ||^2 \leq \langle (\lambda_\Sigma(S_h^{n+1,i}) - \lambda_\Sigma(S_h^{n+1})) \nabla \overline{p}_h^{n+1}, \nabla e_p^{i+1} \rangle$$
$$+ \Big\langle \frac{\lambda_n(S_h^{n+1,i}) - \lambda_n(S_h^{n+1})}{2} \nabla p_c(S_h^{n+1}), \nabla e_p^{i+1} \Big\rangle + \Big\langle \frac{\lambda_w(S_h^{n+1,i}) - \lambda_w(S_h^{n+1})}{2} \nabla p_c(S_h^{n+1}), \nabla e_p^{i+1} \Big\rangle$$
$$+ \Big\langle \frac{\lambda_n(S_h^{n+1,i}) - \lambda_w(S_h^{n+1,i})}{2} \nabla \left(p_c(S_h^{n+1,i}) - p_c(S_h^{n+1})\right), \nabla e_p^{i+1} \Big\rangle.$$

As the functions λ_w, λ_n, λ_Σ, λ_{dif}, p_h^{n+1}, $\bigtriangledown p_c$ are Lipschitz continuous by (A1) we can use the Lipschitz inequality for them. From Lipschitz continuity it follows that these functions are also bounded in the domain Ω which means that we can replace them with the largest value of each function in the domain.

$$\begin{aligned}&\lambda_\Sigma^0 || \bigtriangledown e_p^{i+1}||^2 \leq L_{\lambda_\Sigma} M_p ||e_s^i|| || \bigtriangledown e_p^{i+1}|| + \frac{1}{2} L_{\lambda_n} M_{p_c} ||e_s^i|| || \bigtriangledown e_p^{i+1}|| \\ &+ \frac{1}{2} L_{\lambda_w} M_{p_c} ||e_s^i|| || \bigtriangledown e_p^{i+1}|| + \frac{1}{2} \lambda_{dif}^0 L_{p_c} ||e_s^i|| || \bigtriangledown e_p^{i+1}||.\end{aligned}$$

After additional algebraic manipulations, we can get the following estimation for the error of the pressure function:

$$|| \bigtriangledown e_p^{i+1}|| \leq \left(\frac{L_{\lambda_\Sigma} M_p}{\lambda_\Sigma^0} + \frac{L_{\lambda_n} + L_{\lambda_w}}{2\lambda_\Sigma^0} M_{p_c} + \frac{\lambda_{dif}^0 L_{p_c}}{2\lambda_\Sigma^0} \right) ||e_s^i||. \tag{3.43}$$

Let us return to the second equation.

$$\phi \langle e_s^{i+1}, \upsilon_h \rangle + \Delta t \langle \lambda_w(S_h^{n+1}) \bigtriangledown (p_h^{n+1} - p_h^{n+1,i+1}), \bigtriangledown \upsilon_h \rangle \tag{3.44}$$

$$+ \Delta t \langle (\lambda_w(S_h^{n+1}) - \lambda_w(S_h^{n+1,i})) \bigtriangledown \overline{p}_h^{n+1}, \bigtriangledown \upsilon_h \rangle \tag{3.45}$$

$$+ \frac{\Delta t}{2} \langle \lambda_w(S_h^{n+1,i}) \bigtriangledown p_c(S_h^{n+1,i+1}) - \lambda_w(S_h^{n+1}) \bigtriangledown p_c(S_h^{n+1}), \bigtriangledown \upsilon_h \rangle = 0. \tag{3.46}$$

We have to compute the gradient of the capillary pressure function, which depends on the unknown $S_h^{n+1,i+1}$. To do it, we use the following approximation

$$\bigtriangledown p_c(S_h^{n+1,i+1}) \sim p_c'(S_h^{n+1,i}) \bigtriangledown S_h^{n+1,i+1}.$$

By substituting this approximation in (3.44) we obtain:

$$\begin{aligned}&\phi \langle e_s^{i+1}, \upsilon_h \rangle + \Delta t \langle \lambda_w(S_h^{n+1}) \bigtriangledown e_p^{i+1}, \bigtriangledown \upsilon_h \rangle + \Delta t \langle (\lambda_w(S_h^{n+1}) - \lambda_w(S_h^{n+1,i})) \bigtriangledown \overline{p}_h^{n+1}, \bigtriangledown \upsilon_h \rangle \\ &- \frac{\Delta t}{2} \langle \lambda_w(S_h^{n+1,i}) p_c'(S_h^{n+1,i}) \bigtriangledown (S_h^{n+1} - S_h^{n+1,i+1}), \bigtriangledown \upsilon_h \rangle \\ &+ \frac{\Delta t}{2} \langle (\lambda_w(S_h^{n+1,i}) p_c'(S_h^{n+1,i}) - \lambda_w(S_h^{n+1}) p_c'(S_h^{n+1})) \bigtriangledown S_h^{n+1}, \bigtriangledown \upsilon_h \rangle = 0.\end{aligned}$$

Let us test the equation above with $\upsilon_h = e_s^{n+1}$. At this point we also use the assumption (A2). As the function p_c' is negative, its smallest value is also negative and we denote it as $\min(p_c') = -M_{p_c'}$, where $M_{p_c'} \geq 0$.

$$\begin{aligned}&\phi ||e_s^{i+1}||^2 + \frac{\Delta t}{2} M_{\lambda_w} M_{p_c'} || \bigtriangledown e_s^{i+1}||^2 \\ &\leq \Delta t M_{\lambda_w} || \bigtriangledown e_p^{i+1}|| || \bigtriangledown e_s^{i+1}|| + \Delta t L_{\lambda_w} M_p ||e_s^i|| || \bigtriangledown e_s^{i+1}|| \\ &+ \frac{\Delta t}{2} (M_{p_c'} L_{\lambda_w} + M_{\lambda_w} L_{p_c'}) M_S ||e_s^i|| || \bigtriangledown e_s^{i+1}||.\end{aligned}$$

Using the knowledge about the gradient of the pressure function error (3.43) we can estimate the error for the saturation function.

$$\phi||e_s^{i+1}||^2 + \frac{\Delta t}{2} M_{\lambda_w} M_{p_c'} || \bigtriangledown e_s^{i+1}||^2 \leq \Delta t M_{\lambda_w} \Big(\frac{L_{\lambda_\Sigma} M_p}{\lambda_\Sigma^0} + \frac{L_{\lambda_n} + L_{\lambda_w}}{2\lambda_\Sigma^0} M_{p_c}$$
$$+ \frac{\lambda_{dif}^0 L_{p_c}}{2\lambda_\Sigma^0} + L_{\lambda_w} M_p + \frac{1}{2}(M_{p_c'} L_{\lambda_w} + M_{\lambda_w} L_{p_c'}) M_S \Big) ||e_s^i|| || \bigtriangledown e_s^{i+1}||.$$

Let us say that

$$C(\Delta t) = \Delta t M_{\lambda_w} \left(\frac{L_{\lambda_\Sigma} M_p}{\lambda_\Sigma^0} + \frac{L_{\lambda_n} + L_{\lambda_w}}{2\lambda_\Sigma^0} M_{p_c} + \frac{\lambda_{dif}^0 L_{p_c}}{2\lambda_\Sigma^0} \right. \tag{3.47}$$
$$\left. + L_{\lambda_w} M_p + \frac{1}{2}(M_{p_c'} L_{\lambda_w} + M_{\lambda_w} L_{p_c'}) M_S \right), \tag{3.48}$$

where $C(\Delta t) \geq 0$.

Further we use Young's inequality

$$ab \leq \frac{a^2}{2\varepsilon} + \frac{\varepsilon b^2}{2} \quad \text{for all } \varepsilon > 0 \tag{3.49}$$

and Poincaré inequality

$$||u||_{L^2(\Omega)} \leq C|| \bigtriangledown u||_{L^2(\Omega)}. \tag{3.50}$$

The last holds for any $u \in H_0^1(\Omega)$. Using the inequalities we finally get the following estimate:

$$\phi||e_s^{i+1}||^2 + \left(\frac{\Delta t}{2} M_{\lambda_w} M_{p_c'} - \frac{C(\Delta t)}{2\varepsilon} \right) || \bigtriangledown e_s^{i+1}||^2 \leq \frac{C(\Delta t)\varepsilon}{2} ||e_s^i||^2,$$

$$\left(\phi + C_\Omega \left(\frac{\Delta t}{2} M_{\lambda_w} M_{p_c'} - \frac{C(\Delta t)}{2\varepsilon} \right) \right) ||e_s^{i+1}||^2 \leq \frac{C(\Delta t)\varepsilon}{2} ||e_s^i||^2,$$

which further implies

$$||e_s^{i+1}||^2 \leq \frac{C(\Delta t)\varepsilon}{2\left(\phi + C_\Omega \left(\frac{\Delta t}{2} M_{\lambda_w} M_{p_c'} - \frac{C(\Delta t)}{2\varepsilon} \right) \right)} ||e_s^i||^2.$$

This proves that our scheme (3.40)-(3.41) linearly converges under the following mild restriction on the time step:

$$\frac{C(\Delta t)\varepsilon}{2\left(\phi + C_\Omega \left(\frac{\Delta t}{2} M_{\lambda_w} M_{p_c'} - \frac{C(\Delta t)}{2\varepsilon} \right) \right)} \leq 1. \tag{3.51}$$

□

To verify the applicability of the scheme to realistic scenarios we test and compare it with the other schemes in the next Chapter.

Chapter 4

Numerical Results

In this chapter we present a verification of convergence of the new implicit scheme developed in Section 3.3.2. We look at a few numerical tests with different levels of complexity and study the convergence rate of the iterative process. We also compare the new scheme with IMPES and Newton's implementation of the implicit method (referred to as Newton's method for conciseness). In Section 4.2 we highlight the advantages of the new scheme by comparing it with IMPES and Newton's scheme in terms of the CPU time and stability with respect to time-step.

4.1 Verification of Convergence

In this section we demonstrate the convergence of the new iterative scheme applied to the model problem developed in Chapter 2:

$$\begin{aligned}
& -\bigtriangledown \cdot \boldsymbol{k}(\boldsymbol{\lambda_\Sigma} \bigtriangledown \overline{p} + \frac{\boldsymbol{\lambda_n} - \boldsymbol{\lambda_w}}{2} \bigtriangledown p_c) = F_{pr}, \\
& \phi \frac{\partial S_w}{\partial t} - \bigtriangledown \cdot \big(\boldsymbol{\lambda_w} \boldsymbol{k} \bigtriangledown (\overline{p} - \frac{1}{2} p_c)\big) = F_{sat}, \\
& S_w^0 = S_w(x, t_0), \quad \overline{p}^0 = \overline{p}(x, t_0), \\
& S_w|_{\partial\Omega} = S_w^\Gamma, \quad \overline{p}|_{\partial\Omega} = \overline{p}^\Gamma.
\end{aligned} \tag{4.1}$$

In all the following test cases we choose the right-hand side functions such that the analytical solution of the system above in the domain $\Omega = (0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1)$ is defined as follows:

$$\overline{p}(\boldsymbol{x}, t) = t x_1(1 - x_1) x_2(1 - x_2), \tag{4.2}$$

$$S(\boldsymbol{x}, t) = \frac{1}{2} + t x_1(1 - x_1) x_2(1 - x_2). \tag{4.3}$$

All computations are done on the time interval $t \in [0.0, 1.0]$.

4.1.1 Test case 1 ($\lambda_n = \lambda_w = 1$, $p_c = 0$)

In this test case we consider the system (4.1) where $\lambda_n = \lambda_w = \phi = \rho = 1$ and there is no capillary pressure $p_c = 0$. Thus, we get the following simplified system:

$$\begin{aligned} &-\bigtriangledown \cdot (2 \bigtriangledown \overline{p}) = F_{pr}, \\ &\frac{\partial S}{\partial t} - \bigtriangledown \cdot (\bigtriangledown \overline{p}) = F_{sat}, \\ &\overline{P_0} = 0, \quad \overline{p}_{\partial\Omega} = 0, \\ &S_0 = \frac{1}{2}, \quad S_{\partial\Omega} = \frac{1}{2}. \end{aligned} \tag{4.4}$$

Let us choose the right-hand side functions such that the analytical solution of the system (4.4) is provided by the formulas (4.2)-(4.3). For the analytical solution we present the following relations:

$$\triangle \overline{p} = -2t(x_1 - x_1^2 + x_2 - x_2^2), \tag{4.5}$$

$$\frac{\partial S}{\partial t} = (x_1 - x_1^2)(x_2 - x_2^2). \tag{4.6}$$

Then the right-hand side functions can be easily computed as:

$$\begin{aligned} F_{pr} &= -2\triangle \overline{p}, \\ F_{sat} &= \frac{\partial S}{\partial t} - \triangle \overline{p}. \end{aligned} \tag{4.7}$$

In Figure 4.1 we present the numerical solution, the true solution and the error (the difference between the solutions at each point) for the pressure function at time $T_{final} = 1.0, h = 0.05$ after 20 time steps with $dt = 0.05$. Figure 4.2 presents the same plots for the saturation function.

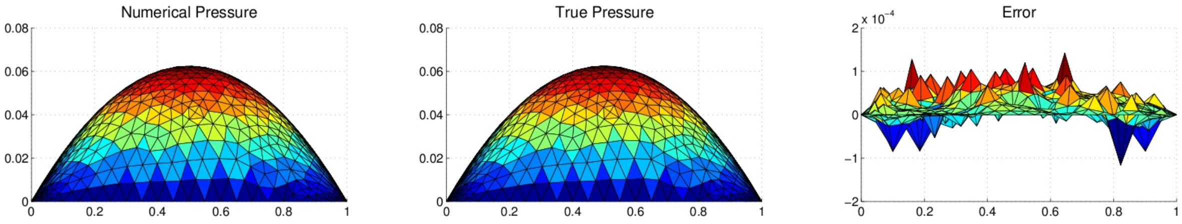

Figure 4.1: Test case 1. Numerical solution, true solution and error for the pressure function at time $T_{final} = 1.0, h = 0.05, dt = 0.05$.

To verify the correctness of the implementation we look at the convergence rates with respect to time and space discretization. Table 4.1 presents the results of our implicit scheme applied to the system (4.4) at different mesh and time-step sizes. The computations are performed on the time interval $t \in [0.0, 1.0]$, each consequent computation is done on the refined by the factor of two mesh and with twice smaller

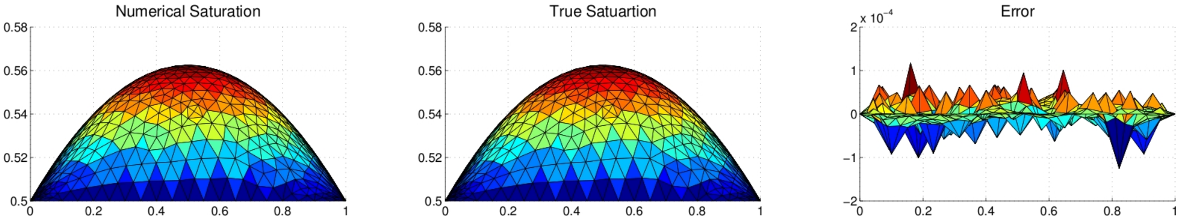

Figure 4.2: Test case 1. Numerical solution, true solution and error for the saturation function at time $T_{final} = 1.0, h = 0.05, dt = 0.05$.

	h	dt	Pressure Error	Saturation Error	E_i^p/E_{i+1}^p	E_i^s/E_{i+1}^s
1	0.1	0.1	1.3026e-04	0.0036		
2	0.05	0.05	2.9646e-05	7.6252e-04	4.3939	4.7737
3	2.5e-02	2.5e-02	7.4590e-06	1.9600e-04	3.9746	3.8903
4	1.25e-02	1.25e-02	1.8976e-06	5.4108e-05	3.9308	3.6224
5	6.25e-03	6.25e-03	4.7244e-07	1.5277e-05	4.0166	3.5418

Table 4.1: Test case 1. L^2-norm errors of pressure and saturation functions for different time steps and mesh sizes, $T_{final} = 1.0$.

time-step. The errors from Table 4.1 are visualized in Figure 4.3.

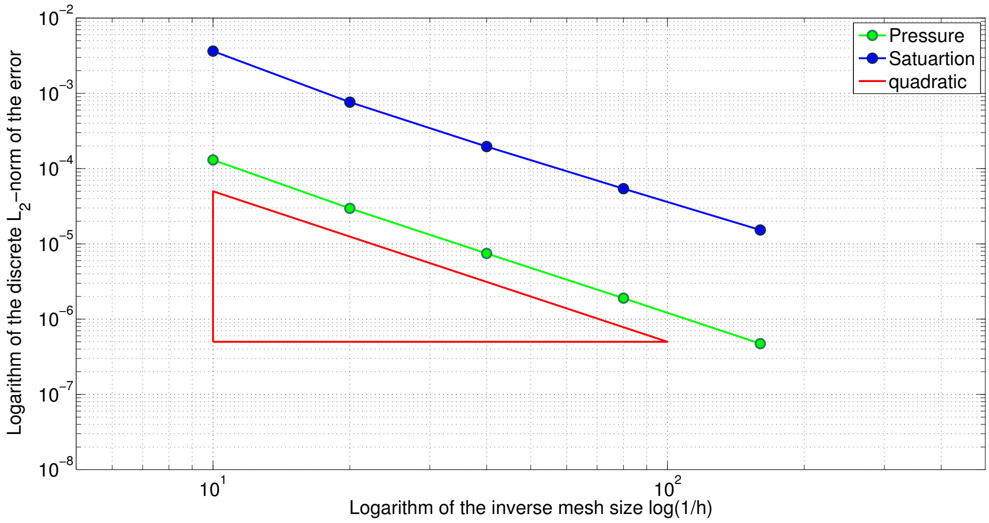

Figure 4.3: Test case 1. Discrete error L^2-norm for the solution of pressure (blue) and saturation (green) equations for different space and time steps. Both space and time steps are refined with the factor of two each time.

We used the finite element discretization in space and backward Euler in time. Then the expected order of the error is $O(dt + h^2)$ for the saturation equation and $O(h^2)$ for the pressure [5, 17]. From Table 4.1 we see that the error of the pressure equation became four times smaller as h decreased twice. This perfectly agrees with theory. For the saturation equation we see that with a decrease in dt and h the error also became nearly four times smaller. However, we also see a downward trend in the

	h	dt	Pressure Error	Saturation Error	E_i^p/E_{i+1}^p	E_i^s/E_{i+1}^s
1	0.1	0.1	1.1842e-04	2.9392e-04		
2	0.05	0.05	2.6503e-05	1.2223e-04	4.4682	2.4047
3	2.5e-02	2.5e-02	6.6447e-06	6.3125e-05	3.9886	1.9363
4	1.25e-02	1.25e-02	1.7059e-06	3.3779e-05	3.8952	1.8687
5	6.25e-03	6.25e-03	4.1680e-07	1.6321e-05	4.0928	2.0697

Table 4.2: Test case 2. L^2-norm errors of the pressure and the saturation functions for different time steps and mesh sizes, $T_{final} = 1.0$.

last column in Table 4.1. Thus, we can expect the order $O(dt + h^2)$ as $dt, h \to 0$.

4.1.2 Test case 2 ($\lambda_n = \lambda_w = 1$, $p_c = 1$)

In this test case we consider the system (4.1) with $\lambda_n = \lambda_w = \phi = \rho = 1$. However this time we have a constant capillary pressure $p_c = 1$. Thus, we get the following simplified system:

$$\begin{aligned}
&-\bigtriangledown \cdot (2 \bigtriangledown \overline{p}) = F_{pr},\\
&\frac{\partial S}{\partial t} - \bigtriangledown \cdot \bigtriangledown(\overline{p} - \frac{1}{2}p_c) = F_{sat},\\
&\overline{P_0} = 0, \quad \overline{p}_{\partial\Omega} = 0,\\
&S_0 = \frac{1}{2}, \quad S_{\partial\Omega} = \frac{1}{2}.
\end{aligned} \tag{4.8}$$

As before, we choose the right-hand side functions such that the analytical solution of the system above in the domain Ω is still the same (4.2) - (4.3). Then the right-hand side functions can be easily computed:

$$\begin{aligned}
F_{pr} &= -2\triangle\overline{p},\\
F_{sat} &= \frac{\partial S}{\partial t} - \triangle\overline{p}.
\end{aligned} \tag{4.9}$$

Let us look at how the error changes at different time and space steps, like we did in the previous test case. Table 4.2 shows that the error decreasing ratio for the saturation equation differs from the previous test case and it tends to the value of two. This indicates that the error is dominated by the Euler approximation for the time derivative. In Figure 4.4 the errors from Table 4.2 are plotted, so that it is easier to see that in this test case we have convergence of order two for the pressure equation and only first order convergence for the saturation equation.

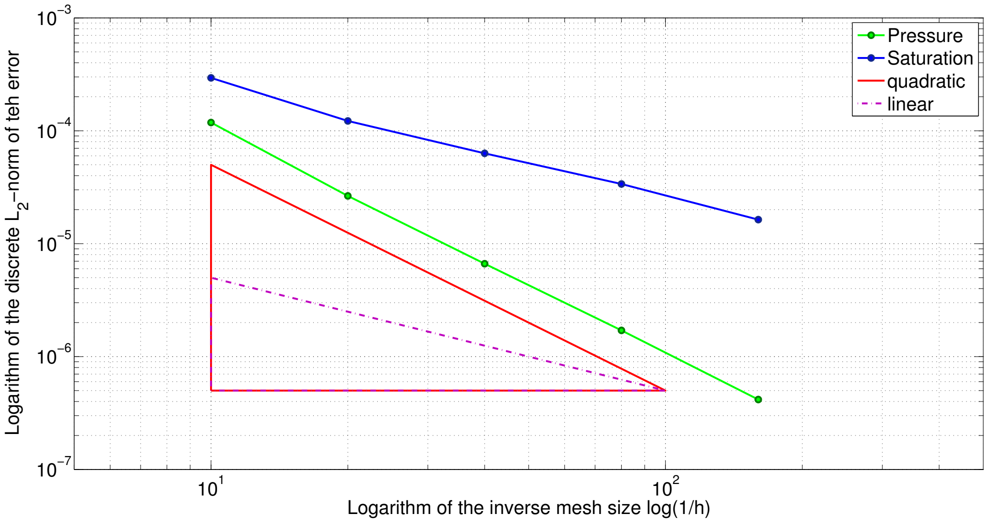

Figure 4.4: Test case 2. Discrete error L^2-norm for the solution of the pressure (blue) and the saturation (green) equations for different space and time steps. Both space and time steps are decreased twice for every next simulation.

4.1.3 Test case 3 ($\lambda_n = \frac{1}{4}$, $\lambda_w = \frac{3}{4}$, $p_c = 1 - S_w^2$)

In this test case we simplify the system (4.1) in sense of parameters $\lambda_n = \frac{1}{4}, \lambda_w = \frac{3}{4}, \phi = \rho = 1$. However, this time we include the capillary pressure as smooth Lipschitz continuous function of the saturation $p_c = 1 - S_w^2$. Then our new system looks like:

$$\begin{aligned} &-\bigtriangledown \cdot (\bigtriangledown \overline{p} + \frac{1}{4} \bigtriangledown p_c) = F_{pr}, \\ &\frac{\partial S}{\partial t} - \bigtriangledown \cdot (\frac{3}{4} \bigtriangledown (\overline{p} - \frac{1}{2} p_c)) = F_{sat}, \\ &\overline{P_0} = 0, \quad \overline{p}_{\partial\Omega} = 0, \\ &S_0 = \frac{1}{2}, \quad S_{\partial\Omega} = \frac{1}{2}. \end{aligned} \tag{4.10}$$

We want the analytical solution of the system to maintain the same form (4.2)-(4.3). Let us first compute the Laplacian of the capillary pressure function:

$$\triangle p_c = 2t(x_2 - x_2^2 + x_1 - x_1^2) - 2t^2((x_2 - x_2^2)^2(1 - 6x_1 + 6x_1^2) + (x_1 - x_1^2)^2(1 - 6x_2 + 6x_2^2)). \tag{4.11}$$

Then the right-hand side functions should be chosen as:

$$F_{pr} = -\triangle\overline{p} + \frac{1}{4}\triangle p_c, \quad F_{sat} = \frac{\partial S}{\partial t} - \frac{3}{4}\triangle\overline{p} + \frac{3}{8}\triangle p_c, \tag{4.12}$$

where $\triangle\overline{p}$ and $\frac{\partial S}{\partial t}$ are as in the test case 1 (4.5)-(4.6).

	h	dt	Pressure Error	Saturation Error	E_i^p/E_{i+1}^p	E_i^s/E_{i+1}^s
1	0.1	0.1	2.2958e-04	0.0012		
2	0.05	0.05	4.9383e-05	2.6342e-04	4.6491	4.4626
3	2.5e-02	2.5e-02	1.2223e-05	6.5993e-05	4.0401	3.9917
4	1.25e-02	1.25e-02	3.0972e-06	1.6763e-05	3.9465	3.9367
5	6.25e-03	6.25e-03	7.6654e-07	4.1586e-06	4.0405	4.0309

Table 4.3: Test case 3. L^2-norm errors of pressure and saturation functions for different time steps and mesh sizes, $T_{final} = 1.0$.

As before, we examine the convergence of the scheme by comparing the errors for different discretizations. The results are presented in Table (4.3) and Figure (4.5).

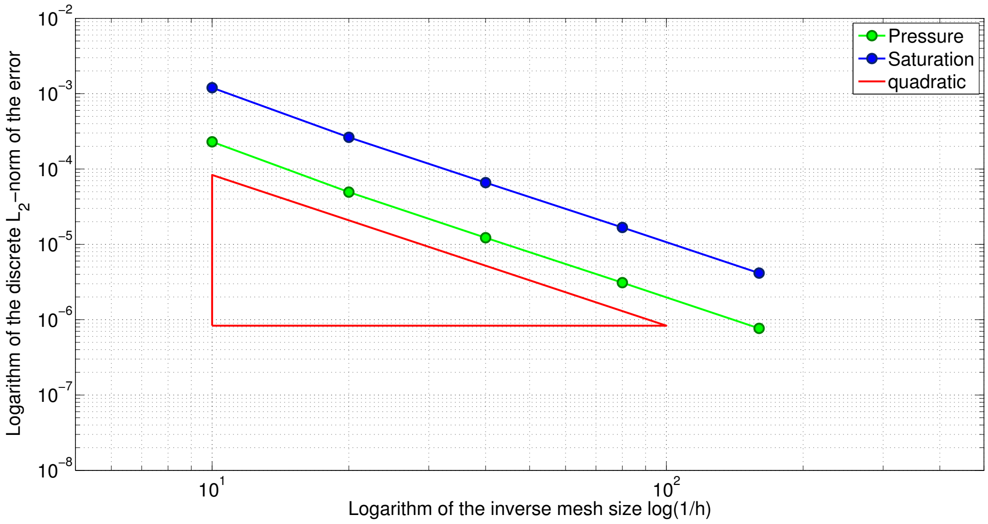

Figure 4.5: Test case 3. Discrete error L^2-norm for the solution of the pressure (blue) and the saturation (green) equations for different space and time steps. Both space and time steps are decreased twice for every next simulation.

In this test case we also examine the errors at each iteration of the scheme. In Section 3.3.3 we proved that our scheme converges at least linearly and here we want to study the real error change and analyze whether it agrees with the theoretical proof. We did not consider the errors in the previous test cases as due to the parameterization choice the scheme converged too quickly and we were not able to catch the real behaviour of the error.

Figures 4.6 and 4.7 show the difference between numerical solutions at each iteration $||p^{i+1} - p^i||_{L_2}$ and $||S^{i+1} - S^i||_{L_2}$ during one time step. The stopping criteria for the iteration is $||p^{i+1}-p^i||_{L_2} \leq \varepsilon$ and $||S^{i+1}-S^i||_{L_2} \leq \varepsilon$, where $\varepsilon = 10^{-9}$, therefore, the lines does not go down further than the level of 10^{-9}. Figures 4.8 - 4.9 presents the error of the numerical solution at each iteration with the analytical solution, $||p^{i+1}-p^{an}||_{L_2}$ and $||S^{i+1}-S^{an}||_{L_2}$. First of all, we see that for this test case the scheme shows convergence faster than linear. Second, in Figures 4.8 - 4.9 the error first goes down quickly and then stabilizes at some level. This level shows how big numerical error we have for this particular mesh and time-step. With the change of mesh and time-step size this numerical error becomes smaller, which also agrees with the numbers in Table 4.3.

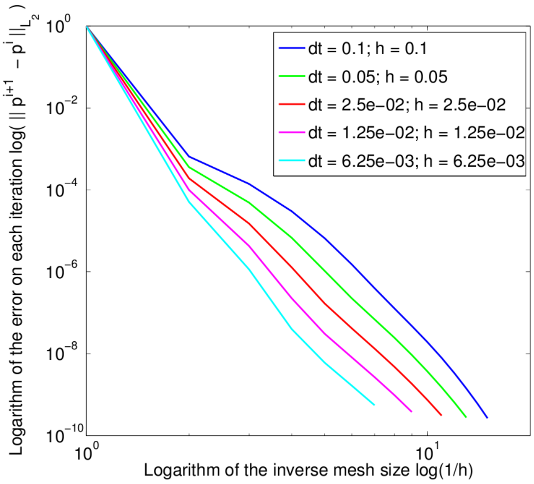

Figure 4.6: The difference between numerical solutions at each iteration of one time-step $||p^{i+1} - p^i||_{L_2}$ for various mesh and time-step sizes.

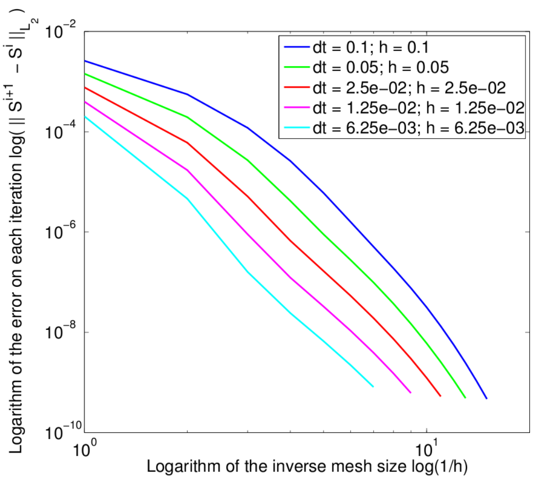

Figure 4.7: The difference between numerical solutions at each iteration of one time-step $||S^{i+1} - S^i||_{L_2}$ for various mesh and time-step sizes.

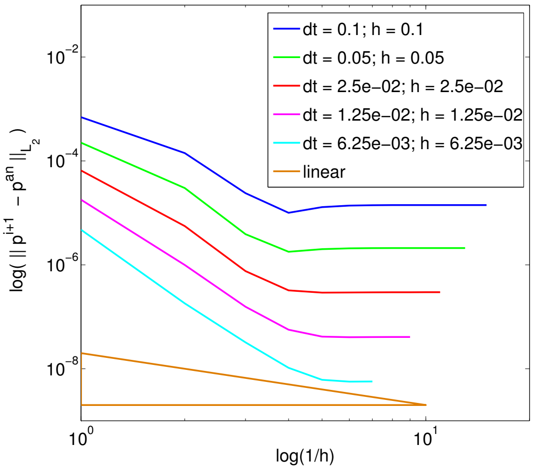

Figure 4.8: The difference between numerical and analytical solutions at each iteration of one time-step $||p^{i+1} - p^{an}||_{L_2}$ for various mesh and time-step sizes.

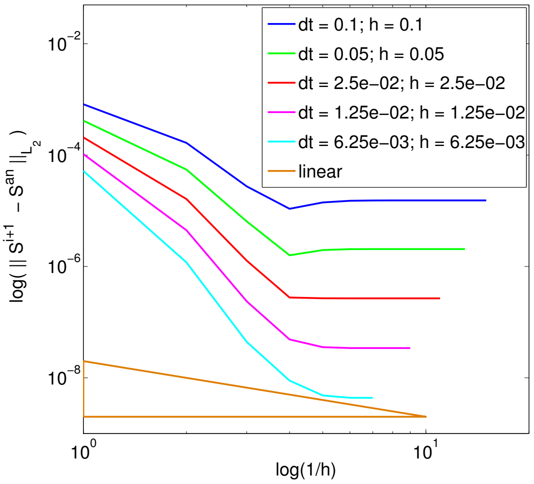

Figure 4.9: The difference between numerical and analytical solutions at each iteration of one time-step $||S^{i+1} - S^{an}||_{L_2}$ for various mesh and time-step sizes.

Let us also take a look at the number of iterations that the new implicit scheme requires at each time step. Figure (4.10) presents this number at different mesh sizes with fixed time step. The plots indicate that the number of iterations does not depend on the mesh size at all and changes with the size of time step only. However, the parametrization used here is relatively simple and in the next test case we will look at the number of iterations again.

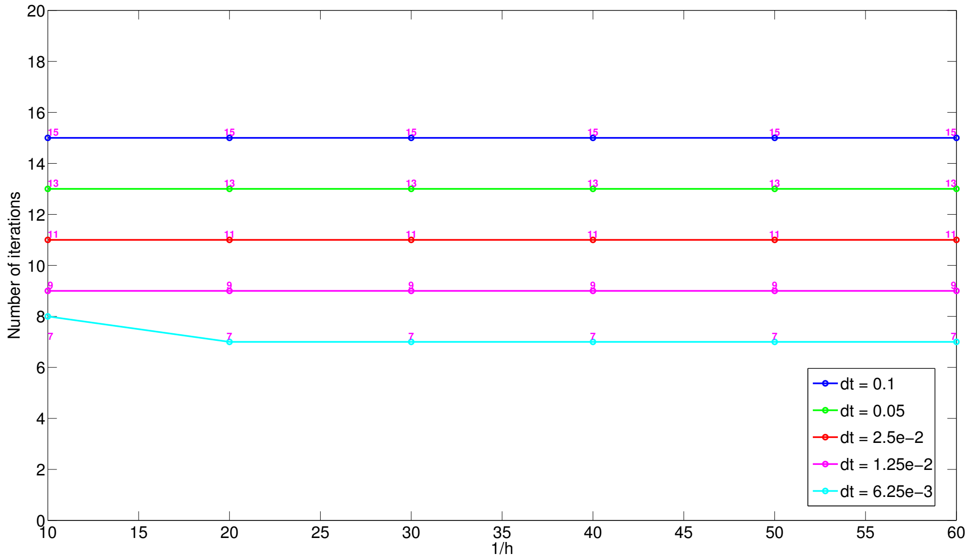

Figure 4.10: Numbers of iterations for several mesh and step sizes.

4.1.4 Test case 4 (λ_n, λ_w from van Genuchten parametrization, $p_c = 1 - S_w^2$)

Now we use van Genuchten parametrization, described in detailed in the appendix A, 5, for the phase mobility functions $\lambda_n = \frac{\boldsymbol{k}_{r,n}}{\mu_n}$, $\lambda_w = \frac{\boldsymbol{k}_{r,w}}{\mu_w}$, while the capillary pressure stays the same: $p_c = 1-S_w^2$. The values of other parameters are $\boldsymbol{k} = 1, \phi = 1, \mu_n = 1, \mu_w = 1$. Then our new system is written as follows:

$$\begin{aligned}
&-\bigtriangledown \cdot (\lambda_\Sigma \bigtriangledown \overline{p} + \frac{\lambda_{dif}}{2} \bigtriangledown p_c) = F_{pr},\\
&\frac{\partial S}{\partial t} - \bigtriangledown \cdot (\lambda_w \bigtriangledown (\overline{p} - \frac{1}{2} p_c)) = F_{sat},\\
&\overline{p_0} = 0, \qquad \overline{p}_{\partial\Omega} = 0,\\
&S_0 = \frac{1}{2}, \qquad S_{\partial\Omega} = \frac{1}{2},\\
&\lambda_\Sigma = \lambda_n + \lambda_w,\\
&\lambda_{dif} = \lambda_n - \lambda_w.
\end{aligned} \tag{4.13}$$

We choose F_{pr} and F_{sat} such that the analytical solution (4.2)-(4.3) stays the same:

$$\begin{aligned}
F_{pr} &= -\bigtriangledown \lambda_\Sigma \cdot \bigtriangledown \overline{p} - \lambda_\Sigma \triangle \overline{p} - \frac{1}{2} \bigtriangledown \lambda_{dif} \cdot \bigtriangledown p_c - \frac{1}{2} \lambda_{dif} \triangle p_c,\\
F_{sat} &= \frac{\partial S}{\partial t} - \bigtriangledown \lambda_w \cdot \bigtriangledown \overline{p} - \lambda_w \triangle \overline{p} + \frac{1}{2} \bigtriangledown \lambda_w \cdot \bigtriangledown p_c + \frac{1}{2} \lambda_w \triangle p_c,
\end{aligned} \tag{4.14}$$

	h	dt	Pressure Error	Saturation Error	E_i^p/E_{i+1}^p	E_i^s/E_{i+1}^s
1	0.1	0.1	9.9101e-05	1.5739e-04		
2	0.05	0.05	2.2000e-05	3.8927e-05	4.5046	4.0433
3	2.5e-02	2.5e-02	5.5877e-06	9.9906e-06	3.9372	3.8964
4	1.25e-02	1.25e-02	1.4325e-06	2.5480e-06	3.9007	3.9210
5	6.25e-03	6.25e-03	3.5866e-07	6.3823e-07	3.9940	3.9923

Table 4.4: Test case 4. L^2-norm errors of pressure and saturation functions for different time steps and mesh sizes, $T_{final} = 1.0$.

where $\triangle\overline{p}$, $\dfrac{\partial S}{\partial t}$ and $\triangledown p_c$ are the same as before. The gradients are computed as:

$$\triangledown\, \overline{p} = \begin{pmatrix} t(1-2x_1)(x_2-x_2^2) \\ t(1-2x_2)(x_1-x_1^2) \end{pmatrix}, \tag{4.15}$$

$$\triangledown\, p_c = \begin{pmatrix} -\,2t(1-2x_1)(x_2-x_2^2)S \\ -\,2t(1-2x_2)(x_1-x_1^2)S \end{pmatrix}, \tag{4.16}$$

$$\begin{aligned} \frac{\partial\lambda_n}{\partial x_1} &= (tx_1x_2(1-x_2) - tx_2(1-x_1)(1-x_2))(\frac{1-S^2}{2\sqrt{1-S}} + 2S\sqrt{1-S}), \\ \frac{\partial\lambda_n}{\partial x_2} &= (tx_1x_2(1-x_1) - tx_1(1-x_1)(1-x_2))(\frac{1-S^2}{2\sqrt{1-S}} + 2S\sqrt{1-S}), \end{aligned} \tag{4.17}$$

$$\begin{aligned} \frac{\partial\lambda_w}{\partial x_1} &= (tx_2(1-x_1)(1-x_2) - tx_1x_2(1-x_2))(\frac{2S^{3/2}(1-\sqrt{1-S^2})}{\sqrt{1-S^2}} + \frac{(1-\sqrt{1-S^2})^2)}{2\sqrt{S}}), \\ \frac{\partial\lambda_w}{\partial x_2} &= (tx_1(1-x_1)(1-x_2) - tx_1x_2(1-x_1))(\frac{2S^{3/2}(1-\sqrt{1-S^2})}{\sqrt{1-S^2}} + \frac{(1-\sqrt{1-S^2})^2)}{2\sqrt{S}}), \\ \triangledown\, \lambda_n &= \left(\frac{\partial\lambda_n}{\partial x_1}, \frac{\partial\lambda_n}{\partial x_2}\right)^T, \quad \triangledown\lambda_w = \left(\frac{\partial\lambda_w}{\partial x_1}, \frac{\partial\lambda_w}{\partial x_2}\right)^T. \end{aligned} \tag{4.18}$$

Table (4.4) shows how the error changes with the refinement of the mesh by the factor of two and a simultaneous decrease in the time step. As in the previous test case, we observe the second order of convergence. Figure 4.11 presents the errors from Table 4.4.

Let us look at the number of iterations for this test case and compare it with what we get in the test case 4.1.3. Here, the phase mobility function is not constant anymore and becomes a function of the water saturation. For this test case we see that the number of iterations vary a little with the change of space step, but these changes are relatively small and the number tends to stabilize at some level. Thus, we can say that the new scheme shows independence of the mesh size in terms of needed number of iterations.

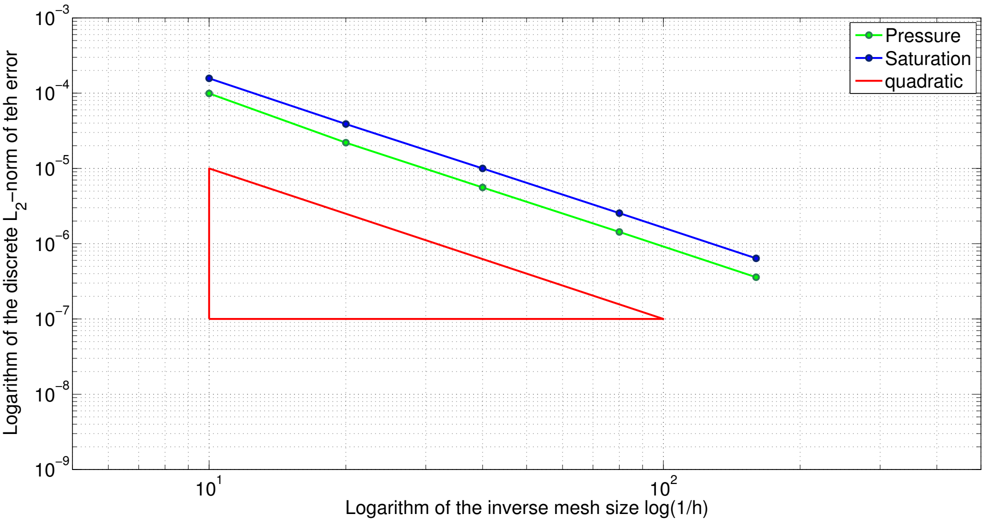

Figure 4.11: Test case 4. Discrete error L^2-norm for the solution of the pressure (blue) and the saturation (green) equations for different space and time steps. Both space and time steps are decreased twice for every next simulation.

4.1.5 Test case 5 ($\lambda_n, \lambda_w, p_c$ from van Genuchten parametrization)

In this test case we use van Genuchten parametrization not only for the phase mobility functions but also for the capillary pressure, see Appendix 5. The values of other parameters are $\boldsymbol{k} = 1, \phi = 1, \mu_n = 1, \mu_w = 1$. The system of equations is the same as in the previous test case. We also want the analytical solution (4.2-4.3) to stay the same. The formulas for the right-hand side functions F_{pr} and F_{sat} are the same as in the previous test case (4.14), however, the gradient and the Laplacian of the capillary pressure function are different.

$$\bigtriangledown p_c = \begin{pmatrix} \dfrac{-2(t(1-x_1)(1-x_2)x_2 - tx_1(1-x_2)x_2)}{s^3\sqrt{1/s^2-1}} \\ \dfrac{-2(t(1-x_1)x_1(1-x_2) - t(1-x_1)x_1x_2)}{s^3\sqrt{1/s^2-1}} \end{pmatrix}, \tag{4.19}$$

$$\begin{aligned} \triangle p_c = & \frac{6(t(1-x_1)(1-x_2)x_2 - tx_1(1-x_2)x_2)^2}{s^4\sqrt{1/s^2-1}} - \frac{2(t(1-x_1)(1-x_2)x_2 - tx_1(1-x_2)x_2)^2}{s^6(1/s^2-1)^{3/2}} \\ & + \frac{4t(1-x_2)x_2}{s^3\sqrt{1/s^2-1}} + \frac{6(t(1-x_1)x_1(1-x_2) - t(1-x_1)x_1x_2)^2}{s^4\sqrt{1/s^2-1}} \\ & - \frac{2(t(1-x_1)x_1(1-x_2) - t(1-x_1)x_1x_2)^2}{s^6(1/s^2-1)^{3/2}} + \frac{4t(1-x_1)x_1}{s^3\sqrt{1/s^2-1}}. \end{aligned} \tag{4.20}$$

We compute the errors on different time and space steps and examine the error

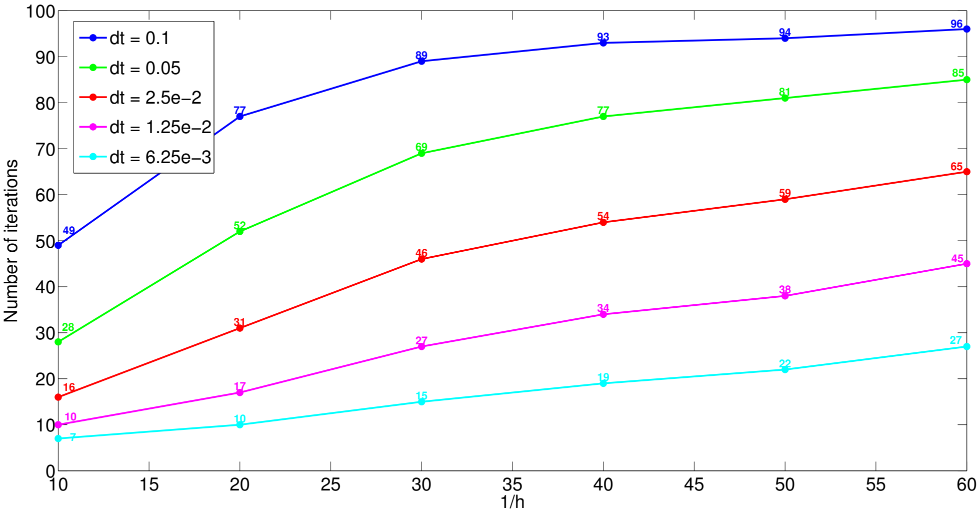

Figure 4.12: Numbers of iterations for several mesh and step sizes when van Genuchten parametrization for λ_n and λ_w is used.

	h	dt	Pressure Error	Saturation Error	E_i^p/E_{i+1}^p	E_i^s/E_{i+1}^s
1	0.1	0.1	1.2577e-04	9.3802e-05		
2	0.05	0.05	3.7186e-05	2.2608e-05	3.3823	4.1491
3	2.5e-02	2.5e-02	1.0848e-05	5.9607e-06	3.4280	3.7928
4	1.25e-02	1.25e-02	2.7961e-06	1.5611e-06	3.8796	3.8183

Table 4.5: Test case 5. L^2-norm errors of pressure and saturation functions for different time steps and mesh sizes, $T_{final} = 1.0$.

change. The results are presented in Table (4.5) and in Figure (4.13)

Let us look at the difference between numerical solutions at each iteration $||p^{i+1} - p^i||_{L_2}$ and $||S^{i+1} - S^i||_{L_2}$. As in the test case 3, the stopping criterion for the iteration is $||p^{i+1} - p^i||_{L_2} \leq \varepsilon$ and $||S^{i+1} - S^i||_{L_2} \leq \varepsilon$, where $\varepsilon = 10^{-9}$, that is why the lines does not go down further then the level of 10^{-9}. The corresponding errors are presented in Figures 4.14 and 4.15. Figures 4.16 - 4.17 show the difference between the numerical solution and the analytical solution at each iteration, $||p^{i+1} - p^{an}||_{L_2}$ and $||S^{i+1} - S^{an}||_{L_2}$. As for the test case 4.1.3, we observe convergence which is faster than linear. The error does not decrease beyond some level which corresponds to mismatch between analytical and numerical solutions for the particular discretization. All the errors are computed for just one time-step which is different for each line and on meshes with different levels of refinement.

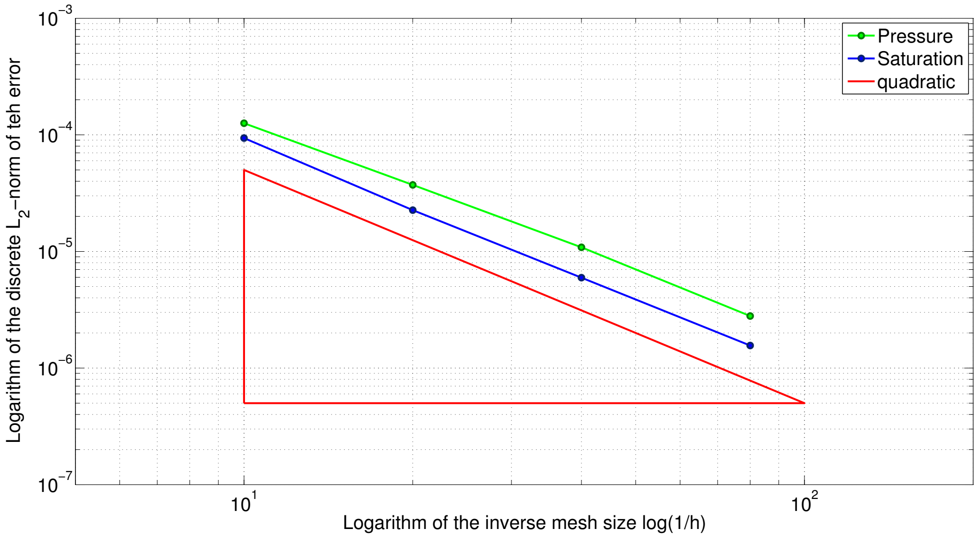

Figure 4.13: Test case 5. Discrete error L^2-norm for the solution of the pressure (blue) and the saturation (green) equations for different space and time steps. Both space and time steps are decreased twice for every next simulation.

4.2 Comparison of the stabilized iterative approach with IMPES and Newton's scheme

In Chapter 3 we presented a new stabilized iterative implementation of the fully implicit scheme for two-phase flow in porous media. In the previous section we studied different test cases to prove its convergence. In this section we compare the new scheme with our implementation of IMPES and Newtons's solver for the fully implicit scheme.

4.2.1 Robustness

First of all, we look at the improvement of the size of time-step compared with IMPES. In the IMPES scheme the saturation equation is solved explicitly. While it simplifies implementation and reduces the computational time, it results in the condition on a time step which is common to all explicit schemes. This problem was studied in several works, for example the authors of [10] examined the stability of IMPES and derived a criterion on the time-step for multidimensional three-phase flow. It is clear that IMPES provides accurate and stable solutions only if the time step is relatively small. Therefore we introduced an implicit scheme which to certain degree maintains the simplicity of IMPES, while having better convergence properties. In Table 4.6 we see the improvement in a time step of the new iteration scheme comparing to IMPES for the mesh size $h = 0.1$. We also included Newton's scheme here. Table 4.7 present the same comparison but for the mesh size $h = 0.05$. The different mesh size does not influence the two implicit schemes, but requires a smaller time step for IMPES.

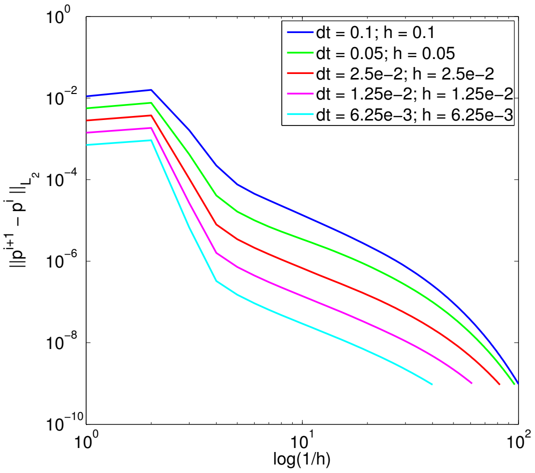

Figure 4.14: The difference between numerical solutions at each iteration of one time-step $||p^{i+1} - p^i||_{L_2}$ for various mesh and time-step sizes.

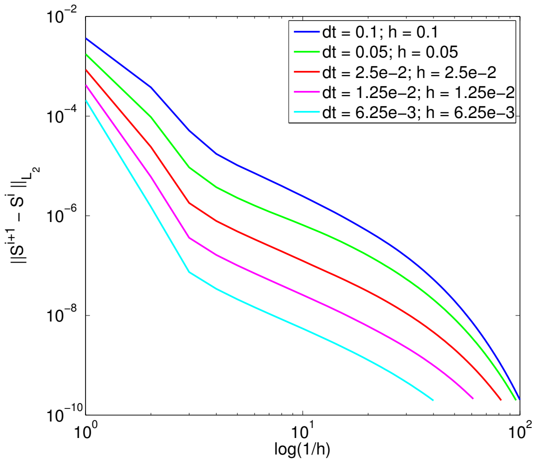

Figure 4.15: The difference between numerical solutions at each iteration of one time-step $||S^{i+1} - S^i||_{L_2}$ for various mesh and time-step sizes.

Numerical method	dt = 0.2	dt = 0.1	dt = 1e-2	dt = 1e-3	dt = 1e-4	dt = 1e-5
IMPES	No	No	No	Yes	Yes	Yes
Implicit Scheme	No	Yes	Yes	Yes	Yes	Yes
Newton's Scheme	Yes	Yes	Yes	Yes	Yes	Yes

Table 4.6: Comparison of IMPES and the implicit scheme in terms of convergence with different time steps, $T_{final} = 1.0, h = 0.1$.

4.2.2 CPU time

Let us now compare the schemes in terms of the required CPU time. In Figure 4.18 the CPU times of the schemes are presented. All computations are done in Matlab with the processor Intel(R) Core(TM)2 Duo E6850 @ 3.00GHz. The numerical solution, the true solution and the error for the last space discretization $h = 0.02$ for the new scheme are presented in Figures 4.19 - 4.20. The time step for IMPES was chosen so that the scheme converges for the smallest mesh size. While it is not optimal for all the simulations, an optimal choice of the step size is a complicated problem in itself [10, 11] and was not in the scope of the thesis.

Numerical method	dt = 0.2	dt = 0.1	dt = 1e-2	dt = 1e-3	dt = 1e-4	dt = 1e-5
IMPES	No	No	No	No	Yes	Yes
Implicit Scheme	No	Yes	Yes	Yes	Yes	Yes
Newton's Scheme	Yes	Yes	Yes	Yes	Yes	Yes

Table 4.7: Comparison of IMPES and the implicit scheme in terms of convergence with different time steps, $T_{final} = 1.0, h = 0.05$.

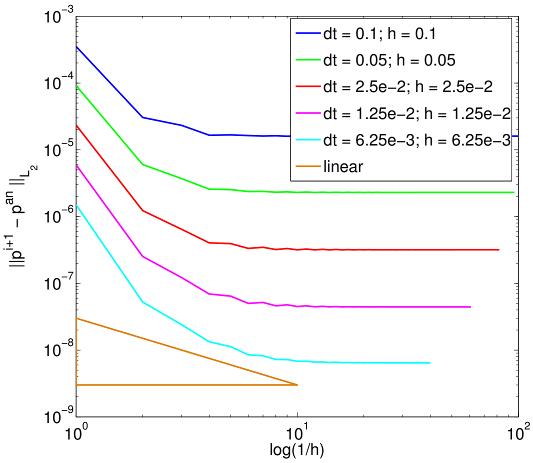

Figure 4.16: The difference between numerical and analytical solutions at each iteration of one time-step $||p^{i+1} - p^{an}||_{L_2}$ for various mesh and time-step sizes.

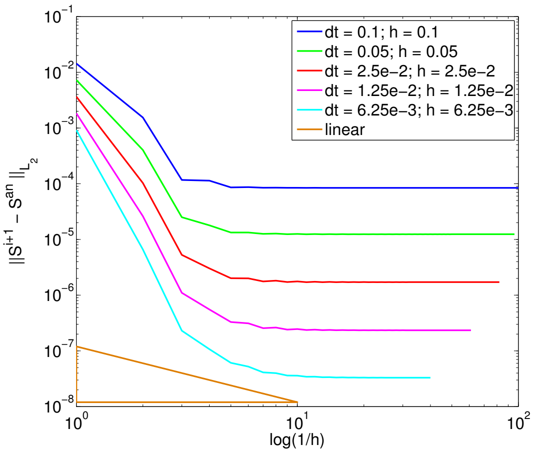

Figure 4.17: The difference between numerical and analytical solutions at each iteration of one time-step $||S^{i+1} - S^{an}||_{L_2}$ for various mesh and time-step sizes.

It is not surprising that with the implicit scheme we get an improvement in the CPU time. Despite the fact that the implicit scheme needs to complete a few iterations at each time step, the improvement in the size of the time step made it more efficient compare to the IMPES scheme. Even more interesting is the fact that we have a smaller computational time even comparing with Newton's method that needs fewer iteration due to the second order of convergence. In order to understand why we have better time performance of the new scheme comparing with Newton's iteration scheme we devote next section to examination of the condition number.

4.2.3 Condition Number

In terms of computational complexity, an important indicator is the condition number of the matrices of the corresponding linear systems at each iteration of the method. As we use finite elements for the space discretization in all schemes, it results in solving a similar system of linear equations at each iteration. An important characteristic of numerical schemes for linear systems is the condition number of the left-hand side matrices. In case of Newton's scheme we calculate the condition number of the Jacobian matrix. The condition number estimates for both schemes were computed using the MATLAB function *condest()* and are presented in Figure (4.21). The numbers for the implicit scheme are averaged over all iterations. In Newton's method we have one matrix for the whole implicit system and in new implicit scheme two (one for the pressure and one for the saturation equations). Therefore we plotted the same numbers for Newton's method and compared them with the numbers for the pressure and the saturation matrices separately.

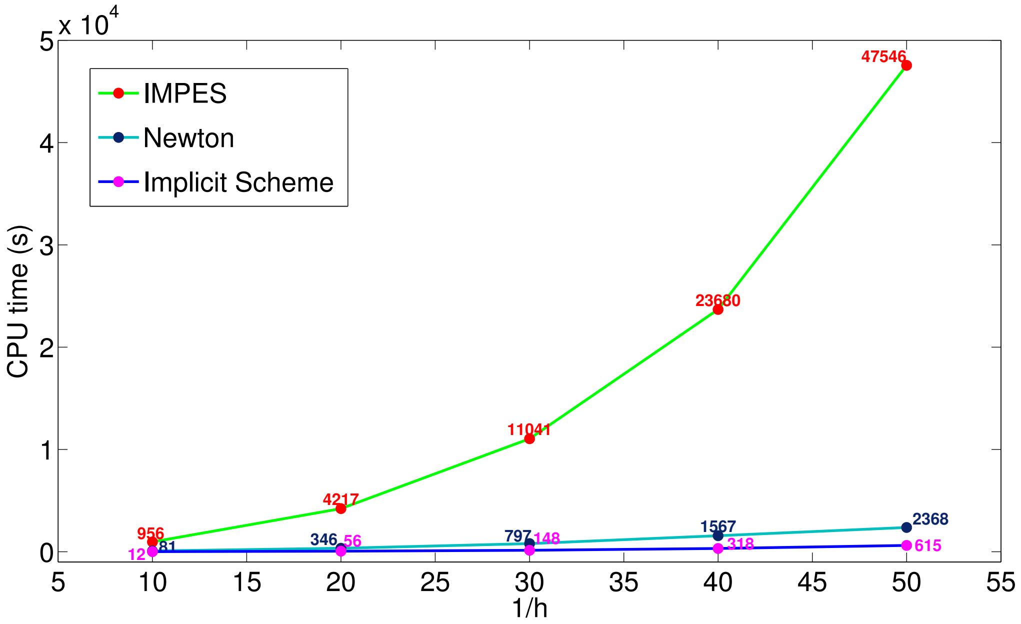

Figure 4.18: CPU time of the implicit scheme, IMPES and Newton's scheme for several mesh sizes, $dt_{implicit} = 0.1, dt_{Newton} = 0.1, dt_{IMPES} = 10^{-4}, T_{final} = 1.0$, parameterization as in Section 4.1.3.

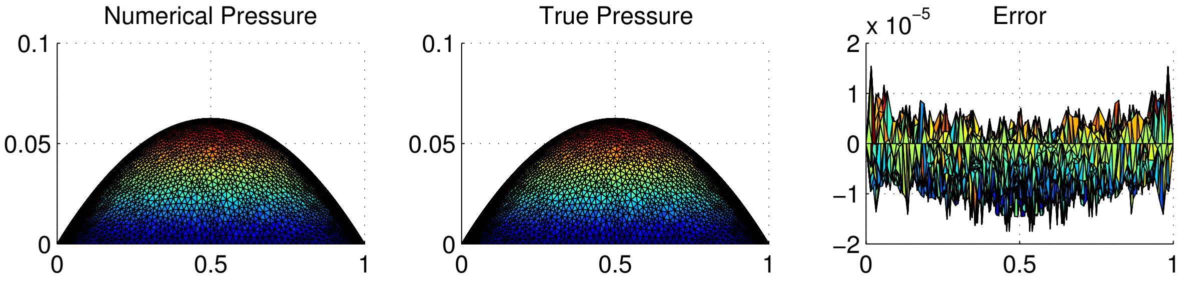

Figure 4.19: The numerical solution, the true solution and the error for the pressure equation at time $T_{final} = 1.0, h = 0.02, dt = 0.1$.

Figure (4.21) shows that the condition number for Newton's method is much higher then the condition number of our implicit scheme. This can explain why Newton's method is slower then the new iterative scheme. In spite of the fact that Newton's method needs fewer iterations to complete one time step, each iteration takes more time because the Jacobian matrix is relatively massive and it takes quite a long time to compute it. Also, as the condition number of the Jacobian matrix is relatively large, it takes more time for the linear solver to reach the needed accuracy for each iteration.

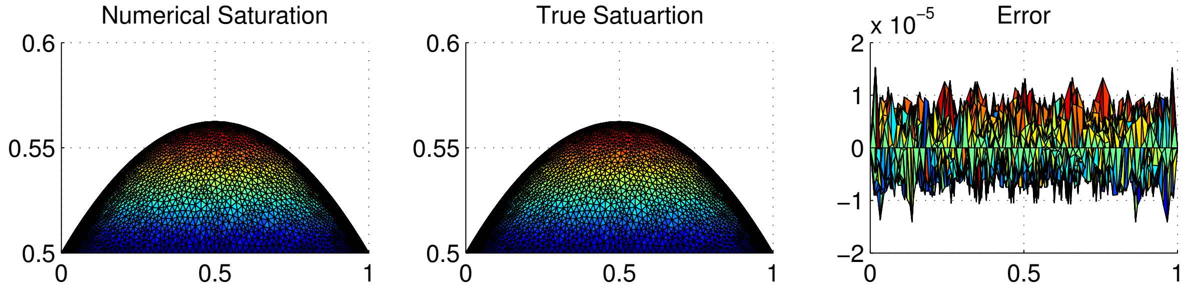

Figure 4.20: The numerical solution, the true solution and the error for the saturation equation at time $T_{final} = 1.0, h = 0.02, dt = 0.1$.

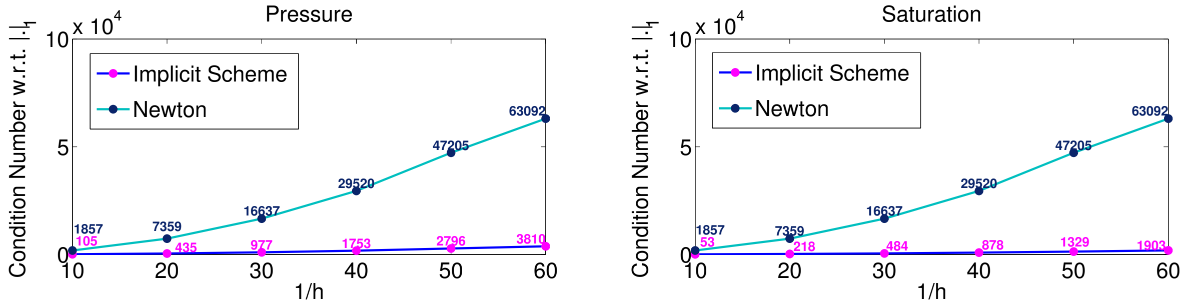

Figure 4.21: Condition number for several mesh sizes.

Chapter 5

Conclusion

In this thesis we presented a new implicit iterative scheme for the two-phase flow model in the averaged pressure formulation. Its main feature is the implicit treatment of the capillary pressure function which makes the scheme stable. This is accomplished by the linear approximation for the capillary pressure gradient which involves the saturation function on both current and previous iterations. Under these assumptions we proved the convergence theorem.

Our numerical experiments contain comparison of the new scheme with the two most often used methods for the two-phase flow problem: IMPES and the fully implicit scheme with Newton's method as a tool for solving the arising nonlinear system. IMPES and the new scheme share the idea of exploiting the structure of the equation. However, the standard IMPES scheme converges only with relatively small time-steps. The new scheme has only a mild condition on the time step size and for the mesh size $h = 0.05$ shows convergence with a three order of magnitude greater time step. Therefore we saw a huge improvement in the CPU time of the new scheme compared with IMPES.

We also compared the new scheme with the fully implicit scheme that uses Newton's method to deal with nonlinearities. Newton's method is widely used for solving nonlinear systems of equations because it can handle general problems and has the second order of convergence. However, comparing with our new scheme it still used more CPU time even though it needed a fewer number of iterations at each time step. This can be explained by the necessity of computing the Jacobian matrix at each iteration, as well as the worse condition number of the resulting Jacobian matrices.

All this makes our new scheme an attractive alternative to the existing methods. Nevertheless, the scheme's advantages comes from its limitations: it uses the specific structure of the problem. While the scheme cannot be used for more complex systems, the methodologies described in this thesis can be applied to extend the scheme to other applications, such that compressible fluid flow and three-phase flow models, other types of problem formulations or for the models where the hysteresis is included in the capillary pressure function.

Appendices

A Van Genuchten Parametrization

In this thesis we use the van Genuchten parametrization [32] for the dependence of the relative permeability and capillary pressure functions on the fluid saturation:

$$
\begin{aligned}
k_{r,n}(S) &= \sqrt{1-S}[1-S^{1/m}]^{2m},\\
k_{r,w} &= \sqrt{S}[1-(1-S^{1/m})^{m}]^{2},
\end{aligned}
\tag{5.1}
$$

where $m = 1 - 1/n$ and n, p_e are the van Genuchten parameters equal to $n = 2$, $p_e = 2MPa$. Curves for the van Genuchten relative permeabilities are presented in Figure 5.1 and for the capillary pressure in Figure 5.2

In most cases, this type of parametrization is enough to capture the real processes well enough [1]. However, it is important to mention that the relation between the capillary pressure and the wetting fluid saturation is not unique in general. The experimental data suggests that capillary pressure curve shows history-dependent behaviour [15], [31], [4]. One way to improve the capillary pressure parametrization is to include time-dependence in the capillary pressure function $p_c = p_c(S_w, \partial_t S_w)$.

Nevertheless, in this thesis we consider algebraic relation for the van Genuchten parametrization as the capillary pressure function and the relative permeabilities functions.

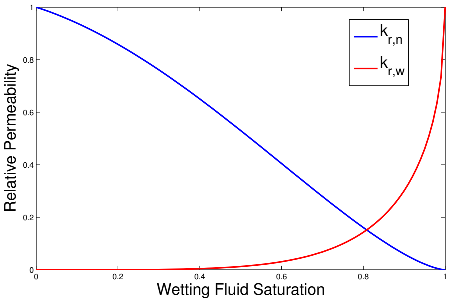

Figure 5.1: An example of the static van Genuchten relative permeabilities curves.

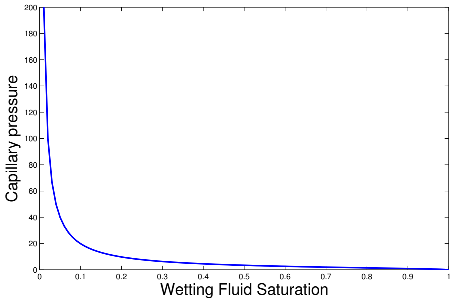

Figure 5.2: An example of the static van Genuchten capillary pressure curve.

Bibliography

[1] B. Amaziane, M. Jurak, and A. Žgaljić Keko. Modeling and numerical simulations of immiscible compressible two-phase flow in porous media by the concept of global pressure. *Transp. Porous Media*, 84(1):133–152, 2010.

[2] U. M. Ascher, S. J. Ruuth, and B. Wetton. Implicit-explicit methods for time-dependent PDE's. 32(3), 1993.

[3] J. Bear. *Dynamics of Fluids in Porous Media.* Courier Corporation, 1988.

[4] a. Y. Beliav and S. M. Hassanizadeh. A theoretical model of hysteresis and dynamic effects in the capillary relation for two-phase flow in porous media. *Transport in Porous Media*, 43(3):487–510, 2001.

[5] Z. Chen. *Finite element methods and their applications.* Springer Science & Business Media, 2005.

[6] Z. Chen, G. Huan, and B. Li. An improved IMPES method for two-phase flow in porous media. *Transport in Porous Media*, 54(3):361–376, 2004.

[7] Z. Chen, G. Huan, and Y. Ma. *Computational methods for multiphase flows in porous media*, volume 2. Siam, 2006.

[8] R. Clough. *The Finite Element Method in Plane Stress Analysis.* American Society of Civil Engineers, 1960.

[9] K. Coats. A Note on IMPES and Some IMPES-Based Simulation Models. *SPE Journal*, 5(3):14–17, 2000.

[10] K. H. Coats. IMPES stability: The stable step, 2001.

[11] K. H. Coats. IMPES Stability: The CFL Limit. *SPE Journal*, 8(3):291–297, 2003.

[12] J. J. Connor and C. A. Brebbia. *Finite element techniques for fluid flow.* Newnes, 1976.

[13] C. Frepoli, J. H. Mahaffy, and K. Ohkawa. Notes on the implementation of a fully-implicit numerical scheme for a two-phase three-field flow model. *Nuclear Engineering and Design*, 225:191–217, 2003.

[14] B. Ganis, K. Kumar, G. Pencheva, M. F. Wheeler, and I. Yotov. A Multiscale Mortar Method and Two-Stage Preconditioner for Multiphase Flow Using a Global Jacobian Approach. *SPE Large Scale Computing and Big Data Challenges in Reservoir Simulation Conference and Exhibition*, 2014.

[15] S. M. Hassanizadeh, M. a. Celia, and H. K. Dahle. Dynamic Effect in the Capillary Pressure–Saturation Relationship and its Impacts on Unsaturated Flow. *Vadose Zone Journal*, 1(1):38, 2002.

[16] W. Hundsdorfer and S. J. Ruuth. IMEX extensions of linear multistep methods with general monotonicity and boundedness properties. *Journal of Computational Physics*, 225(2007):2016–2042, 2007.

[17] A. Iserles. *A first course in the numerical analysis of differential equations.* Number 44. Cambridge University Press, 2009.

[18] P. Knabner and L. Angermann. *Numerical methods for elliptic and parabolic partial differential equations.* 2003.

[19] J. Kou and S. Sun. A new treatment of capillarity to improve the stability of IMPES two-phase flow formulation. *Computers and Fluids*, 39(10):1923–1931, 2010.

[20] J. Kou and S. Sun. On iterative IMPES formulation for two phase flow with capillarity in heterogeneous porous media. *International Journal of Numerical Analysis and Modeling, series B*, 1(1):20–40, 2010.

[21] B. H. Kueper and E. Frind. Two-Phase Flow in Heterogeneous Porous Media 1. Model Development. *Water Resources Research*, 27(6):1049–1057, 1991.

[22] S. Lacroix, J. Wheeler, and M. Wheeler. Flow in Porous Media Fully Implicitly. 25(3):905–926, 2003.

[23] T. Lee, M. Leok, and N. H. McClamroch. Geometric numerical integration for complex dynamics of tethered spacecraft. *Proceedings of the 2011 American Control Conference*, (July 2006):1885–1891, 2011.

[24] F. List and F. A. Radu. A study on iterative methods for solving Richards‘ equation. *Computational Geosciences*, (arXiv:1507.07837), 2015.

[25] B. Lu and M. F. Wheeler. Iterative coupling reservoir simulation on high performance computers. *Petroleum Science*, 6(1):43–50, 2009.

[26] J. M. Nordbotten and M. A. Celia. *Geological storage of CO2: modeling approaches for large-scale simulation*. John Wiley & Sons, 2011.

[27] I. S. Pop, F. Radu, and P. Knabner. Mixed finite elements for the Richards' equation: Linearization procedure. *Journal of Computational and Applied Mathematics*, 168(1-2):365–373, 2004.

[28] F. A. Radu, J. M. Nordbotten, I. S. Pop, and K. Kumar. A robust linearization scheme for finite volume based discretizations for simulation of two-phase flow in porous media. *Journal of Computational and Applied Mathematics*, 289:134–141, 2015.

[29] J. Stoer and R. Bulirsch. Introduction to numerical analysis, 1993.

[30] V. Thomée. From finite differences to finite elements. *Journal of Computational and Applied Mathematics*, 128(1-2):1–54, 2001.

[31] C. J. van Duijn, X. Cao, and I. S. Pop. Two-Phase Flow in Porous Media: Dynamic Capillarity and Heterogeneous Media. *Transport in Porous Media*, 2015.

[32] M. T. van Genuchten. A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils1, 1980.