Iterative schemes for solving coupled, non-linear flow and transport in porous media

University of Bergen

Davide Illiano

December 2016

Abstract

In this thesis we compare different iterative approaches for solving the non-linear, coupled multiphase flow and reactive transport in porous media. Especially, we consider two-phase flow and one a-half phase flow (modeled by *Richards equation*) coupled with an one component transport equation. Implicit and explicit iterative schemes will be compared in terms of efficiency, robustness and accuracy. The best approach seems to be a fully implicit scheme, which is a slightly modified variant of the classical splitting iterative scheme for coupled equations.

We concentrate on three linearization methods: *L-scheme, Modified Picard* and *Newton*. We implement them in Matlab and tested on both an academic example, built from a manufactured analytical solution and on a realistic problem, the *Salinity Problem*.

In the first part of the thesis we will also briefly study the generic two-phase flow plus transport equation in porous media, presenting some of the most common solving algorithms such as the *IMPES* method and its *Fully Implicit* reformulation.

To my girlfriend Siri and to my family in Italy.

Acknowledgements

I would first like to thank my supervisor Professor Adrian Florin Radu, without whom this thesis would not exist. During the last year he has always been available answering my continuous questions and giving me important suggestions and concrete help.

I also would like to thank the Abel Prize Organization, thanks to them, and the funds that I received from them, I have been able to spend almost two months at the Princeton University working on my thesis. I must also thank, for their incredible help, Professor Michael Celia and the Associate Researcher Karl Bandilla from the Civil Environment Engineering Department at Princenton.

Finally, I must express my deepest gratitude to my family in Italy, who has always believed in me and supported me since I came to this world and my girlfriend Siri, the one who has pushed me forward and helped me in any difficult moment, not only during this long work but in every single day.

Contents

Chapter 1

Introduction

Multiphase flow and *transport* in *porous media* represents an important branch of applied mathematics, many problems from different areas can be studied thanks to these models. To name a few, whether you are studying enhanced oil recovery, CO_2 storage, diffusion of medical agents into the body or groundwater flows, you will have to model multiphase flow and transport in porous media. It is then fundamental to be able to predict the behaviour of such fluids, in the view of e.g. increasing the oil recovery or understanding how dangerous a contaminated site is or can become. In the same spirit, numerical simulations are extremely important, they are our only instrument to forecast these complex situations.

Considered the vast range of problems covered by this particular field of applied mathematics, continuous are the challenges offered by the industries. Our aim, in this thesis, is to contribute on the understanding and further development of numerical algorithms for flow and transport in porous media, which can be used for reliable and efficient simulations of realistic problems.

The mathematical models for flow and transport in porous media are given in terms of coupled non-linear Partial Differential Equations (*PDEs*). Despite of intensive research in the last decades [18, 20, 37, 41], there is still a strong need for reliable algorithms to solve them.

We will start this work studying a generic two-phase flow and transport in porous media problem, presenting some of the common solving techniques. After that, we will move our attention to a particular case of flow: groundwater flow in saturated/unsaturated media. The *Richards equation* will be used to model such phenomena. This equation are fully coupled to a reactive transport due to the presence, for example, of a surfactant [1] into the water phase. The two equations will result coupled due to the double dependence of the *water content* and the *hydraulic conductivity* from both pressure head and concentration of the external component.

To solve coupled flow and transport equations numerically, we need first to discretize both in time and in space. There exist many techniques to proceed into the discretization of *PDEs*. Regarding the time, the most used are backward and forward Euler approaches [15, 16]. In this thesis we concentrate mainly on the second one. For the space discretization there are even more techniques available, to name the most common, we have: Finite Element Method (*FEM*) [24], Mixed Finite Element Method (*MFEM*) [33, 50, 46, 45, 13, 44], Multi-Point Flux Approximation (*MPFA*) [43] and Finite Volume Method (*FVM*) [39, 40, 36]. Two Point Flux Approximation (*TPFA*), a particular case of *MPFA*, will be the space discretization

scheme used in this work. A simulationn for one dimensional, fully coupled flow (two-phase or Richards) and transport in porous media, based on forward Euler and TPFA has been implemented in Matlab.

Regarding the reactive transport we can mention e.g. [18, 19, 20, 37, 38, 41, 42, 48, 49]. We refer also to [47] for a discussion over the numerical dispersion for different schemes for transport and flow in porous media. For an overview over convergence results for reactive flow we cite [48].

The main part of this work will be solving the Richards and transport equations by efficient and reliable algorithms. These two coupled equations are characterized by their nonlinearity. To deal with this, we implement the most used linearization schemes. We will concentrate on:

- L-scheme
- Modified Picard
- Newton method

Once the two equations have been linearized we still have a coupled system to solve. We will then present five different approaches to study Richards and transport, concentrating mainly on a *fully implicit iterative* method and a *monolithic* method. We will observe that such approaches represent a valid alternative to the solving technique usually used. The fully implicit scheme proposed here is remarkably faster and equally accurate. A fully explanation and comparison of the different approaches will be given in Chapter 6. Such comparison will be based on two numerical examples:an academic one, having a known analytical solution and a real case study.

Regarding the latter we considered the *salinity problem*, a classical example of one dimensional Richards and reactive transport problem.

A comprehensive study on the different approaches for solving coupled flow and transport is performed. We considered five main schemes, and each of them in combination with one of the linearization schemes: L-scheme, modified Picard and Newton scheme. The performance of the scheme was studied with respect to:

- robustness
- accuracy
- CPU time

The rest of the thesis is organized as follow. In Chapter 2 we provide an overview of the two-phase flow and transport in porous media, presenting also, in Chapter 3, the numerical model used to simulate such problem. We will use the *IMPES* method and its *fully implicit* reformulation to solve the system of equations obtained. In Chapter 4 we will move our attention to the Richards and transport equations, in the same chapter we will also present the linearization schemes implemented to treat the nonlinearity of the two equations. Chapter 5 presents some interesting numerical results, we built such examples on a manufactured solution used to verify the efficiency of our algorithms, for both the generic two-phase flow plus transport

and Richards plus transport. In Chapter 6 we will present five different approaches to treat the coupled problem given by Richards and transport and we will observe as, the schemes used in the previous chapter, represent a valid alternative to the more common formulation. Such comparison is built using the same academic example of Chapter 5. These different approaches will be also investigated applying them to a real case study, in Chapter 7 , the *salinity problem*, will be studied and again the different methods will be compared. Chapter 8 will end our thesis, presenting our final conclusions and observations.

Chapter 2

Mathematical model

This chapter will be used to introduce the mathematical models for multiphase flow and transport in porous media. To do so we need to define the quantities and equations which will be used to simulate these phenomena.

We will start obtaining the mass conservation equation, after that, the Darcy's law will be introduced and we will complete our system of equations with the transport. Such system is not closed and so it can not be solved without the introduction of additional constitutive equations.

2.1 Mass conservation

Porosity

The soil under our feet has always a complex structure, we have to understand that many porous and so empty spaces are included in what we call *ground*. To better describe this situation a new quantity has been introduced. Such quantity, the *porosity* (Φ), is defined as the fraction between the volume of the voids space over the total volume. It is a dimensionless quantity between 0 and 1.

$$\Phi = \frac{V_v}{V_T}$$

The definition given above is clearly not good enough. Look at the point P in the figure 2.1, considering different circles (total volumes) centred in it, we obtain different values of porosity. To give a better definition of this quantity we must observe that, as defined now, porosity is a continuous function and so it has an average, figure 2.2. We call REV (Representative Elementary Volume) the part of the domain A that reflects the average of the porosity. Considering this new portion of A we can give a precise definition of $\Phi(P)$.

$$\Phi(P) = \lim_{V_T \to REV} \frac{V_v}{V_T} \tag{2.1}$$

With this new definition we can try to understand how to compute the mass, of oil for example, contained in our sample of soil.

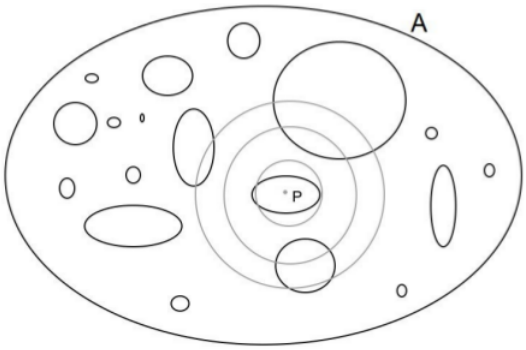


Figure 2.1: Domain A, the empty spaces are represented with elliptical shapes

Any change in the mass must be balanced, by the inflow/outflow through the boundaries of the volume itself, or by any mass added not related with the boundary fluxes (*source* or *sink*). We can try to express that using the integral notation [2]

$$\int_A \frac{\partial m}{\partial t} dV \;=\; -\oint_{\partial A} \boldsymbol{f} \cdot \mathbf{v}_n dS \;+\; \int_A \boldsymbol{r} dV \tag{2.2}$$

In the equation above A is the volume taken in consideration and ∂A is its boundary, m is the measure of mass per total volume, $\boldsymbol{f}$ the mass flux vector, $\mathbf{v}_n$ the normal vector of the boundary pointing outward and $\boldsymbol{r}$ the sources or the sinks inside the volume. If $\boldsymbol{r}$ is equal to zero then the equation 2.2 is defined as a *conservation law*. The quantities introduced above can be rewritten as

$$m = \rho\Phi \quad \boldsymbol{f} = \rho\mathbf{u} \quad \boldsymbol{r} = \boldsymbol{\Psi} \tag{2.3}$$

Ψ is the external source or sink in term of mass, ρ is the density of the fluid and $\mathbf{u}$ is the flux vector. Using the *divergence theorem*

$$\int_V (\nabla \cdot \mathbf{F}) dV \;=\; \oint_S (\mathbf{F} \cdot \mathbf{v}_n) dS$$

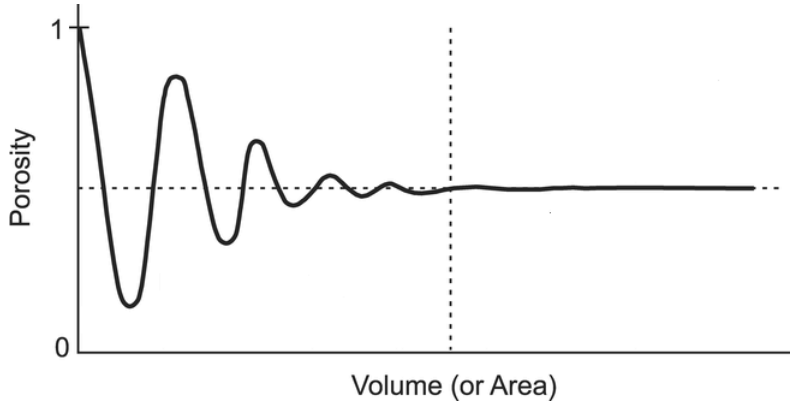


Figure 2.2: Domain A, graph of the porosity, imagine modified from [6]

the equation 2.2 can be reformulated as

$$\int_A \left(\frac{\partial m}{\partial t} + \nabla \cdot \boldsymbol{f} - \boldsymbol{r} \right) dV = \int_A \left(\frac{\partial \rho \Phi}{\partial t} + \nabla \cdot (\rho \mathbf{u}) - \boldsymbol{\Psi} \right) dV = 0. \tag{2.4}$$

Assuming sufficient regularity of the terms involved we can obtain the final expression for the *mass conservation* equation

$$\frac{\partial (\rho \Phi)}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = \boldsymbol{\Psi}. \tag{2.5}$$

2.1.1 Component mass conservation

In this thesis we will give extreme importance to multiphase flows, for this reason we must introduce a formulation of (2.5) valid also for more general cases, precisely cases in which more phases and components are involved. Before to do that we need to define a new quantity, the *phase saturation* S_α. Such quantity represents the portion of pore space occupied by the phase α.

With this new notation we can rewrite the quantities defined above as

$$m = \sum_\alpha \rho_\alpha \Phi S_\alpha m_\alpha^i \quad \boldsymbol{f} = \sum_\alpha \rho_\alpha \mathbf{u}_\alpha m_\alpha^i + \mathbf{j}_\alpha^i \quad r = \sum_\alpha \boldsymbol{\Psi}_\alpha^i,$$

these expressions are now valid for multi phases flow in which each phase α can have more components i (m_α^i represents the mass of the component i in the phase α, $\mathbf{j}_\alpha^i$ its non-advectives flux). In the following we will consider immiscible fluids so that $\mathbf{j}_\alpha^i$ will be set to zero. Precisely, for each component of the system, we obtain

$$\sum_\alpha \frac{\partial (\rho_\alpha \Phi S_\alpha m_\alpha^i)}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha m_\alpha^i + \mathbf{j}_\alpha^i) = \sum_\alpha \boldsymbol{\Psi}_\alpha^i. \tag{2.6}$$

2.2 Darcy's law

In all the equations above we find the flux $\boldsymbol{f}$, then rewritten using the notation $\boldsymbol{f} = \rho \mathbf{u}$ but we have not explained the meaning of this quantity. The flux $\boldsymbol{u}$ represents the volume of fluid per total area (both solid and fluid) per time.

The *Darcy's law* will give us a better expression for this quantity. The equation has been named after Henry Darcy, a French engineer, who, after a series of experiments, managed to express, with an equation, the quantity of water flowing through a column of sand. The law obtained is clearly empirical and valid only in particular conditions[1], conditions that we will always fulfil. The Darcy velocity or flux $\mathbf{u}$ is then given by

$$\mathbf{u} = -\frac{\mathbf{k}}{\mu}(\boldsymbol{\nabla} p + \rho g \boldsymbol{\nabla} z) \tag{2.7}$$

[1] laminar flow in saturated media under steady state conditions

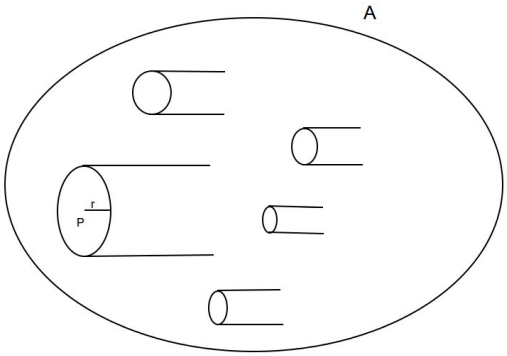


Figure 2.3: figure
Lamina flow through cylindrical pipes

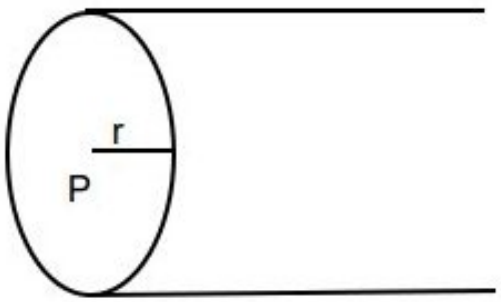


Figure 2.4: figure
A particular cylindrical pipe

where $\mathbf{k}$ is the permeability, a property of the soil not depending from the fluid, μ is the viscosity, which represents how easily a fluid can flow, p the pressure and g the gravity.
Such equation can be rewritten in different forms

$$\mathbf{u} = -\frac{\mathbf{k}}{\mu}(\boldsymbol{\nabla} p + \rho g \boldsymbol{\nabla} z) = -\frac{\mathbf{k}}{\mu}(\boldsymbol{\nabla} p + \rho g \mathbf{e}_z) = -\frac{\mathbf{k}}{\mu}(\boldsymbol{\nabla} p + \rho \mathbf{g}) \tag{2.8}$$

assuming that the vertical direction is denoted with the coordinate z.

It is possible, for easy configurations, to prove analytically the Darcy's Law, our aim is to consider one of such configurations and proceed into the demonstration.

Let's study a laminar flow through cylindrical pipes, the porous media is represented by the figure (2.3). Consider one of such cylindrical pipes, figure (2.4) and proceed to compute the flux. Starting from the *Navier-Stocke's* equation and using the cylindrical coordinates we get

$$\mu \left(\frac{d^2 \mathbf{u}}{d^2 r} + \frac{1}{r} \frac{d\mathbf{u}}{dr} \right) = \frac{dp}{dx} \tag{2.9}$$

with the boundary condition $\mathbf{u}(R) = 0$, R the radius of the cylinder.
By simple computation it is possible to obtain

$$\mathbf{u}(r) = \frac{1}{4\mu} \frac{dp}{dx} r^2 + c_1 \log r + c_2$$

and using the boundary condition

$$\mathbf{u}(r) = \frac{1}{4\mu} \frac{dp}{dx} (r^2 - R^2) + c_3.$$

If we proceed to study the averaged velocity $\overline{\mathbf{u}}$

$$\overline{\mathbf{u}} = \frac{1}{\pi R^2} \iint_D \mathbf{u} dA$$

we obtain, by solving the double integral

$$\overline{\mathbf{u}} = -\frac{R^2}{8\mu} \frac{dp}{dx}$$

Defining the discharge through the cylinder i as

$$Q_i = \overline{\mathbf{u}_i} \cdot \pi R_i^2$$

then the specific discharge $\mathbf{u}$ through the total area is

$$\begin{aligned}\mathbf{u} &= \frac{Q_{tot}}{A_{tot}} = \frac{1}{A_{tot}} \sum_{i=1}^{n} Q_i \\ &= -\frac{1}{\mu} \underbrace{\left(\frac{\pi}{8A_{tot}} \sum_{i=1}^{n} R_i^4 \right)}_{\mathrm{K}} \frac{dp}{dx}\end{aligned}$$

We have then obtained the one dimensional Darcy's law for non gravitational flows, where the permeability $\mathbf{k}$ is given by the quantity between the parenthesis.

2.2.1 Two-phase extension of Darcy's Law

As in the case of the mass conservation, also for the Darcy's law an extension is required, especially considering the aim of this thesis. In the sample of soil that we are studying two fluids are moving. Let's state the Darcy's velocity for each of the phases represented by α

$$\mathbf{u}_\alpha = -\frac{\mathbf{k}^{r,\alpha}\mathbf{k}}{\mu_\alpha}(\nabla p_\alpha + \rho_\alpha g \mathbf{e}_z) \;=\; -\frac{\mathbf{k}^{r,\alpha}\mathbf{k}}{\mu_\alpha}(\nabla p_\alpha + \rho_\alpha \mathbf{g}) \tag{2.10}$$

In our problem two fluids are taken in consideration. The situation is much more complex and for this reason the *relative permeability* $\mathbf{k}^{r,\alpha}$ is introduced. This new quantity represent the reduction of pore space available for the phase α. All the pores in which the other phase is flowing are considered as solid soil because α can not flow through them. For this reason a reduction of the permeability is required.

The equation above can be rewritten thanks to the introduction of a new quantity, the *mobility* $\boldsymbol{\lambda}_\alpha$:

$$\boldsymbol{\lambda}_\alpha \;=\; \frac{\mathbf{k}^{r,\alpha}}{\mu_\alpha} \tag{2.11}$$

we then obtain

$$\mathbf{u}_\alpha = \; -\boldsymbol{\lambda}_\alpha \mathbf{k}(\nabla p_\alpha + \rho_\alpha \mathbf{g}) \tag{2.12}$$

Relative permeability

The *relative permeability* is a function of the saturation. Very explicative can be the graph, figure 2.5, which shows the relation between $\mathbf{k}_{r,\alpha}$ and S_α.

Two main facts must be noticed, the *permeability* is a non-linear function of the *saturation* and $\mathbf{k}_{r,\alpha}$ goes to zero before the phase saturation itself goes to zero. The value of the saturation for which the relative permeability is zero is called *residual saturation*, denoted by S_α^{res}. Such quantity shows that part of the phase is still into the soil and it can not be recovered. The phase has failed in creating a continuous path through which flow of the phase itself can take place.

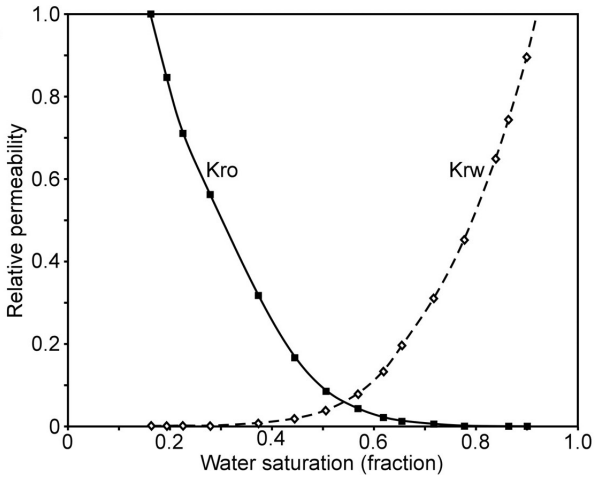


Figure 2.5: Graph of relative permeabilities as function of the saturation, oil phase and water phase, [3]

2.3 Transport equation

The last equation that must be introduced for modeling our problem is the *transport equation*. In case of multiple components, it describes how they are transported in the phase. Precisely, later in our work, we will study how the salt is transported, through the water, into the soil. The equation, which models this phenomenon, is

$$\Phi \frac{\partial (S_w c)}{\partial t} = \nabla \cdot (D\Phi S_w \nabla c - \mathbf{u}c) \ + \ Q(\mathbf{x}, t) \ + \ R(c) \tag{2.13}$$

where $c(x,t)$ is the concentration of the component C, D is the diffusion-dispersion coefficient and Q is used to describe all sources and sinks of the quantity C and R is the reaction term. Some words must be spent to described the quantities introduced above. The mass concentration c is defined as the fraction between the mass of the component and the volume of the phase. The term $\nabla \cdot (D\Phi S_w \nabla c)$ represents the diffusion-dispersion term, it describes both the molecular and the mechanical dispersion, in our work we won't differentiate them but we will consider them as a singular phenomenon; this is justified by the fact that they can not be distinguished on the Darcy scale, we should consider a molecular scale. $\nabla \cdot (\mathbf{u}c)$ is the advection term and it represents the transport of the components with the flow velocity. The main difference between the advective transport and the diffusion is that the first is due to the bulk motion of the fluid, the second represents instead the movement of the molecular from area with high concentration, of the component, to area with lower concentration.

We have then obtained all the equations necessary to describe our problem. We obtained a system of five equations, two mass conservation and two Darcy's velocities, one for each of the phases, and the transport equation.

$$
\begin{aligned}
&\frac{\partial(\rho_o \Phi S_o)}{\partial t} + \nabla \cdot (\rho_o \mathbf{u}_o) \;=\; \mathbf{\Psi}_o \\
&\frac{\partial(\rho_w \Phi S_w)}{\partial t} + \nabla \cdot (\rho_w \mathbf{u}_w) \;=\; \mathbf{\Psi}_w \\
&\mathbf{u}_o = \; -\boldsymbol{\lambda}_o \mathbf{k}(\nabla p_o + \rho_o \mathbf{g}) \\
&\mathbf{u}_w = \; -\boldsymbol{\lambda}_w \mathbf{k}(\nabla p_w + \rho_w \mathbf{g}) \\
&\Phi \frac{\partial(S_w c)}{\partial t} = \nabla \cdot (D \Phi S_w \nabla c - \mathbf{u} c) \;+\; Q(\mathbf{x}, t) \;+\; R(c)
\end{aligned}
$$

2.4 Reformulation of the mathematical model through constitutive relations

The model presented in the section above is a well known problem of Applied Mathematics and Physic, unfortunately, such system is not closed and it can not be solved. Even in the case of a simpler situation, in which only one component for phase is considered and the transport equation is neglected, the system remains not closed. In the simplify case we have four equations (2 Darcy's equations and 2 mass conservation equations, one for each phase) and six unknowns (2 Darcy's velocity, 2 saturations and 2 pressures).

We can introduce an other equation observing that the sum of the saturations must be 1

$$S_w + S_o \;=\; 1 \tag{2.14}$$

even with this new requirement the system remains unclosed.

It is then clear that some constitutive relations are required, in the following section we present them and we obtain a reformulation of the system which allowed us to solve it.

2.4.1 Capillary pressure

In a two-phase flow we always have a *wetting* phase and a *non wetting* phase. We define wetting phase the phase most attracted by the solid, with a contact angle $\theta < 90°$.

Studying the water-oil configuration, water is the wetting phase, oil the non wetting. We can then define the wetting pressure p_w and the non wetting pressure p_n, the difference between these two quantities is defined as *capillary pressure*

$$p_c \;=\; p_n - p_w.$$

Such quantity is used to express the iteration between the two phases. From a microscopic point of view when the two phases present a different pressure the

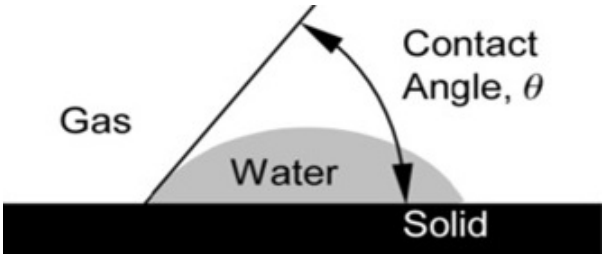


Figure 2.6: Contact angle [4]

interface between them deforms and achieves a curvature given by the Young-Laplace equation:

$$(p_{cap})_{max} = \frac{2\gamma \cos\theta}{r}$$

γ is the interfacial tension, θ is the contact angle and r is the radius of the pore. Such equation represents the maximum value of the capillary pressure, if such value is overpassed then the membrane of interface will break and the fluids will start to move. Precisely, assuming we are studying a porous sample with oil phase and water phase in it, if we increase the oil pressure p_o the fluids will not automatically start to move. First the interfacial membranes will be deformed until some of them, the one in the bigger pores (bigger radius implies smaller maximum value for p_{cap}) will break and only then there will be a movement of the fluids.

As for the *relative permeability*, the *capillary pressure* is a function of the saturation. The graph given by Fig 2.7 shows us a classicle profile for the capillary pressure.

To understand the graph we must define some quantities which are present in it, when non wetting fluid is displacing wetting fluid, the process is called *drainage*, *imbibition* in the opposite case [2]. We can then observe that the image is an example of hysteresis, we obtain different curves in the two cases defined above. As for the *relative permeability* also the *capillary pressure* is a non-linear function of the saturation. Other similitude, for the same reason as before, is the presence of a *residual saturation*.

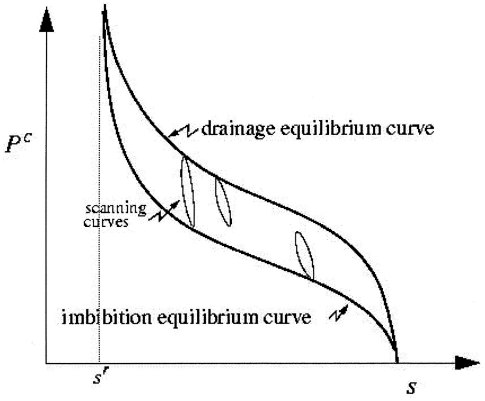


Figure 2.7: Capillary pressure as function of the saturation. [5]

2.4.2 Average Pressure

Our formulation can be further simplified introducing the *average pressure* as in [2]

$$\overline{p} = \frac{p_w + p_n}{2}. \tag{2.15}$$

The phase pressures can then be rewritten:

$$p_w = \overline{p} - \frac{1}{2}p^{cap} \quad and \quad p_n = \overline{p} + \frac{1}{2}p^{cap} \tag{2.16}$$

and defining the mobilities

$$\boldsymbol{\lambda}_\alpha = \frac{\mathbf{k}_{r,\alpha}}{\mu_\alpha} \quad and \quad \boldsymbol{\lambda}_\Sigma = \boldsymbol{\lambda}_w + \boldsymbol{\lambda}_n \tag{2.17}$$

we obtain the final expressions for the Darcy's velocities

$$\mathbf{u}_w = -\boldsymbol{\lambda}_w \mathbf{k}[\nabla(\overline{p} - \frac{1}{2}p^{cap}) - \rho_w \mathbf{g}] \tag{2.18}$$

$$\mathbf{u}_n = -\boldsymbol{\lambda}_n \mathbf{k}[\nabla(\overline{p} + \frac{1}{2}p^{cap}) - \rho_n \mathbf{g}] \tag{2.19}$$

$$\mathbf{u}_\Sigma = -\boldsymbol{\lambda}_\Sigma \mathbf{k}\nabla\overline{p} + (\rho_w \boldsymbol{\lambda}_w + \rho_n \boldsymbol{\lambda}_n)\mathbf{k}\mathbf{g} - \frac{(\boldsymbol{\lambda}_n - \boldsymbol{\lambda}_w)\mathbf{k}}{2}\nabla p^{cap} \tag{2.20}$$

2.4.3 Pressure equation and Saturation equation

In this section we will take in consideration many conditions which will led us to some important simplifications. We will consider: fluids and solid matrix (the soil) incompressible and *density* and *porosity* constant in time. Also, the fluids are immiscible, non-diffusive, and the spatial derivatives of the density are neglected. Thanks to these conditions the equation 2.6 can be rewritten as

$$\Phi\frac{\partial S_w}{\partial t} + \nabla \cdot \mathbf{u}_w = \frac{\boldsymbol{\Psi}_w}{\rho_w} \tag{2.21}$$

$$\Phi\frac{\partial S_n}{\partial t} + \nabla \cdot \mathbf{u}_n = \frac{\boldsymbol{\Psi}_n}{\rho_n} \tag{2.22}$$

summing the two equations and recalling (2.14) we obtain the *pressure equation*

$$\nabla \cdot (\mathbf{u}_w + \mathbf{u}_n) = \nabla \cdot \mathbf{u}_\Sigma = \sum_{\alpha=n,w} \frac{\boldsymbol{\Psi}_\alpha}{\rho_\alpha} \tag{2.23}$$

The *saturation equation* is given by the substitution of 2.19 into 2.22

$$\Phi\frac{\partial S_w}{\partial t} - \nabla \cdot \{\boldsymbol{\lambda}_w \mathbf{k}[\nabla(\overline{p} - \frac{1}{2}p^{cap}) - \rho_w \mathbf{g}]\} = \frac{\boldsymbol{\Psi}_w}{\rho_w} \tag{2.24}$$

We have then obtained a system of equations that can describe our two-phase problem

$$\begin{aligned} \Phi\frac{\partial S_w}{\partial t} - \nabla\cdot\{\boldsymbol{\lambda}_w\mathbf{k}[\nabla(\overline{p} - \frac{1}{2}p^{cap}) - \rho_w\mathbf{g}]\} &= \frac{\boldsymbol{\Psi}_w}{\rho_w} \\ \nabla\cdot(\mathbf{u}_w + \mathbf{u}_n) = \nabla\cdot\mathbf{u}_\Sigma &= \sum_{\alpha=n,w}\frac{\boldsymbol{\Psi}_\alpha}{\rho_\alpha} \end{aligned} \tag{2.25}$$

such system, imposing appropriate boundary and initial conditions, can be solved numerically.

2.4.4 Transport equation

The last equation that we must reformulate to model our problem is the transport equation 2.26. In such equation no indexes are included because we wanted to state a general case. If we apply it to our particular problem, in which the component is transported into the wetting phase, some specifications are required

$$\Phi\frac{\partial(S_w c)}{\partial t} + \nabla\cdot(-\Phi D S_w\nabla(c) + \mathbf{u}_w c) = R(c) + f(x,t). \tag{2.26}$$

As we said before we will study, for example, the salt dissolved into the water phase and, for this reason, we are considering S_w and $\mathbf{u}_w$. $f(x,t)$ represents the external forces while $R(c)$ is the reaction term. We will study the second quantity supposing different dependences from the concentration.

Chapter 3

Numerical model for two-phase flow and transport

The system given by the equations 2.25 can not be solved analytically for general cases, this chapter will be used to develop a *numerical model* to solve such system. We will then be able to obtain approximated solutions, which will be presented in the next chapters. Numerical modelling has a fundamental role in many subjects as: math, physics and engineering, physical phenomena are described by complex equations which can not be solved analytically. For this reason many solving techniques have been developed, we will present and use here some of them. Our first step will be to discretize the domain of the problem given by the system 2.25 and to implement a code to solve it in *Matlab*.

We will use, for example, a so called *IMPES* approach, both explicit and implicit techniques will be implemented, the pressure equation will be solved implicitly (*IMP*) and the saturation explicitly (*ES*).

The pressure equation will be solved and the solution obtained will be used in the computation of the saturation, the system 2.25 is an example of a *coupled problem*.

We will then used the values of pressure and saturation obtained into the transport equation.

To implement this models we need, first of all, to discretize our domain, passing from a continuous to a discretized problem.

3.1 Discretization

In this section we will present how to define a mesh on our domain and how to use such mesh in the numerical computation.We will study our problem in 1D (one dimension) to make it easier and more understandable. Our domain will then be a segment of length L, for example $X = [0, L]$. We want now to divide this interval in smaller intervals I_j of the form $I_j = [x_j, x_{j+1}]$ where $0 = x_1 \leq x_2 \leq \cdots \leq x_{N-1} \leq x_N = L$. Two approaches can be used, we can decide to divide it in intervals of different length $h_j = \|x_{j+1} - x_j\|$ or of equal length $h_j = h_i = h$.
For simplicity reasons we will choose the second approach obtaining the discretization given in figure 3.1

Such discretization has been obtained using a Matlab command, *linspace*, which

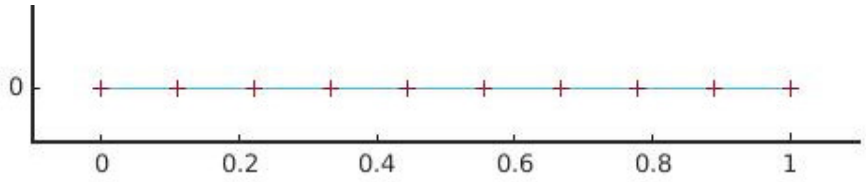


Figure 3.1: Space Discretization

divide an interval in a fixed number of points producing subintervals of equal length. This process is called gridding and it can be implemented also for 2D and 3D domains. In the first case the shapes used are triangles, we call it *triangulation*, in the second case they are usually tetrahedron. During this process some properties must be verified, in the case of the triangulation it is required that:

- the set of the triangles gives a partition of the domain
- if two triangles have more than a point in common, such points are on the common edge

In our system of equation a dependence from the time is also evident. We will then proceed to discretize the time domain, in our case $[0, T]$ where T represents the ending time. We will divide such interval in subintervals of equal length Δt. Combining the two grids obtained, respectively for the space and the time domains, we get a time-space grid, which will represents our new domain. Each function of our problem will be then computed in each point of the grid, considering for example the pressure $p(t, x)$ we will obtain:

$$\mathbf{p}(t,x) = \begin{bmatrix} p(t_0,x_1) & p(t_0,x_2) & \dots & \dots & p(t_0,x_{N+1}) \\ p(t_1,x_1) & p(t_1,x_2) & \dots & \dots & p(t_1,x_{N+1}) \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ p(t_N,x_1) & p(t_N,x_2) & \dots & \dots & p(t_N,x_{N+1}) \end{bmatrix}$$

3.2 Approximation techniques

In this section the main approximation techniques used in the thesis will be presented. We will study how to compute integrals and derivatives numerically.

3.2.1 Taylor series

The most known approximation technique for derivatives is the *Taylor series*. Given the definition of derivative of a function f in the point x

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{3.1}$$

we can then express the function in the point $x + h$ as

$$f(x+h) \;=\; f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x) + \dots \tag{3.2}$$

and from the equation above is possible to obtain an expression, *forward difference,* for the derivative in the point x

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} = f'(x) + \frac{h}{2}f''(x) + \frac{h^3}{3!}f^{(3)}(x) + O(h^3). \tag{3.3}$$

An alternative formulation, called *centred difference* is given by

$$f'(x) \approx \frac{f(x+\frac{h}{2}) - f(x-\frac{h}{2})}{h} = f'(x) + \frac{h^2}{24}f^{(3)}(x) + O(h^4). \tag{3.4}$$

such equation is obtained combining 3.2 and the series of the function computed in $x - h$

$$f(x-h) \;=\; f(x) - hf'(x) + \frac{h}{2}f''(x) - \frac{h^3}{3!}f^{(3)}(x) + O(h^3). \tag{3.5}$$

Observing equations 3.3 and 3.5 it is possible to notice that in the first case the truncation error is $O(h)$ while, using a centred differences is $O(h^2)$.

3.2.2 Two Point Flux Approximation

The *TPFA* is a particular type of *finite volume method* used to solve *pde*s expressing conservation of one or more quantities. The meshes introduced above will be used, each subinterval is called control volume and the integration of the *pde* over each volumes results in a balanced equation [9].

In the one dimensional case, considering the domain $X = [0, L]$ and a mesh over it, to solve the differential equation

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial x}(K\frac{\partial p}{\partial x}) \tag{3.6}$$

we must first approximate the spacial derivative of $K\frac{\partial p}{\partial x}$ as follow

$$\begin{aligned}\frac{\partial}{\partial x}(K\frac{\partial p}{\partial x}) &= \frac{1}{\Delta x}\Big(\Big(K\frac{\partial p}{\partial x}\Big)_{(x_{i+1/2})} - \Big(K\frac{\partial p}{\partial x}\Big)_{(x_{i-1/2})}\Big) \\ &= \frac{1}{\Delta x}\Big(\Big(K\mid_{(x_{i+1/2})}\frac{\partial p}{\partial x}\mid_{(x_{i+1/2})}\Big) - \Big(K\mid_{(x_{i-1/2})}\frac{\partial p}{\partial x}\mid_{(x_{i-1/2})}\Big)\Big)\end{aligned}$$

rewriting $K\mid_{(x_{i+1/2})}$ as

$$K\mid_{(x_{i+1/2})} = \;\frac{K(x_i) + K(x_{i+1})}{2}$$

and an analogous function for $K\mid_{(x_{i-1/2})}$, we then obtain

$$\frac{\partial}{\partial x}(K\frac{\partial p}{\partial x}) = \frac{1}{\Delta x}\Big(\Big(\frac{K(x_i) + K(x_{i+1})}{2}\frac{\partial p}{\partial x}\mid_{(x_{i+1/2})}\Big) - \Big(\frac{K(x_i - 1) + K(x_i)}{2}\frac{\partial p}{\partial x}\mid_{(x_{i-1/2})}\Big)\Big).$$

Last step now is to approximate the remaining space derivatives in the points $x_{i+1/2}$ and $x_{i-1/2}$ as

$$\frac{\partial p}{\partial x} \mid_{(x_{i+1/2})} = \frac{p(x_{i+1}) - p(x_i))}{dx}$$

again an analogous formula can be used for the derivative in $x_{i-1/2}$.

We have then written an approximation of the right hand side, we will then be able to solve the given *pde*.

This easy example has been presented to show how to use *TPFA*, a scheme which will be fundamental in the following.

3.2.3 Midpoint approximation

Compute integrals can also be challenging, for this reason many techniques have been developed, one of the most common is the midpoint rule. Again the interval $[a, b]$ on which the integral is defined will be divided in N subintervals of equal length $h = \frac{b-a}{N}$. The values of the function in the middle points l_i, of such intervals, will be added together giving

$$\int_a^b f(x)dx \approx h \sum_{i=1}^{N} f(l_i) \qquad 1 \leq i \leq N \tag{3.7}$$

and on each interval

$$\int_{x_i-1/2}^{x_i+1/2} f(x)dx \approx hf(x_i). \tag{3.8}$$

This method will be implemented when, using the *fundamental theorem of calculus*, we will reduce the order of the derivatives.

3.2.4 Forward and backward Euler

In the system 2.25 some time derivatives must be solved, the most common approximation is given simply by

$$\frac{\partial}{\partial t} p(x, t^{n+1}) = \frac{p^{n+1} - p^n}{\Delta t} \tag{3.9}$$

where Δt represents clearly the length of the mesh on the time grid. The expression above is called *forward Euler approximation*, a *backward* formulation is given instead by

$$\frac{\partial}{\partial t} p(x, t^n) = \frac{p^n - p^{n-1}}{\Delta t}. \tag{3.10}$$

We have now presented all the techniques that will be used in the following parts of the thesis. We can now start to discretize and solve the equations of the system 2.25, beginning from the *pressure equation* 2.23. What follows will be a re-elaboration of the scheme presented in [8].

3.3 Pressure equation

In this section we will explain how to discretize and implement a code in Matlab able to solve the *pressure equation*

$$\nabla \cdot (-\boldsymbol{\lambda}_\Sigma \mathbf{k} \nabla \overline{p} + (\rho_w \boldsymbol{\lambda}_w + \rho_n \boldsymbol{\lambda}_n)\mathbf{k}\mathbf{g} - \frac{(\boldsymbol{\lambda}_n - \boldsymbol{\lambda}_w)\mathbf{k}}{2} \nabla p^{cap}) = \sum_{\alpha=n,w} \frac{\boldsymbol{\Psi}_\alpha}{\rho_\alpha} \tag{3.11}$$

as said before we will study the one dimensional case and the gravity $\mathbf{g}$ will be neglected obtaining

$$\frac{\partial}{\partial x}(-\lambda_\Sigma k \frac{\partial}{\partial x} p - \frac{(\lambda_n - \lambda_w)k}{2} \frac{\partial}{\partial x} p^{cap}) = \sum_{\alpha=n,w} \frac{\Psi_\alpha}{\rho_\alpha}$$

To eliminate the first space derivative is sufficient to integrate on all the control volumes $[x_{i-1/2}, x_{i+1/2}]$

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial}{\partial x}(-\lambda_\Sigma k \frac{\partial}{\partial x} p - \frac{(\lambda_n - \lambda_w)k}{2} \frac{\partial}{\partial x} p^{cap}) dx = \int_{x_{i-1/2}}^{x_{i+1/2}} \sum_{\alpha=n,w} \frac{\Psi_\alpha}{\rho_\alpha} dx$$

and by the FTC[1]

$$(-\lambda_\Sigma k \frac{\partial}{\partial x} p - \frac{(\lambda_n - \lambda_w)k}{2} \frac{\partial}{\partial x} p^{cap})_{x_i+1/2} - (-\lambda_\Sigma k \frac{\partial}{\partial x} p - \frac{(\lambda_n - \lambda_w)k}{2} \frac{\partial}{\partial x} p^{cap})_{x_i-1/2} = \\ = \sum_{\alpha=n,w} \frac{\Psi_{\alpha,i}}{\rho_\alpha} \Delta x_i$$

We need now to evaluate all the quantities above in the points $x_{i-1/2}$ and $x_{i+1/2}$, we must then recall the variables of each of these quantities. The *permeability* K is a function of both space and time so that $k|_{x_{i+1/2}} = k(x_{i+1/2}, t)$. The *relative permeabilities* are, instead, functions of the *saturation*. We then have that $k^r(S)|_{x_{i+1/2}} = k^r(S(x_{i+1/2}, t))$ and simply using the arithmetic average $k^r(S(x_{i+1/2}, t)) = \frac{k^r(S(x_i,t)) + k^r(S(x_{i+1},t))}{2}$.

Substituting now these quantities into the equation above we obtain:

$$\begin{aligned} &- \frac{k_{i+1} + k_i}{2} \frac{\lambda_{\Sigma,i+1} + \lambda_{\Sigma,i}}{2} \frac{p_{i+1}^{n+1} - p_i^{n+1}}{\Delta x} - \frac{1}{2} \frac{\lambda_{\Delta,i+1} + \lambda_{\Delta,i}}{2} \frac{k_{i+1} + k_i}{2} \frac{p_{i+1}^{cap} - p_i^{cap}}{\Delta x} \\ &+ \frac{k_{i-1} + k_i}{2} \frac{\lambda_{\Sigma,i-1} + \lambda_{\Sigma,i}}{2} \frac{p_i^{n+1} - p_{i-1}^{n+1}}{\Delta x} + \frac{1}{2} \frac{\lambda_{\Delta,i-1} + \lambda_{\Delta,i}}{2} \frac{k_{i-1} + k_i}{2} \frac{p_i^{cap} - p_{i-1}^{cap}}{\Delta x} \\ &= \sum_{\alpha=n,w} \frac{\Psi_{\alpha,i}}{\rho_\alpha} \Delta x_i \end{aligned}$$

An explanation of the quantities introduced is required, $k_i = k(x_i)$, $p_i^{n+1} = p(x_i, t^{n+1})$, $p_i^{cap} = p^{cap}(S(x_i, t^n))$ and $\lambda_{\Sigma,i} = \lambda_\Sigma(S(x_i, t^n))$, analogous for the other quantities. As said before we will implement an *IMPES* scheme and for these reason

[1] *Fundamental Theorem of Calculus*

the pressure is computed implicitly. Rewriting the last equation in term of the unknown quantities, we obtain:

$$\begin{aligned}
&\left(-\frac{k_{i-1}+k_i}{2}\frac{\lambda_{\Sigma,i-1}+\lambda_{\Sigma,i}}{2}\frac{1}{\Delta x}\right)p_{i-1}^{n+1}\\
&+\left(\frac{k_{i-1}+k_i}{2}\frac{\lambda_{\Sigma,i-1}+\lambda_{\Sigma,i}}{2}\frac{1}{\Delta x}+\frac{k_{i+1}+k_i}{2}\frac{\lambda_{\Sigma,i+1}+\lambda_{\Sigma,i}}{2}\frac{1}{\Delta x}\right)p_i^{n+1}\\
&\left(-\frac{k_{i+1}+k_i}{2}\frac{\lambda_{\Sigma,i+1}+\lambda_{\Sigma,i}}{2}\frac{1}{\Delta x}\right)p_{i+1}^{n+1}\\
&=\sum_{\alpha=n,w}\frac{\Psi_{\alpha,i}}{\rho_\alpha}\Delta x_i+\frac{1}{2}\frac{\lambda_{\Delta,i+1}+\lambda_{\Delta,i}}{2}\frac{k_{i+1}+k_i}{2}\frac{p_{i+1}^{cap}-p_i^{cap}}{\Delta x}\\
&-\frac{1}{2}\frac{\lambda_{\Delta,i-1}+\lambda_{\Delta,i}}{2}\frac{k_{i-1}+k_i}{2}\frac{p_i^{cap}-p_{i-1}^{cap}}{\Delta x}
\end{aligned}$$

The complex system above can be rewritten as a linear system thanks to:

$$\begin{aligned}
a_i &= \frac{k_{i+1}+k_i}{2}\frac{\lambda_{\Sigma,i+1}+\lambda_{\Sigma,i}}{2} \qquad 1\le i\le n,\\
b_i &= \sum_{\alpha=n,w}\frac{\Psi_{\alpha,i}}{\rho_\alpha}\Delta x_i+\frac{1}{2}\frac{\lambda_{\Delta,i+1}+\lambda_{\Delta,i}}{2}\frac{k_{i+1}+k_i}{2}\frac{p_{i+1}^{cap}-p_i^{cap}}{\Delta x}\\
&\quad-\frac{1}{2}\frac{\lambda_{\Delta,i-1}+\lambda_{\Delta,i}}{2}\frac{k_{i-1}+k_i}{2}\frac{p_i^{cap}-p_{i-1}^{cap}}{\Delta x} \qquad 1<i\le n,
\end{aligned}$$

assembling now the matrix $\mathbf{A}$ as follow

$$\mathbf{A}=\begin{bmatrix}
BCs & & & & & & 0\\
-a_1 & a_1+a_2 & -a_2 & 0 & & & \\
0 & -a_2 & a_2+a_3 & -a_3 & 0 & & \\
 & \ddots & \ddots & \ddots & \ddots & & \\
 & \ddots & -a_{i-1} & a_{i-1}+a_i & -a_{i+1} & & \\
 & & \ddots & \ddots & \ddots & \ddots & \\
 & & & & -a_{n-1} & a_{n-1}+a_n & -a_n\\
0 & & & & & & BCs
\end{bmatrix}$$

we obtain the system $\mathbf{A}\mathbf{p}^{n+1}=\mathbf{b}$ with BC entrences due to the boundary conditions.

Solving the system above we obtain the new values of the unknown pressure and we can use these results in the computation of the saturation.

3.4 Saturation equation

As said before we will compute the *saturation* using an explicit method and the pressure p^{n+1} obtained above will be implemented into it.

$$\Phi\frac{\partial S_w}{\partial t}-\nabla\cdot\{\boldsymbol{\lambda}_w\mathbf{k}[\nabla(\overline{p}-\frac{1}{2}p^{cap})-\rho_w\mathbf{g}]\} \;=\; \frac{\boldsymbol{\Psi}_w}{\rho_w}$$

Such equation can be simplified using the fact that we are working in one dimension, without the influence of gravity, obtaining

$$\Phi\frac{\partial S_w}{\partial t} - \frac{\partial}{\partial x}\{\lambda_w k[\frac{\partial}{\partial x}(p - \frac{1}{2}p^{cap})]\} \;=\; \frac{\Psi_w}{\rho_w}$$

Using the same approximation techniques as before

$$\Delta x\Phi\frac{S_i^{n+1} - S_i^n}{\Delta t} - \Big[\Big(k\lambda_w\Big(\frac{\partial}{\partial x}(p - \frac{1}{2}p^{cap})\Big)\Big)_{i+1/2} - \Big(k\lambda_w\Big(\frac{\partial}{\partial x}(p - \frac{1}{2}p^{cap})\Big)\Big)_{i-1/2}\Big] = \frac{\Psi_w}{\rho_w}\Delta x$$

and further

$$\Delta x\Phi\frac{S_i^{n+1} - S_i^n}{\Delta t} - \Big(\frac{k_{i+1} - k_i}{2}\frac{\lambda_w(S_{i+1}^n) + \lambda_w(S_i^n)}{2}\Big(\frac{p_{i+1}^{n+1} - p_i^{n+1}}{\Delta x} - \frac{1}{2}\frac{p^{cap}(S_{i+1}^n) - p^{cap}(S_i^n))}{\Delta x}\Big)\Big) + \Big(\frac{k_i - k_{i-1}}{2}\frac{\lambda_w(S_i^n) + \lambda_w(S_{i-1}^n)}{2}\Big(\frac{p_i^{n+1} - p_{i-1}^{n+1}}{\Delta x} - \frac{1}{2}\frac{p^{cap}(S_i^n) - p^{cap}(S_{i-1}^n))}{\Delta x}\Big)\Big) = \frac{\Psi_w}{\rho_w}\Delta x$$

We then obtain the unknown S_i^{n+1} by simple computations

$$S_i^{n+1} \;=\; \frac{\Delta t}{\Delta x\Phi}\Big(\Big(\frac{k_{i+1} - k_i}{2}\frac{\lambda_w(S_{i+1}^n) + \lambda_w(S_i^n)}{2}\Big(\frac{p_{i+1}^{n+1} - p_i^{n+1}}{\Delta x} - \frac{1}{2}\frac{p^{cap}(S_{i+1}^n) - p^{cap}(S_i^n)}{\Delta x}\Big)\Big) + \Big(\frac{k_i - k_{i-1}}{2}\frac{\lambda_w(S_i^n) + \lambda_w(S_{i-1}^n)}{2}\Big(\frac{p_i^{n+1} - p_{i-1}^{n+1}}{\Delta x} - \frac{1}{2}\frac{p^{cap}(S_i^n) - p^{cap}(S_{i-1}^n)}{\Delta x}\Big)\Big)\frac{F_{w,i}^{n+1}}{\rho_w}\Delta x\Big) + S_i^n = \frac{\Psi_w}{\rho_w}\Delta x$$

3.5 Transport equation

Thanks to the methods shown in the two sections above we have been able to implement an *IMPES* scheme to solve our coupled problem. Our final aim is now to use such results in the resolution of the *transport equation*:

$$\Phi\frac{\partial S_w c}{\partial t} + \nabla\cdot(-\Phi D S_w\nabla(c) + \mathbf{u}_w c) \;=\; R(c) + f(x,t). \tag{3.12}$$

Same techniques as before will be implemented and a *fully implicit scheme* will be obtained. Starting from the equation above we express it in the 1D case

$$\Phi\frac{\partial S_w c}{\partial t} + \frac{\partial}{\partial x}(-\Phi D S_w\frac{\partial c}{\partial x} + u_w c) \;=\; R(c) + f(x,t). \tag{3.13}$$

Let's assume we have an expression for the reaction term $R(c) = ac(x,t)$ where $a \in \mathbb{R}$, different formulations will be presented later. We start discretizing the time derivative and the first space derivative, obtaining

$$\Phi \frac{S_i^{n+1}c_i^{n+1} - S_i^n c_i^n}{\Delta t} - \left(\left(\Phi D S \frac{\partial c}{\partial x} - u_w c \right)_{i+1/2} - \left(\Phi D S \frac{\partial c}{\partial x} - u_w c \right)_{i-1/2} \right) \frac{1}{\Delta x} = \; a c_i^{n+1} + f_i^n$$

and with a further step

$$\Phi \frac{S_i^{n+1}c_i^{n+1} S_i^n c_i^n}{\Delta t} - \frac{1}{\Delta x} \{ \Phi D \frac{S_{i+1}^{n+1} + S_i^{n+1}}{2} \frac{c_{i+1}^{n+1} - c_i^{n+1}}{\Delta x} - u_{w,i+1/2}^{n+1} \frac{c_{i+1}^{n+1} + c_i^{n+1}}{2} + - \Phi D \frac{S_i^{n+1} + S_{i-1}^{n+1}}{2} \frac{c_i^{n+1} - c_{i-1}^{n+1}}{\Delta x} + u_{w,i-1/2}^{n+1} \frac{c_i^{n+1} + c_{i-1}^{n+1}}{2} \} = \; a c_i^{n+1} + f_i^{n+1}$$

where $u_{w,i+1/2}^{n+1}$ represents $-\frac{1}{\Delta x}(p_{i+1}^{n+1} - \frac{1}{2}p^{cap}(S_{i+1}^{n+1}) - p_i^{n+1} + \frac{1}{2}p^{cap}(S_i^{n+1}))$ and, rearranging the terms in function of the unknowns, we obtain

$$c_{i-1}^{n+1} \underbrace{\left(-\Phi D \Delta t \frac{S_i^{n+1} + S_{i-1}^{n+1}}{2} - \Phi \frac{\Delta t \Delta x}{2} u_{w,i-1/2}^{n+1} \right)}_{\alpha_i} +$$

$$+ c_i^{n+1} \underbrace{\left(-\Phi \; \Delta x^2 (2S_i^{n+1} - S_i^n) + \Delta t D \frac{2S_i^{n+1} + S_{i-1}^{n+1} + S_{i+1}^{n+1}}{2} \right.}_{\beta_i}$$

$$\underbrace{\left. + \Delta t \Delta x \frac{w_{w,i+1/2}^{n+1} - u_{w,i-1/2}^{n+1}}{2} - a \right)}_{\beta_i}$$

$$+ c_{i+1}^{n+1} \underbrace{\left(-\Phi D \Delta t \frac{S_i^{n+1} + S_{i+1}^{n+1}}{2} + \Phi \frac{\Delta t \Delta x}{4} u_{w,i+1/2}^{n+1} \right)}_{\gamma_i}$$

$$= \underbrace{\frac{S_i^n c_i^n}{\Delta t} \Phi + f_i^n}_{g_i}$$

As in the case of the *pressure equation* we can reduce the complex equation above into a linear system of the form $\mathbf{B}\mathbf{c}^{n+1} = \mathbf{g}$ with $\mathbf{B}$ given by

$$\mathbf{B} = \begin{bmatrix} BCs & & & & & & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & 0 & & & \\ 0 & \alpha_3 & \beta_3 & \gamma_3 & 0 & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & \ddots & \alpha_i & \beta_i & \gamma_i & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & & \alpha_N & \beta_N & \gamma_N \\ 0 & & & & & & BCs \end{bmatrix}$$

Chapter 4

Numerical model for Richards and transport equations

4.1 Introduction

In this chapter we will move our attention to a particular case of *Two-Phase Flows*, we will study underground water flows. Such phenomena are described by the *Richards* equation. With underground water flow is intended a multiphase flow in porous media in which both water phase and air phase coexist.

The formulation obtained before, in the previous chapters, is still valid but it can be simplified, we assume the air pressure to be equal to the *atmospheric pressure*. The system of equations is reduced to an equation regarding the saturation of the water, such equation is known as *Richards equation*. We will also study a reactive transport, investigating how the presence of the external component can influence the flow, in particular we will concentrate on the changes in the surface tension. Before to write the expression for such equations we have to introduce a couple of quantities, the *water content* θ and the *head pressure* Ψ [1]

$$\theta = S_w * \Phi$$
$$\Psi = \frac{2\gamma}{\rho g r_m}$$

where γ is the *surface tension* of the water, g the acceleration gravity, ρ the density and r_m is the mean radius of curvature of the surface tension. Thanks to this expression we can investigate the influence also of the gravity. Such dependence was neglected in the first part of the thesis, where we studied horizontal flows, now we will instead investigate a one dimensional problem in the vertical direction where g plays an important role.

These two quantities have been defined because we will present a formulation slightly different from the one given in the previous part of this work but closer to the literature consulted [1]. The radius r_m introduced above is a function of the water content and it relates between each others: Ψ, γ and the water content θ.

The water content θ is commonly related to the pressure head through the van

Genuchten [31] equation:

$$\theta = \frac{\theta_s - \theta_r}{(1+ \mid \alpha\Psi \mid^n)^m} + \theta_r$$

with θ_s and θ_r respectively the saturated and residual volumetric water content, α, n and m fitting parameters with $m = 1 - 1/n$.

Many articles have been written investigating how an external component influences the flow of the phase, in particular, in this work, we will concentrate on the changes in the surface tension. If such quantity is not constant the pressure head Ψ must be rescaled [1] as follow:

$$\tilde{\Psi} = \frac{\gamma_0(c_0)}{\gamma(c)}\Psi$$

where γ_0 is the reference surface tension at the reference concentration c_0 and γ the tension at the concentration c. Thanks to this rescaling, the water content, usually expressed as function of only Ψ, becomes now a function of both pressure and concentration

$$\theta(\Psi, c) = \frac{\theta_s - \theta_r}{(1+ \mid \alpha\frac{\gamma_0(c_0)}{\gamma(c)}\Psi \mid^n)^m} + \theta_r \tag{4.1}$$

Using again the relations developed in [31] we can obtain an expression for the unsaturated hydraulic conductivity

$$K(\Theta) = K_s\Theta^l(1 - (1 - \Theta^{l/m})^m)^2$$

with K_s the saturated conductivity, l and m fitting parameters and Θ the effective permeability defined as

$$\Theta = \frac{\theta - \theta_r}{\theta_s - \theta_r}$$

Given the Darcy velocity for vertical unsaturated flux

$$u = K(\theta)(\frac{\partial\Psi}{\partial x} + 1)$$

we can obtain, from the *mass conservation* equation, the *Richards equation*:

$$\frac{\partial\theta(\Psi, c)}{\partial t} - (\frac{\partial}{\partial x}K(\theta(\Psi, c))(\frac{\partial\Psi}{\partial x} + 1)) = H(x, t) \tag{4.2}$$

Such equation can be easily coupled with the transport described by:

$$\frac{\partial(\theta(\Psi, c)c)}{\partial t} + \frac{\partial}{\partial x}(-\theta(\Psi, c)D\frac{\partial c}{\partial x} + u_w * c) = H_c(x, t) \tag{4.3}$$

4.2 Linearization methods

The difficult aspect in treating the equations above is not only in the double dependence of θ and K form Ψ and c but also in the non linearity of the these terms. In this section we will analyze the most common linearization methods and we will compare them. We will present and then implement in Matlab, the following schemes: *L-scheme*, *Newton method* and *modified Picard*.

To study the non linearity of θ and K a new iteration $j \in \mathbb{N}$ must be introduced, we will use the notation $\Psi^{n+1,j}$ to indicate the fixed solution at the time step $n+1$ and iteration j. Such iteration, common to all the linearization schemes, will start using the solution computed at the previous step, precisely $\Psi^{n+1,1} = \Psi^n$.

4.2.1 L-scheme

The *L-scheme* is extremely common due to its simplicity, it doesn't compute in fact any derivative and it reads as follow. Given the equation 4.2 we can rewrite the time derivative as

$$\begin{aligned} &\theta(\Psi^{n+1,j}, c^n) - \theta(\Psi^n, c^n) + L_1(\Psi^{n+1,j+1} - \Psi^{n+1,j}) \\ &- \Delta t \frac{\partial}{\partial x}(K(\Psi^{n+1,j}, c^{n+1,j})(\frac{\partial}{\partial x}\Psi^{n+1,j+1} + 1)) = \Delta t H(x,t) \end{aligned} \tag{4.4}$$

where L_1 is a constant, verifying $L_1 \geq max\|\frac{\partial \theta}{\partial \Psi}\|$ and the quantity $(\Psi^{n+1,j+1} - \Psi^{n+1,j})$ goes to zero for $j = 1, 2, \dots$

Analogous for the transport equation 4.3, computing the time derivative of the product

$$\begin{aligned} &[\theta(\Psi^{n+1,j+1}, c^{n+1,j}) - \theta(\Psi^n, c^n) + L_2(c^{n+1,j+1} - c^{n+1,j})] * c^{n+1,j} \\ &+ \theta(\Psi^{n+1,j+1}, c^{n+1,j}) * (c^{n+1,j+1} - c^n) + \Delta t \frac{\partial}{\partial x}(-\theta(\Psi^{n+1,j+1}, c^{n+1,j}) D \frac{\partial c^{n+1,j+1}}{\partial x} \\ &+ u_w^{n+1,j} * c^{n+1,j+1}) = \Delta t H_c(x,t) \end{aligned} \tag{4.5}$$

$L_2 \geq max\|\frac{\partial \theta}{\partial c}\|$ and again $(c^{n+1,j+1} - c^{n+1,j})$ goes to zero for $j = 1, 2, \dots$. We built a while loop that will stop whenever the condition $\|\Psi^{n+1,j+1} - \Psi^{n+1,j}\| \leq \epsilon$ and $\|c^{n+1,j+1} - c^{n+1,j}\| \leq \epsilon$ are satisfied, with ϵ a known constant. Later in the work we will show some numerical results to study the convergence of this method.

4.2.2 Newton monolithic method

The Newton Monolithic method is remarkably fast and for this reason it is often used, at the same time it has an evident disadvantage, it requires the computation of the *Jacobian* matrix. For any system of equations, the Newton method can be implemented as follow. Given the multilinear system

$$\mathbf{K}(\mathbf{v}) = \mathbf{F}$$

we can rewrite it as

$$\mathbf{r}(\mathbf{v}) = \mathbf{K}(\mathbf{v}) - \mathbf{F}$$

and proceed to search the zeros of the function $\mathbf{r}(\mathbf{v})$. We can now present the Newton method as explained in [10]. Suppose ξ a solution of $\mathbf{r}$ which is differentiable

in an interval $I(\xi)$. It is then possible to write the Taylor expansion $\mathbf{r}$ around $\mathbf{v}_0$, a point in $I(\xi)$

$$\mathbf{r}(\xi) = 0 = \mathbf{r}(\mathbf{v}_0 + (\xi - \mathbf{v}_0)) = \mathbf{r}(\mathbf{v}_0) + D\mathbf{r}(\mathbf{v}_0)(\xi - \mathbf{v}_0) + D^2\mathbf{r}(\mathbf{v}_0)\frac{(\xi - \mathbf{v}_0)^2}{2!} + \dots$$

Neglecting the term of second or higher order we obtain an estimation for the root of $\mathbf{r}$

$$\overline{\xi} \approx \mathbf{v}_0 - J^{-1}\mathbf{r}(\mathbf{v}_0)\mathbf{r}(u)_0$$

where J is the Jacobian matrix computed as

$$J\mathbf{r}(\mathbf{v}_0) = \frac{\partial \mathbf{r}_i}{\partial \mathbf{v}_j}_{\mathbf{v}=\mathbf{v}_0}$$

and

$$\frac{\partial \mathbf{r}_i}{\partial \mathbf{v}_j}_{\mathbf{v}=\mathbf{v}_0} = \begin{bmatrix} \frac{\partial \mathbf{r}_1}{\partial \mathbf{v}_1} \frac{\partial \mathbf{r}_1}{\partial \mathbf{v}_2} \cdots \frac{\partial \mathbf{r}_1}{\partial \mathbf{v}_n} \\ \frac{\partial \mathbf{r}_2}{\partial \mathbf{v}_1} \frac{\partial \mathbf{r}_2}{\partial \mathbf{v}_2} \cdots \frac{\partial \mathbf{r}_2}{\partial \mathbf{v}_n} \\ \vdots \ddots \ddots \vdots \\ \frac{\partial \mathbf{r}_n}{\partial \mathbf{v}_1} \frac{\partial \mathbf{r}_n}{\partial \mathbf{v}_2} \cdots \frac{\partial \mathbf{r}_n}{\partial \mathbf{v}_n} \end{bmatrix}_{\mathbf{v}=\mathbf{v}_0} \tag{4.6}$$

The approximation $\overline{\xi}$ should be close to the real root. From the expression above we can build the sequence

$$\mathbf{v}^{n+1} = \mathbf{v}^n - J^{-1}(\mathbf{r}(\mathbf{v}^n))\mathbf{r}(\mathbf{v}^n) \tag{4.7}$$

For our particular problem we have $\mathbf{v} = [\Psi, c]$, $\mathbf{F} = [H, H_c]$, $\mathbf{r}$ is expressed as

$$r_1 = \frac{\partial \theta}{\partial t} - \frac{\partial}{\partial x}(K(\Psi, c)(\frac{\partial \Psi}{\partial x} + 1)) - H(x,t)$$
$$r_2 = \frac{\partial \theta(\Psi, c)c}{\partial t} + \frac{\partial}{\partial x}(-\theta(\Psi, c)D\frac{\partial c}{\partial x} + u_w * c) - H_c(x,t)$$

and J

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{r}_1}{\partial \mathbf{\Psi}} \frac{\partial \mathbf{r}_1}{\partial \mathbf{c}} \\ \frac{\partial \mathbf{r}_2}{\partial \mathbf{\Psi}} \frac{\partial \mathbf{r}_2}{\partial \mathbf{c}} \end{bmatrix}$$

Such method is defined *monolthic* because we are considering the two equations, Richards and transport, as a system of the form

$$F_1(\Psi^{n+1,j+1}, c^{n+1,j+1}) = H(x,t)$$
$$F_2(\Psi^{n+1,j+1}, c^{n+1,j+1}) = H_c(x,t)$$

4.2.3 Newton method

In this section we will briefly present a *non*-monolithic approach to the Newton method. The two equations: Richards and transport will be solve separately, considering, each time only the dependence from one of the two unknowns. Precisely

we will solve the Richards equation assuming θ of being a function of only Ψ, the dependence from c will be instead studied in the transport equation.

Recalling the formulation used above for the monolithic approach we can write, respectively for the Richards and transport equations:

$$\Psi^{n+1} = \Psi^n - J_\Psi^{-1}(r_\Psi(\Psi^n, c^n))r_\Psi(\Psi^n, c^n) \tag{4.8}$$

$$c^{n+1} = c^n - J_c^{-1}(r_c(\Psi^n, c^n))r_c(\Psi^n, c^n) \tag{4.9}$$

where J_Ψ and J_c are the Jacobian matrices related, respectively, to the Richards and the transport equations while r_Ψ and r_c are the two expressions them self.

4.2.4 Modified Picard

In this section we will describe the *modified Picard* method introduce by Celia [11] in 1990. The linearization of the θ and K is obtained using the chain rule.

Precisely we will write the equations as

$$\begin{aligned}&\theta(\Psi^{n+1,j}, c^n) - \theta(\Psi^n, c^n) + \theta_\Psi(\Psi^{n+1,j}, c^{n+1,j})(\Psi^{n+1,j+1} - \Psi^{n+1,j}) \\ &- \Delta t \frac{\partial}{\partial x}(K(\Psi^{n+1,j}, c^{n+1,j})(\frac{\partial}{\partial x}\Psi^{n+1,j+1} + 1)) = \Delta t H(x,t)\end{aligned} \tag{4.10}$$

and

$$\begin{aligned}&[\theta(\Psi^{n+1,j+1}, c^{n+1,j}) - \theta(\Psi^n, c^n) + \theta_c(\Psi^{n+1,j}, c^{n+1,j})(c^{n+1,j+1} - c^{n+1,j})] * c^{n+1,j} \\ &+ \theta(\Psi^{n+1,j+1}, c^{n+1,j}) * (c^{n+1,j+1} - c^n) + \Delta t \frac{\partial}{\partial x}(-\theta(\Psi^{n+1,j+1}, c^{n+1,j}) D \frac{\partial c^{n+1,j+1}}{\partial x} \\ &+ u_w^{n+1,j} * c^{n+1,j+1}) = \Delta t H_c(x,t)\end{aligned} \tag{4.11}$$

For the Richards equation we have considered the dependence of θ only from Ψ, we have done the opposite in the transport, where the dependence from c has been used.

4.2.5 Monolithic Picard scheme

We will now implement the *Monolithic-Picard scheme*, it treats the equations 4.2 and 4.3 exactly as in the case of the *monolithic Newton* method, we consider them as a system of the form:

$$\begin{aligned}F_1(\Psi^{n+1,j+1}, c^{n+1,j+1}) &= H(x,t) \\ F_2(\Psi^{n+1,j+1}, c^{n+1,j+1}) &= H_c(x,t)\end{aligned}$$

the main difference from the Picard scheme described above is that, in this case we will investigate the double dependence of θ from both Ψ and c computing both

of the derivatives in each equation, precisely the quantities F_1 and F_2 are expressed as

$$\begin{aligned}
F_1 =&\theta(\Psi^{n+1,j}, c^n) - \theta(\Psi^n, c^n) + \theta_\Psi(\Psi^{n+1,j}, c^{n+1,j})(\Psi^{n+1,j+1} - \Psi^{n+1,j}) \\
&+ \theta_c(\Psi^{n+1,j}, c^{n+1,j})(c^{n+1,j+1} - c^{n+1,j}) - \Delta t \frac{\partial}{\partial x} \\
&(K(\Psi^{n+1,j}, c^{n+1,j})(\frac{\partial}{\partial x}\Psi^{n+1,j+1} + 1)) \\
F_2 =&[\theta(\Psi^{n+1,j+1}, c^{n+1,j}) - \theta(\Psi^n, c^n) + \theta_c(\Psi^{n+1,j}, c^{n+1,j})(c^{n+1,j+1} - c^{n+1,j}) \\
&+ \theta_\Psi(\Psi^{n+1,j}, c^{n+1,j})(\Psi^{n+1,j+1} - \Psi^{n+1,j})] * c^{n+1,j} + \theta(\Psi^{n+1,j+1}, c^{n+1,j}) \\
&* (c^{n+1,j+1} - c^n) + \Delta t \frac{\partial}{\partial x}(-\theta(\Psi^{n+1,j+1}, c^{n+1,j}) D \frac{\partial c^{n+1,j+1}}{\partial x} + u_w^{n+1,j} * c^{n+1,j+1})
\end{aligned}$$

The disadvantage of this method is the creation of a massive matrix A and this can give instability and high computational times to the scheme. Such matrix A has the following form

$$A = \begin{bmatrix} BD_c \\ D_\Psi C \end{bmatrix}$$

B and C are two banded matrices similar to the ones obtained in the previous schemes, D_Ψ and D_c are instead two diagonal matrices obtained respectively from the dependence of the Richards equation from c^{n+1} and from the dependence of the transport equation from Ψ^{n+1}.

4.2.6 Monolithic L-scheme

As for the method above a system of the same form will be created, the linearization method used in this case will be the L-scheme. Precisely we will obtain the following expressions for the quantities F_1 and F_2

$$\begin{aligned}
F_1 =&\theta(\Psi^{n+1,j}, c^n) - \theta(\Psi^n, c^n) + L_1(\Psi^{n+1,j+1} - \Psi^{n+1,j}) + L_2(c^{n+1,j+1} - c^{n+1,j}) \\
&- \Delta t \frac{\partial}{\partial x}(K(\Psi^{n+1,j}, c^{n+1,j})(\frac{\partial}{\partial x}\Psi^{n+1,j+1} + 1)) \\
F_2 =&[\theta(\Psi^{n+1}, c^{n+1,j}) - \theta(\Psi^n, c^n) + L_1(\Psi^{n+1,j+1} - \Psi^{n+1,j}) + L_2(c^{n+1,j+1} - c^{n+1,j})] * c^{n+1,j} \\
&+ \theta * (c^{n+1,j+1} - c^n) + \Delta t \frac{\partial}{\partial x}(-\theta(\Psi^{n+1,j+1}, c^{n+1,j}) D \frac{\partial c^{n+1,j+1}}{\partial x} + u_w^{n+1,j} * c^{n+1,j+1})
\end{aligned}$$

again a matrix of the form

$$A = \begin{bmatrix} BD_c \\ D_\Psi C \end{bmatrix}$$

where B and C are two banded matrices while D_Ψ and D_c are two diagonal matrices.

In all of the schemes above, the space derivatives will be computed using the two point flux approximation as in the first part of the thesis.

4.3 Convergence

In this section we will prove the convergence of these linearization schemes, we will concentrate on the *L-scheme* but similar proves will be valid for the other methods, we will also, for simplicity, neglect the double dependence of θ and K.

Many article have been written covering this subject and we will consider [15] as guide for the following prove.

4.3.1 Richards equation

Let's start studying the Richards equation, first of all we must define the following quantity

$$e^{n+1,j+1} = \Psi^{n+1,j+1} - \Psi^{n+1}$$

representing the error at the $j+1$ iteration, the scheme converges if $e^{n+1,j+1} \to 0$ as $j \to \infty$.
Some conditions are required on the functions present in our problem

- θ must be monotonically increasing and Lipschitz continuous
- K is also Lipschitz continuous and there exist two constants K_m and K_M such that $0 < K_m \leqslant K(\theta) \leqslant K_M < \infty$
- the solution to the problem satisfies $\| \nabla\Psi^{n+1} \|_\infty \leqslant M < \infty$

We can now rewrite the Richards equation using the *Finite Element Formulation*, we will try to find a $\Psi \in H_0^1([0,1])$ such that

$$< \partial_t\theta(\Psi), \Phi > + < K(\theta(\Phi))\nabla\Psi, \nabla\Phi >=< H, \Phi > \tag{4.12}$$

holds for all $\Phi \in H_0^1([0,1])$, where $H_0^1([0,1])$ is the space of the Φ such that Φ is real valued, square integrable, its first derivative is also square integrable and $\Phi(0) = \Phi(1) = 0$. We won't analyse the finite element method, many books [24] have been written on the subject.

After the discretization in time 4.12 becomes

$$< \theta(\Psi^{n+1}) - \theta(\Psi^n), v_h > +\tau < K(\theta(\Psi^{n+1}))(\nabla(\Psi^{n+1}) + 1), \nabla v_h >= \tau < H, v_h > \tag{4.13}$$

where τ represents the time step Δt, $v_h \in V_h$ with V_h *Galerkin* finite element space, again consult [24] for further clarifications.
Implementing now the *L-scheme* into the equation 4.13

$$\begin{aligned} &< \theta(\Psi^{n+1,j}) - \theta(\Psi^n), v_h > +L < \Psi^{n+1,j+1} - \Psi^{n+1,j}, v_h > + \\ &+\tau < K(\theta(\Psi^{n+1,j}))(\nabla(\Psi^{n+1,j+1}) + 1), \nabla v_h >= \tau < H, v_h > \end{aligned} \tag{4.14}$$

with L the positive constant characterizing the scheme.
Subtracting now 4.13 from 4.14 one obtains

$$\begin{aligned} &< \theta(\Psi^{n+1,j}) - \theta(\Psi^{n+1}), v_h > +L < \Psi^{n+1,j+1} - \Psi^{n+1,j}, v_h > + \\ &+ \tau < K(\theta(\Psi^{n+1,j}))\nabla(\Psi^{n+1,j+1}) - K(\theta(\Psi^{n+1}))\nabla(\Psi^{n+1}), \nabla v_h >= 0 \end{aligned} \tag{4.15}$$

substituting v_h with $e^{n+1,j+1}$ and with some simple manipulations

$$\begin{aligned}
&< \theta(\Psi^{n+1,j}) - \theta(\Psi^{n+1}), e^{n+1,j+1} > + L < e^{n+1,j+1} - e^{n+1,j}, e^{n+1,j+1} > + \\
&+ \tau < K(\theta(\Psi^{n+1,j}))\nabla e^{n+1,j+1}), \nabla e^{n+1,j+1} > + \tau < (K(\theta(\Psi^{n+1,j})) + \\
&- K(\theta(\Psi^{n+1})))\nabla\Psi^{n+1}, \nabla e^{n+1,j+1} >= 0
\end{aligned} \tag{4.16}$$

Using now the property of Lipschitz continuity required for the functions θ and K, the boundedness of K and $\nabla\Psi^{n+1}$, the *parallelogram* rule, and the *Young* and *Causchy-Schwartz* inequalities one obtains, after rearranging the terms

$$\begin{aligned}
&\frac{1}{L_\theta} \parallel \theta(\Psi^{n+1,j}) - \theta(\Psi^{n+1}) \parallel^2 + \frac{L}{2} \parallel e^{n+1,j+1} \parallel^2 + \tau K_m \parallel \nabla e^{n+1,j+1} \parallel^2 \\
&\leqslant \frac{L}{2} \parallel e^{n+1,j} \parallel^2 + \frac{1}{2L} \parallel \theta(\Psi^{n+1,j}) - \theta(\Psi^{n+1}) \parallel^2 + \\
&- \tau\frac{M}{2k_m} \parallel \theta(\Psi^{n+1,j}) - \theta(\Psi^{n+1}) \parallel^2 - \tau\frac{K_m}{2} \parallel \nabla e^{n+1,j+1} \parallel^2
\end{aligned} \tag{4.17}$$

where L_θ is the Lipschitz constant for the function θ and the Young inequality has been used twice with constants L and K_m.
The equation above can be rewritten as

$$\begin{aligned}
&L \parallel e^{n+1,j+1} \parallel^2 + (\tau M + \frac{1}{L_\theta}) \parallel \theta(\Psi^{n+1,j}) - \theta(\Psi^{n+1}) \parallel^2 + \\
&+ 3\tau K_m \parallel \nabla e^{n+1,j+1} \parallel^2 \leqslant L \parallel e^{n+1,j} \parallel^2
\end{aligned} \tag{4.18}$$

observing that the quantity $\tau M + \frac{1}{L_\theta} \geqslant 0$ and using the *Poincare* inequality

$$\frac{1}{\alpha^2} \parallel e^{n+1,j+1} \parallel^2 \leqslant \parallel \nabla e^{n+1,j+1} \parallel^2$$

α a constant depending on the domain $[0, 1]$

$$(L + \frac{3\tau K_m}{\alpha^2}) \parallel e^{n+1,j+1} \parallel^2 \leqslant L \parallel e^{n+1,j} \parallel^2 \tag{4.19}$$

and finally the result

$$\parallel e^{n+1,j+1} \parallel \leqslant \sqrt{\frac{L}{L + \frac{3\tau K_m}{\alpha^2}}} \parallel e^{n+1,j} \parallel \tag{4.20}$$

We have in this way proved that the L-scheme converges linearly, with a rate of convergence given by:

$$\sqrt{\frac{L}{L + \frac{3\tau K_m}{\alpha^2}}}$$

4.3.2 Transport equation

We will now prove, as in the section above, the convergence of the *L-scheme* applied to the transport equation decoupled from the Richards. To consider the

problem separately, as said before, we must assume that θ and K are functions of only c.

We will use again the *finite element formulation*, starting from

$$< \partial_t(\theta(c)c), \Phi > + < K(\theta(c))D\nabla\Psi - u_w c, \nabla\Phi >=< H_c, \Phi > \tag{4.21}$$

$\Phi \in H_0^1$, discretizing now in time

$$\begin{aligned} &< \theta(c^{n+1}c^{n+1} - \theta(c^n)c^n), v_h > + < K(\theta(c^{n+1}))D\nabla\Psi - u_w c^{n+1}, \nabla v_h > \\ &=< H_c, v_h > \end{aligned} \tag{4.22}$$

where $v_h \in V_h$.

Using the *L-scheme* linearization

$$\begin{aligned} &< \theta(c^{n+1,j}c^{n+1,j+1} - \theta(c^n)c^n), v_h > + L < c^{n+1,j+1} - c^{n+1,j}, v_h > + \\ &+ \tau < K(\theta(c^{n+1,j}))D\nabla c^{n+1,j+1} - u_w^{n+1,j}c^{n+1,j+1}, \nabla v_h >= \tau < H_c, v_h > \end{aligned} \tag{4.23}$$

At this point we must define the new quantity e

$$e^{n+1,j+1} = c^{n+1,j+1} - c^{n+1}$$

and require that:

- θ is monotonically increasing and Lipschitz continuous
- K and u_w are also Lipschitz continuous and there exist six constants K_m, K_M, D_m, D_M, $u_{w,m}$ and $u_{w,M}$ such that $0 < K_m \leqslant K(\theta) \leqslant K_M < \infty$, $0 < D_m \leqslant D \leqslant D_M < \infty$ and $0 < u_{w,m} \leqslant u_w(x,t) \leqslant u_{w,M} < \infty$
- the solution to the problem satisfies: $\| \nabla c^{n+1} \|_\infty \leqslant M < \infty$ and $0 \leqslant c_m \leqslant \| c^{n+1} \|_\infty$

As before we want to prove that $e^{n+1,j+1} \to 0$ as $j \to \infty$.

Subtracting 4.22 from 4.23 and substituting v_h with $e^{n+1,j+1}$, one obtains

$$\begin{aligned} &< \theta(c^{n+1,j})c^{n+1,j+1} - \theta(c^{n+1})c^{n+1}, e^{n+1,j+1} > + L < e^{n+1,j+1} - e^{n+1,j}, e^{n+1,j+1} > + \\ &+ \tau < K(\theta(c^{n+1,j}))D\nabla c^{n+1,j+1} - K(\theta(c^{n+1}))\nabla c^{n+1} + \\ &- u_w^{n+1,j}(c^{n+1,j+1} - c^{n+1}), \nabla e^{n+1,j+1} >= 0 \end{aligned} \tag{4.24}$$

thanks to further computations

$$\begin{aligned} &< \theta(c^{n+1,j})e^{n+1,j+1}, e^{n+1,j+1} > + < (\theta(c^{n+1,j}) - \theta(c^{n+1}))c^{n+1} > + \\ &+ L < e^{n+1,j+1} - e^{n+1,j}, e^{n+1,j+1} > + \tau < K(\theta(c^{n+1,j}))D\nabla e^{n+1,j+1}, \nabla e^{n+1,j+1} > + \\ &+ \tau < (K(\theta(c^{n+1,j})) - K(\theta(c^{n+1})))D\nabla c^{n+1}, \nabla e^{n+1,j+1} > \\ &- \tau < u_w^{n+1,j}e^{n+1,j+1}, \nabla e^{n+1,j+1} >= 0 \end{aligned} \tag{4.25}$$

Using the property of our functions as Lipschitz continuity and boundedness and the same Young, Causchy-Schwartz and Poincare inequalities:

$$\begin{aligned} &(\theta_m + \frac{L}{2} - \tau\alpha u_{w,M}) \| e^{n+1,j+1} \|^2 + (\frac{c_m}{L_\theta} + \frac{\tau M}{2K_m D_m}) \| \theta(c^{n+1,j}) - \theta(c^{n+1}) \|^2 + \\ &+ \frac{3\tau K_m D_m}{2} \| \nabla e^{n+1,j+1} \|^2 \leqslant \frac{L}{2} \| e^{n+1,j} \|^2 \end{aligned} \tag{4.26}$$

where θ_m is the lower bound for θ, α the Poincare constant, c_m the lower bound for c and L_θ the Lipschitz constant. Observing now that $\frac{c_m}{L_\theta} + \frac{\tau M}{2K_m D_m} \geqslant 0$ and applying again the Poincare inequality

$$\| e^{n+1,j+1} \|^2 \leqslant \frac{L}{2\theta_m + L - 2\tau u_{w,M}\alpha + \frac{3\tau K_m D_m}{\alpha}} \| e^{n+1,j} \|^2 \tag{4.27}$$

we can now simply obtain the final result

$$\| e^{n+1,j+1} \| \leqslant \sqrt{\frac{L}{2\theta_m + L - 2\tau u_{w,M}\alpha + \frac{3\tau K_m D_m}{\alpha}}} \| e^{n+1,j} \| \tag{4.28}$$

We have proved that the L-scheme converges linearly, with a rate of convergence given by:

$$\sqrt{\frac{L}{2\theta_m + L - 2\tau u_{w,M}\alpha + \frac{3\tau K_m D_m}{\alpha}}}$$

Chapter 5

Code validation through numerical examples

In this chapter we will try to validate the solving algorithms defined in the previous chapters. We will first concentrate on the *IMPES* and its *fully implicit* reformulation, developed for the two-phase flow and transport, after that we will concentrate on the linearization methods built for the *Richards* and transport equations.

To do that we need to define a problem for which we can compute an analytical solution and then compute the *error* between the numerical and the analytical results. There exists many types of error, in this work we will use the *absolute error*, defined as the magnitude of the difference between the numerical and the analytical solutions. Again, many options are presented in the choice of the norm used to compute such magnitude, we will use the *eucledian norm*.

We will also observe that the models defined above have precise *rates* and *orders* of convergence. The rate of convergence is defined as the ratio between the errors e_n/e_{n+1} where $e_n = ||u_{anal,n} - u_{num,n}||_2$. The order of convergence is instead the number p such that $||u_{anal,n} - u_{num,n}||_2 < c\Delta x^p$ with c a positive constant.

5.1 Two-phase flow

The idea is now to find a function that can be used as analytical solution of our problem, we will require that, such function satisfies particularly easy boundary conditions. An example in the 1-dimensional case can be the function
$f(x,t) = xt(1-x)$. Such function on the domain $D = [0,1] \times \mathbb{R}^+$ has initial and boundary conditions of the form $f(x,0) = 0$ and $f(0,t) = f(1,t) = 0$.

Let's require that both the pressure $\overline{p}$ and the saturation S have the same expression and precisely they are equal to the function defined above. We need now to compute the external forces used in the *pressure* and *saturation* equations using the new expression for $\overline{p}$ and S. All the other quantities will be defined in the way to obtain easier computations, precisely we will impose $\lambda_w = 1$, $\lambda_o = 2$, $k = 1$, $\Phi = 1$, $\rho_w = 1$ and $\rho_n = 1$. These quantities can be unrealistic but that is not our concern, as said before we are trying to verify that our code works, we are not, at the moment, interested in a simulation of a realistic physical problem.

Last quantity to define is the *capillary pressure*, we will express it as a non-linear

function of the *saturation*. For simplicity in the computation we define $p^{cap} = 1 - \frac{1}{2}S^2$ even if it can be an unrealistic value.
We have now all the quantities required to compute the external forces, precisely from 2.23 and 2.24

$$\Psi_w = \rho_w(\Phi\frac{\partial S}{\partial t} - \frac{\partial}{\partial x}(\lambda_w k\frac{\partial(p - \frac{1}{2}p^{cap})}{\partial x}))$$
$$\Psi_n = \rho_n(-\frac{F_w}{\rho_w} + \frac{\partial}{\partial x}(-k(\lambda_\Sigma\frac{\partial p}{\partial x} + \frac{\lambda_\Delta}{2}\frac{\partial p^{cap}}{\partial x})))$$

We will now compute such quantities using the parameters defined above

$$\Psi_w = 6x^2t^2 - 6xt^2 - x^2 + x + t^2 + 2t^2$$
$$\Psi_n = -12x^2t^2 + 12xt^2 + x^2 - x - 2t^2 + 4t$$

Whit these new external forces we can redefine the equations 2.23 and 2.24 obtaining a system for which we already have an analytical solution.

IMPES

We will now solve the new system obtained using the *IMPES* method defined in the previous sections. Remembering that the domain of our problem is $D = [0,1] \times [0, t_{max}]$ where $t_{max} = 1$ and using the quantities defined above we can solve the system given by

$$\Phi\frac{\partial S_w}{\partial t} - \nabla \cdot \{\boldsymbol{\lambda}_w \mathbf{k}[\nabla(\overline{p} - \frac{1}{2}p^{cap}) - \rho_w \mathbf{g}]\} = \frac{\boldsymbol{\Psi}_w}{\rho_w}$$
$$\nabla \cdot (\mathbf{u}_w + \mathbf{u}_n) = \nabla \cdot \mathbf{u}_\Sigma = \sum_{\alpha=n,w}\frac{\boldsymbol{\Psi}_\alpha}{\rho_\alpha}$$

obtaining the results expressed in the figures 5.1 and 5.2 and in the table 5.1.

Table 5.1: Computational errors and Orders of Convergence for Pressure and Saturation

$\Delta x = \Delta t$	Pressure er	Saturation er	r_p	r_S	o_p	o_S
0.1	0.00018007	0.00019362	-	-	-	-
0.05	4.5013e-05	4.7459e-05	4.0004	4.0797	2.0002	2.0285
0.025	1.1253e-05	1.176e-05	4.0001	4.0357	2.0000	2.0128
0.0125	2.8132e-06	2.9278e-06	4.0000	4.0166	2.0000	2.0060

In the figures 5.1 and 5.2 we decided to plot the numerical and the analytical results, precisely each sub-figure is obtained with a different mesh size. Lines are used to represent the analytical solutions while the +s are the points obtained with the numerical computations. It is possible to observe as the graphs of the saturation and the pressure are similar. We were hoping in such results considering that the

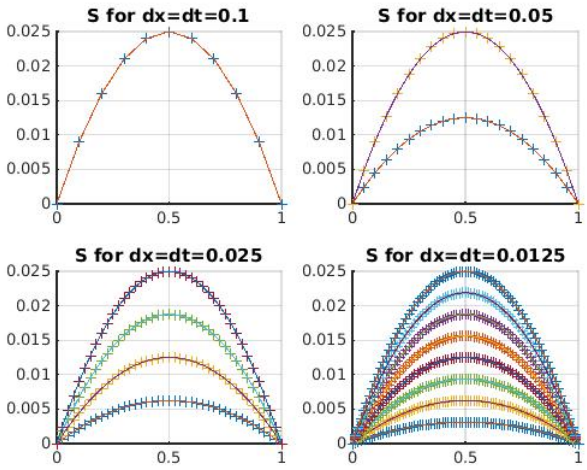


Figure 5.1: figure
Analytical and Numerical Pressure obtained with different meshes (IMPES Method)

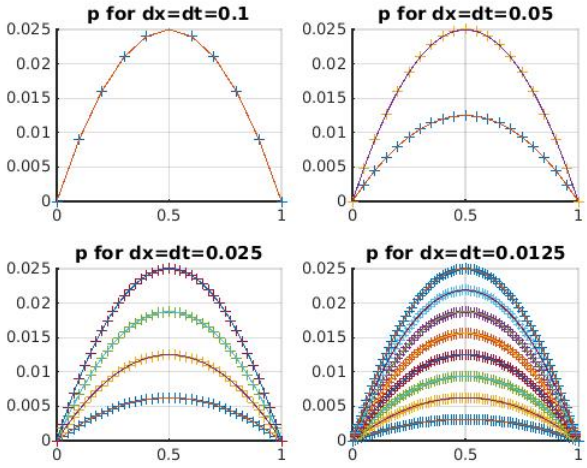


Figure 5.2: figure
Analytical and Numerical Saturation obtained with different meshes (IMPES Method)

analytical solutions are equal. Table 5.1 shows us the different errors obtained for different mesh sizes.

Given the errors obtained with different mesh size we can also study the *rates* and *orders* of convergence for both pressure and saturation. Such quantities are again presented into 5.1. We can observe as, for both the equations, the rates of convergence converge (r) to 4, while the orders (o) to 2.

Fully implicit method

The method presented above, the IMPES, is one of the most common scheme for solving two-phase problems, it can also be easily improved. Solving an equation explicitly, in this case the saturation, brings great instability to the system, we can then try to solve such equation using an implicit algorithm. Exactly as for the pressure equation an analogous scheme can be presented:

$$\Delta x\Phi\frac{S_i^{n+1}-S_i^n}{\Delta t}-(\frac{k_{i+1}+k_i}{2}\frac{\lambda_w(S_{i+1}^{n+1})+\lambda_w(S_i^{n+1})}{2}(\frac{p_{i+1}^{n+1}-p_i^{n+1}}{\Delta x}+$$
$$-\frac{1}{2}\frac{(p^{cap}(S_{i+1}^{n+1})-p^{cap}(S_i^{n+1})}{\Delta x}))+(\frac{k_i+k_{i-1}}{2}\frac{\lambda_w(S_i^{n+1})+\lambda_w(S_{i-1}^{n+1})}{2}$$
$$(\frac{p_i^{n+1}-p_{i-1}^{n+1}}{\Delta x}-\frac{1}{2}\frac{(p^{cap}(S_i^{n+1})-p^{cap}(S_{i-1}^{n+1})}{\Delta x}))=\frac{\Psi_w}{\rho_w}\Delta x$$

In this case solve the equation above is much more complex due to the fact that the unknown quantity S^{n+1} is contained in the capillary pressure and the permeability. The common solution of this problem is the introduction of an inside loop $S^{n+1,j+1}$ such that the permeability will be computed with the saturation obtained in the previous step. Such loop starts with the assumption $S^{n+1,1}=S^n$ and will stop whenever $\|S^{n+1,j+1}-S^{n+1,j}\|<\epsilon$, with ϵ a known constant, which gives the accuracy of the method.

$$\Delta x\Phi\frac{S_i^{n+1,j+1}-S_i^n}{\Delta t}-(\frac{k_{i+1}+k_i}{2}\frac{\lambda_w(S_{i+1,j}^{n+1})+\lambda_w(S_i^{n+1,j})}{2}$$
$$(\frac{p_{i+1}^{n+1}-p_i^{n+1}}{\Delta x}+\frac{1}{2}\frac{p^{cap}(S_{i+1,j+1}^{n+1})-p^{cap}(S_i^{n+1,j+1}))}{\Delta x}))+$$
$$(\frac{k_i+k_{i-1}}{2}\frac{\lambda_w(S_i^{n+1,j})+\lambda_w(S_{i-1}^{n+1,j})}{2}(\frac{p_i^{n+1}-p_{i-1}^{n+1}}{\Delta x}+$$
$$-\frac{1}{2}\frac{p^{cap}(S_i^{n+1,j+1})-p^{cap}(S_{i-1}^{n+1,j+1}))}{\Delta x}))=\frac{\Psi_w}{\rho_w}\Delta x$$

Using the following approximation

$$\frac{\partial p^{cap}(S_i^{n+1,j+1})}{\partial x}\sim p'_{cap}(S_i^{n+1,j})\cdot\frac{\partial}{\partial x}S_i^{n+1,j+1}$$

we can proceed in rewriting the scheme as:

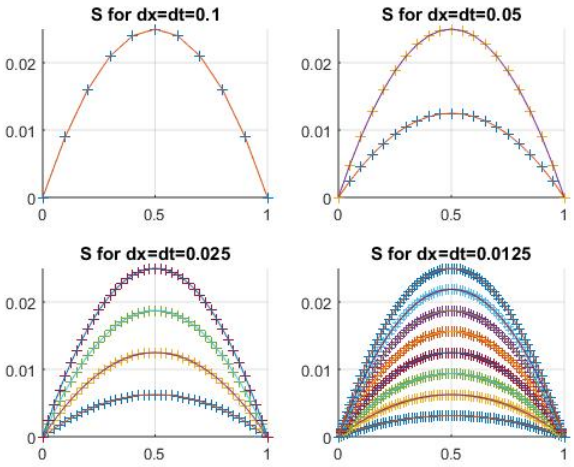


Figure 5.3: figure
Analytical and Numerical Saturation for different meshes (Implicit Scheme)

$$
\begin{aligned}
&S_{i-1}^{n+1,j+1}\left(\frac{\lambda_w(S_i^{n+1,j})+\lambda_w(S_{i-1}^{n+1,j})}{2}\frac{k_i+k_{i-1}}{2}\frac{\Delta t}{2}\left(\frac{p'_{cap}(S_i^{n+1,j})-p'_{cap}(S_{i-1}^{n+1,j})}{2}\right)\right)+\\
&S_i^{n+1,j+1}\left(\Delta x^2\Phi-\frac{\lambda_w(S_{i+1}^{n+1,j})+\lambda_w(S_i^{n+1,j})}{2}\frac{k_{i+1}+k_i}{2}\frac{\Delta t}{2}\left(\frac{p'_{cap}(S_{i+1}^{n+1,j})-p'_{cap}(S_i^{n+1,j})}{2}\right)+\right.\\
&\left.-\frac{\lambda_w(S_i^{n+1,j})+\lambda_w(S_{i-1}^{n+1,j})}{2}\frac{k_i+k_{i-1}}{2}\frac{\Delta t}{2}\left(\frac{p'_{cap}(S_i^{n+1,j})-p'_{cap}(S_{i-1}^{n+1,j})}{2}\right)\right)+\\
&S_{i+1}^{n+1,j+1}\left(\frac{\lambda_w(S_{i+1}^{n+1,j})+\lambda_w(S_i^{n+1,j})}{2}\frac{k_{i+1}+k_i}{2}\frac{\Delta t}{2}\left(\frac{p'_{cap}(S_{i+1}^{n+1,j})-p'_{cap}(S_i^{n+1,j})}{2}\right)\right)=\\
&=\Delta x^2\Delta t\frac{F_{w,i}}{\rho_w}+\Delta t\left(\frac{\lambda_w(S_{i+1}^{n+1,j})+\lambda_w(S_i^{n+1,j})}{2}\right)\frac{k_{i+1}+k_i}{2}(p_{i+1}^{n+1}-p_i^{n+1})+\\
&-\Delta t\left(\frac{\lambda_w(S_i^{n+1,j})+\lambda_w(S_{i-1}^{n+1,j})}{2}\right)\frac{k_i+k_{i-1}}{2}(p_i^{n+1}-p_{i-1}^{n+1})
\end{aligned}
$$

We can now solve the equation above using the same system approach introduced with the pressure equation.

Again we will presents our results through the graph of the solution, plots 5.3 and the errors obtained.

Table 5.2: Computational errors and convergence order for the Saturation (Implicit Scheme)

$\Delta x = \Delta t$	Saturation er	r_S	o_S
0.1	0.00019362	-	-
0.05	4.7459e-05	3.9992	2.0000
0.025	1.176e-05	3.9991	2.0000
0.0125	2.9278e-06	3.9995	2.0000

5.1.1 Transport equation

We need now to verify that the code developed above for the resolution of the transport equation actually works. We will again build the equation around a known function, our analytical solution, and we will then compute the error.

We will choose as analytical solution the same function as before: $c(x,t) = tx(1-x)$. The external force can be computed thanks to

$$f(x,t) = \Phi\frac{\partial(Sc)}{\partial t} + \frac{\partial}{\partial x}(-\Phi DS\frac{\partial c}{\partial x}) + \frac{\partial(u_w c)}{\partial x} - R(c)$$

where $D = 1$ and $R(c)$ represents the reaction term.
With the new external force f we can redefine the transport equation 2.26 and proceed in present the numerical results obtained for different reaction terms.

R(c) = a c(x,t)

For this case the reaction term is given by a linear equation, therefore the method built before can be used without further modifications.
Recall:

$$\begin{aligned}
&c_{i-1}^{n+1}\underbrace{\left(-\Phi D\frac{S_i^{n+1}+S_{i-1}^{n+1}}{2\Delta x^2}\right)}_{\alpha_i} + \\
&+c_i^{n+1}\underbrace{\left(-\Phi\ \frac{S_i^{n+1}}{\Delta t} + \frac{2S_i^{n+1}+S_{i-1}^{n+1}+S_{i+1}^{n+1}}{2\Delta x^2}\frac{w_{w,i}}{\Delta x} - a\right)}_{\beta_i} \\
&+c_{i+1}^{n+1}\underbrace{\left(-\Phi D\frac{S_{i+1}^{n+1}+S_i^{n+1}}{2\Delta x^2}\frac{u_{w,i+1}}{\Delta x}\right)}_{\gamma_i} \\
&=\underbrace{\frac{S_i^n c_i^n}{\Delta t}\Phi + f_i^n}_{g_i}
\end{aligned}$$

obtaining then the linear system $\mathbf{B}\mathbf{c}^{n+1} = \mathbf{g}$.
The solutions of such system are presented through the figure 5.4, as expected the graph is similar to the other already shown, this is due to the fact that the analytical solution is the same. We will also presents a table, 5.3, with the numerical error and the rate and order of convergence of the method.

It is possible to observe that also for this algorithm we have a rate converging to 4 and a order converging to 2.

R(c) = c(x,t)/a+c(x,t)

In this case the reaction term $R(c) = \frac{c}{a+c}$ is called *monod term*. It does represent an important example because it gives us a slightly more difficult differential equation

$$\Phi\frac{\partial(S_w c)}{\partial t} + \frac{\partial}{\partial x}(-\Phi DS_w\frac{\partial c}{\partial x} + u_w c) \ = \ \frac{c}{a+c} + f(x,t). \tag{5.1}$$

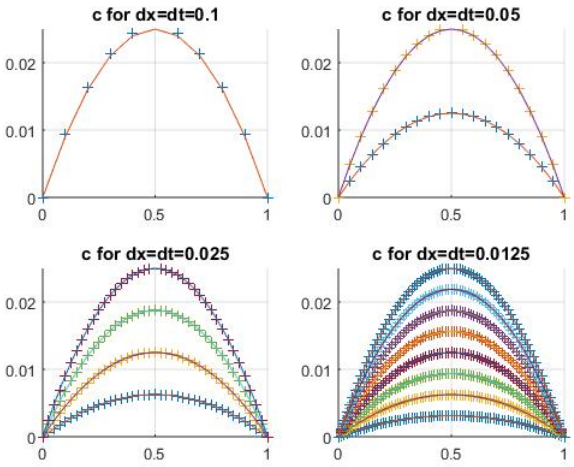


Figure 5.4: R(c)=ac
Analytical and Numerical Concentration

Table 5.3: Computational errors and Orders of Convergence for Transport equation (R(c)=ac)

$\Delta x = \Delta t$	Concentration er	r_c	o_c
0.1	0.00018788	-	-
0.05	4.2689e-05	4.4012	2.1379
0.025	1.0106e-05	4.4242	2.1379
0.0125	2.4561e-06	4.1147	2.1379

Explicit Method

The easiest way to solve this equation is to use an explicit method, precisely:

$$\Phi\frac{S_i^{n+1}c_i^{n+1}-S_i^n c_i^n}{\Delta t}-\frac{1}{\Delta x}\{\Phi D\frac{S_{i+1}^n+S_i^n}{2}\frac{c_{i+1}^n-c_i^n}{\Delta x}-\frac{u_{w,i+1}^n+u_{w,i}^n}{2}\frac{c_{i+1}^n+c_i^n}{2}+$$
$$-\Phi D\frac{S_i^n+S_{i-1}^n}{2}\frac{c_i^n-c_{i-1}^n}{\Delta x}+\frac{u_{w,i}^n+u_{w,i-1}^n}{2}\frac{c_i^n+c_{i-1}^n}{2}\}=\frac{c_i^n}{a+c_i^n}+f_i^{n+1}$$

obtaining the following expression for c_i^{n+1}

$$c_i^{n+1}=\frac{S_i^n}{S_i^{n+1}}c_i^n+\frac{\Delta t}{S_i^{n+1}\Delta x\Phi}\frac{1}{\Delta x}\{\Phi D\frac{S_{i+1}^n+S_i^n}{2}\frac{c_{i+1}^n-c_i^n}{\Delta x}-\frac{u_{w,i+1}^n+u_{w,i}^n}{2}\frac{c_{i+1}^n+c_i^n}{2}+$$
$$-\Phi D\frac{S_i^n+S_{i-1}^n}{2}\frac{c_i^n-c_{i-1}^n}{\Delta x}+\frac{u_{w,i}^n+u_{w,i-1}^n}{2}\frac{c_i^n+c_{i-1}^n}{2}\}+\frac{\Delta t}{S_i^{n+1}\Phi}\{\frac{c_i^n}{a+c_i^n}+f_i^{n+1}\}$$

The explicit method defined above is, as any other explicit formulation, characterized by the instability. To obtain a realistic solution the time step Δt must be much smaller than the mesh size Δx. We will show some examples and then we will analyse the results.

Table 5.4: Computational errors and Orders of Convergence for Transport equation, $\Delta t = \Delta x$

$\Delta x = \Delta t$	Concentration er	r_c	o_c
0.1	0.020292	-	-
0.05	0.024011	0.8451	-0.2428
0.025	76.274	0.0003	-11.6333
0.0125	5.4547e+11	0.0000	-32.7356

Table 5.5: Computational errors and Orders of Convergence for Transport equation, $\Delta t = 0.1\Delta x$

Δx	Δt	Concentration er	r_c	o_c
0.1	0.01	0.030641	-	-
0.05	0.005	2.0927e+09	1.0e-10*0.1464	-35.9911
0.025	0.0025	1.5462e+38	1.0e-10*0.0000	-95.8993
0.0125	0.00125	1.5071e+109	1.0e-10*0.0000	-235.8200

Table 5.6: Computational errors and Orders of Convergence for Transport equation, $\Delta t = 0.01\Delta x$

Δx	Δt	Concentration er	r_c	o_c
0.1	0.001	0.00010446	-	-
0.05	0.0005	3.3814e-05	3.0893	1.6273
0.025	0.00025	2.1806e-05	1.5506	0.6329
0.0125	0.000125	Inf	0	-Inf

Table 5.7: Computational errors and Orders of Convergence for Transport equation, $\Delta t = e - 03\Delta x$

Δx	Δt	Concentration er	r_c	o_c
0.1	0.0001	0.00017076	-	-
0.05	5e-05	4.0239e-05	4.2437	2.0853
0.025	2.5e-05	8.9145e-06	4.5139	2.1744
0.0125	1.25e-05	1.7658e-06	5.0483	2.3358

Table 5.8: Computational errors and Orders of Convergence for Transport equation, $\Delta t = e - 04\Delta x$

Δx	Δt	Concentration er	r_c	o_c
0.1	1e-05	0.00017962	-	-
0.05	5e-06	4.4642e-05	4.0235	2.0084
0.025	2.5e-06	1.1034e-06	4.0458	2.0164
0.0125	1.25e-06	2.6962e-06	4.0925	2.0330

Table 5.9: Computational errors and Orders of Convergence for Transport equation, $\Delta t = e - 05\Delta x$

Δx	Δt	Concentration er	r_c	o_c
0.1	1e-06	0.00018051	-	-
0.05	5e-07	4.5095e-05	4.0030	2.0011
0.025	2.5e-07	1.1261e-06	4.0047	2.0010
0.0125	1.25e-07	2.8088e-06	4.0090	2.0033

It is possible to observe that the results shown in the first tables are completely unrealistic, with errors increasing or, anyway too big. Only in the last two tables 5.8 and 5.9 the error is reasonable and again we obtain rates converging to 4 and orders to 2. Such results could be optimize even more with smaller time steps.

Implicit Method

An alternative way to solve such equation is again an implicit method. Exactly as in the case of the pressure equation a similar algorithm will be developed, an inside loop given by $j = 1, 2, ...$ is required to treat the non linearity of the equation. Precisely

$$\Phi\frac{S_i^{n+1}c_i^{n+1,j+1} - S_i^n c_i^n}{\Delta t} = \frac{1}{\Delta x}\Phi D\frac{S_{i+1}^n + S_i^n}{2}\frac{c_{i+1}^{n+1,j+1} - c_i^{n+1,j+1}}{\Delta x} +$$
$$- \frac{u_{w,i+1}^n + u_{w,i}^n}{2}\frac{c_{i+1}^{n+1,j+1} + c_i^{n+1,j+1}}{2} - \Phi D\frac{S_i^n + S_{i-1}^n}{2}\frac{c_i^{n+1,j+1} - c_{i-1}^{n+1,j+1}}{\Delta x}$$
$$+ \frac{u_{w,i}^n + u_{w,i-1}^n}{2}\frac{c_i^{n+1,j+1} + c_{i-1}^{n+1,j+1}}{2} + \frac{c_i^{n+1,j}}{a + c_i^{n+1,j}} + f_i^{n+1}$$

Here the results, plots 5.5 and errors 5.10 are presented

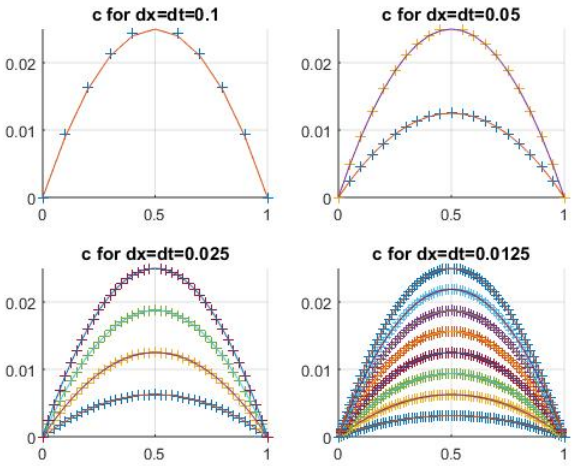


Figure 5.5: Implicit Scheme
Analytical and Numerical Saturation of transport

Table 5.10: Computational errors and Orders of Convergence for Transport equation

$\Delta x = \Delta t$	Concentration er	r_c	o_c
0.1	0.00017901	-	-
0.05	4.493e-05	3.9420	1.9943
0.025	1.1258e-05	3.9910	1.9967
0.0125	2.8179e-06	3.9952	1.9983

It is evident as the implicit method requires more effort in writing the code but after that the new algorithm is much more efficient. We obtained better results, smaller errors and faster convergences, already for $\Delta t = \Delta x$, compared with the explicit method.

5.2 Richards and transport equations

This section will be used to study and validate the linearization methods developed before, we will use the same analytical functions as for the two-phase flow problems and we will once again compute the errors. Using the analytical expressions of the pressure head and the concentration, into 4.2 and 4.3 we can obtain the external forces of the two equations. To compute such quantities we need an expression for the *conductivity* K and the *water content* θ, such quantities have been defined first thanks to manufactured functions not directly related to physical phenomena, functions with easy expressions in which it was possible to underline the double dependence form Ψ and c. Later we will refer to previous works as [15, 12, 22] and we are going to study a real physical case: the salinity problem.

We won't plot any graph showing the numerical and the analytical solutions because they are extremely similar to the one presented before, we used the same manufactured solution. We will instead study the errors and more in the details computational times and conditional numbers.

To start our studies we need to define two expressions for θ and K, the first example will regard two easy expressions for such quantities, precisely:

$$\theta = \alpha(\Psi^2 + exp(c) + \frac{1}{2})$$
$$K = \theta^3$$

with α a fitting parameter. Substituting these two expressions and the analytical solutions

$$\Psi_{anal} = x * (1 - x) * t$$
$$c_{anal} = x * (1 - x) * t$$

into 4.2 and 4.3 we can finally obtain an expression for the external forces. We can also present a graph 5.6 of the water content observing that such quantity can also be written as: $\theta = \Phi S_w$

From the graph 5.6, observing the values of S_w, we can understand the movement of the water inside the soil. At $t = 0$ some water is already inside the media, $S_w \neq 0$ and such quantity will increase until the final time $t = 1$, it is also interesting to observe that not all of the pores will be filled by the liquid, S_w is different from 1. For this reason, the example represents an unsaturated configuration of the soil.

5.2.1 L-Scheme

We will now study the *L-scheme* introduced before rewriting the equations 4.2 and 4.3 and showing the error obtained. Recalling the two expressions 4.4 and 4.5

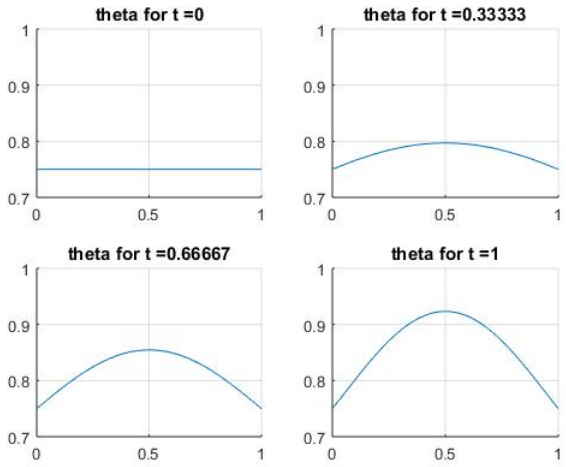


Figure 5.6: figure
Profile of S_w for different time

and computing the space derivatives using the two point flux approximation, we obtain

$$\Delta x^2(\theta(\Psi_i^{n+1,j}, c_i^n) - \theta(\Psi_i^n, c_i^n) + L_1(\Psi_i^{n+1,j+1} - \Psi_i^{n+1,j}))+$$
$$- \Delta t(\frac{K(\Psi_{i+1}^{n+1,j}, c_{i+1}^n) + K(\Psi_i^{n+1,j}, c_i^n)}{2}(\Psi_{i+1}^{n+1,j} - \Psi_i^{n+1,j})+$$
$$- \frac{K(\Psi_i^{n+1,j}, c_i^n) + K(\Psi_{i-1}^{n+1,j}, c_{i-1}^n)}{2}(\Psi_i^{n+1,j} - \Psi_{i-1}^{n+1,j})) = \Delta t \Delta x^2 H(x,t)$$

and

$$\Delta x^2[\theta(\Psi_i^{n+1}, c_i^{n+1,j}) - \theta(\Psi_i^n, c_i^n) + L_2(c_i^{n+1,j+1} - c_i^{n+1,j})] * c_i^{n+1,j}+$$
$$+ \Delta x^2 \Theta(\Psi_i^{n+1}, c_i^{n+1,j}) * (c_i^{n+1,j+1} - c_i^n) + \Delta t(-\frac{\theta(\Psi_{i+1}^{n+1}, c_{i+1}^{n+1,j} + \theta(\Psi_i^{n+1}, c_i^{n+1,j}}{2})$$
$$(c_{i+1}^{n+1,j+1} - c_i^{n+1,j+1}) + u_{w,i+\frac{1}{2}} * \frac{c_{i+1}^{n+1,j+1} + c_i^{n+1,j+1}}{2} + \frac{\theta(\Psi_i^{n+1}, c_i^{n+1,j} + \theta(\Psi_{i-1}^{n+1}, c_{i-1}^{n+1,j}}{2})$$
$$(c_i^{n+1,j+1} - c_{i-1}^{n+1,j+1}) - u_w, i - \frac{1}{2} * \frac{c_i^{n+1,j+1} + c_{i-1}^{n+1,j+1}}{2}) = \Delta t \Delta x^2 H_c(x,t)$$

each of the equations above can be rewritten in the linear system form, exactly as for the other implicit method implemented before, for example the *Richard equation* becomes $\mathbf{A}\mathbf{\Psi}^{n+1,j+1} = \mathbf{f}$ with $\mathbf{A}$ given by

$$\mathbf{A} = \begin{bmatrix} BCs & & & & & & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & 0 & & & \\ 0 & \alpha_3 & \beta_3 & \gamma_3 & 0 & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & \ddots & \alpha_i & \beta_i & \gamma_i & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & & \alpha_N & \beta_N & \gamma_N \\ 0 & & & & & & BCs \end{bmatrix}$$

α, β and γ obtainable from the first of the two equations above. The values for L_1 and L_2 are respectively 0.6 and 1.3.

The epsilon used for the while loop regarding j has been define as $\epsilon = 10^{-7}$. We can now proceed presenting into the table 5.11 the errors and the convergence rates obtained.

Table 5.11: Computational errors and Orders of Convergence for L-scheme

$\Delta x = \Delta t$	Ψ er	c er	r_p	r_c	o_p	o_c
0.1	0.0016099	0.0012708	-	-	-	-
0.05	0.00039	0.00030953	4.1280	4.1057	2.0454	2.0376
0.025	8.9591e-05	7.3449e-05	4.3535	4.2142	2.1222	2.0753
0.0125	1.9905e-06	1.6927e-06	4.5013	4.3391	2.1703	2.1174
0.00352	5.8186e-06	4.2113e-06	3.4203	4.0195	1.7741	2.0070

We can observe as, for both the equations, the rates of convergence converge to 4, while the orders to 2.

The *L-scheme* is characterized by the introduction of these constants L_1 and L_2, it can be interesting to presents the results obtained for different values of such constants. The main differences are in the computational times, here into the table 5.13, we present the CPU time for each couple of constants

Table 5.12: Computational times for L-scheme

L_1	L_2	$CPUtime$
0.6	1.3	47.75
0.8	1.5	58.40
1	1.7	61.46
1.5	2	79.43
2	3	122.20

Table 5.13: table

$$max\|\frac{\partial\theta}{\partial\Psi}\| = 0.5001 \text{ and } max\|\frac{\partial\theta}{\partial c}\| = 1.2803$$

It is possible to observe as the code results faster for values of the constants closer to the respective derivatives of θ, clearly some differences are obtained also in the computation of the errors and the order of convergence but they are less notable.
Such result have been proved analytically, for the decoupled problem, in the previous chapter. We showed that there is a precise relation between the errors obtained at the time step *n+1* and *n*, as shown by 4.20. We can observe that for small L we have a faster convergence of the scheme, at the same time the constant L must verify the condition $L \geq max\|\frac{\partial\theta}{\partial\Psi}\|$.

5.2.2 Newton monolithic method

In this section the *Newton (Monolithic) method* will be analysed, we will observe its velocity of computation but also some disadvantages. The main problem consists

in the derivation of the *Jacobian matrix*, each monolithic method is characterized in fact by a massive matrix, such matrix can have high condition number and this can bring to the instability of the system. Even more problematic is the computation of the entrance of this matrix, as explained in one of the previous sections, each entrance represents a derivative. As we all know it is not always possible to evaluate derivatives and in such cases the scheme can not be used.

Again the errors and convergence orders are presented into 5.14

Table 5.14: Computational errors and Orders of Convergence for Newton scheme

$\Delta x = \Delta t$	Ψ er	c er	r_p	r_c	o_p	o_c
0.1	0.00165	0.001151	-	-	-	-
0.05	0.0004119	0.00029567	4.0127	3.8930	2.0046	1.9609
0.025	9.8688e-05	0.0014603	4.1666	2.0247	2.0589	1.0177
0.0125	2.3456e-05	8.2619e-05	4.2074	1.7675	2.0729	0.8217
0.00625	6.5191e-06	4.4455e-05	3.5980	1.8585	1.8472	0.8941

Very interesting is the *computation time*, for this code, with this particular parameters, the elapsed time is only *33.06s*, much smaller than the fastest result using the L-scheme.

The results presented in the Tab 5.14 can be easily improved reducing the size of the time mesh, precisely defining $\Delta t = 1/5\Delta x$. With the new formulation, obtained updating the time step we have new results, presented into Tab 5.15.

Table 5.15: Computational errors and Orders of Convergence for Newton scheme

Δx	Ψ er	c er	r_p	r_c	o_p	o_c
0.1	0.0016368	0.0015431	-	-	-	-
0.05	0.00041281	0.0003716	3.9650	4.1524	1.9873	2.0539
0.025	0.00010183	8.5862e-05	4.0539	4.3279	2.0193	2.1137
0.0125	2.4592e-05	1.8567e-05	4.1408	4.6246	2.0499	2.2093
0.00625	5.7283e-06	4.2057e-06	4.2931	4.4146	2.1020	2.1423

5.2.3 Newton method

We will now present the results obtained using the non monolithic approach of the Newton method. As always we will present the computational errors and the orders of convergence, Tab 5.16.

In this case two different Jacobian matrix will be computed, one for each equation, investigating separately the dependence of θ from both Ψ and c.

Again to obtain these particular result we have defined, as in the previous case, $\Delta t = 1/5\Delta x$.

Table 5.16: Computational errors and Orders of Convergence for Newton scheme

Δx	Ψ er	c er	r_p	r_c	o_p	o_c
0.1	0.0016368	0.0015431	-	-	-	-
0.05	0.00041281	0.0003716	3.9650	4.1524	1.9873	2.0539
0.025	0.00010183	8.5862e-05	4.0539	4.3279	2.0193	2.1137
0.0125	2.4592e-05	1.8567e-05	4.1408	4.6246	2.0499	2.2093
0.00625	5.7283e-06	4.2057e-06	4.2931	4.4146	2.1020	2.1423

5.2.4 Modified Picard

The *Modified Picard* scheme, introduced by Celia [11] and presented in the last chapter, will be here analysed. We will, as before, present computational errors and orders of convergence. Before to do that we must rewrite the system of equations, computing also the space derivatives and obtaining

$$\Delta x^2(\theta(\Psi_i^{n+1,j}, c_i^n) - \theta(\Psi_i^n, c_i^n) + \theta_\Psi(\Psi_i^{n+1,j}, c_{i+1}^{n+1,j})(\Psi_i^{n+1,j+1} - \Psi_i^{n+1,j}))+$$
$$- \Delta t(\frac{K(\Psi_{i+1}^{n+1,j}, c_{i+1}^n) + K(\Psi_i^{n+1,j}, c_i^n)}{2}(\Psi_{i+1}^{n+1,j} - \Psi_i^{n+1,j})+$$
$$- \frac{K(\Psi_i^{n+1,j}, c_i^n) + K(\Psi_{i-1}^{n+1,j}, c_{i-1}^n)}{2}(\Psi_i^{n+1,j} - \Psi_{i-1}^{n+1,j})) = \Delta t \Delta x^2 H(x,t)$$

and

$$\Delta x^2[\theta(\Psi_i^{n+1}, c_i^{n+1,j}) - \theta(\Psi_i^n, c_i^n) + \theta_c(\Psi_i^{n+1,j}, c_{i+1}^{n+1,j})(c_i^{n+1,j+1} - c_i^{n+1,j})] * c_i^{n+1,j}+$$
$$+ \Delta x^2 \Theta(\Psi_i^{n+1}, c_i^{n+1,j}) * (c_i^{n+1,j+1} - c_i^n) + \Delta t(-\frac{\theta(\Psi_{i+1}^{n+1}, c_{i+1}^{n+1,j} + \theta(\Psi_i^{n+1}, c_i^{n+1,j}}{2})$$
$$(c_{i+1}^{n+1,j+1} - c_i^{n+1,j+1}) + u_{w,i+\frac{1}{2}} * \frac{c_{i+1}^{n+1,j+1} + c_i^{n+1,j+1}}{2} + \frac{\theta(\Psi_i^{n+1}, c_i^{n+1,j} + \theta(\Psi_{i-1}^{n+1}, c_{i-1}^{n+1,j}}{2})$$
$$(c_i^{n+1,j+1} - c_{i-1}^{n+1,j+1}) - u_w, i - \frac{1}{2} * \frac{c_i^{n+1,j+1} + c_{i-1}^{n+1,j+1}}{2}) = \Delta t \Delta x^2 H_c(x,t)$$

Same expression as before for θ and K so that we will be able to compare the results obtained.

Now the errors and convergence orders are presented into 5.17

Table 5.17: Computational errors and Orders of Convergence Newton scheme

$\Delta x = \Delta t$	Ψ er	c er	r_p	r_c	o_p	o_c
0.1	0.0016895	0.0013088	-	-	-	-
0.05	0.00042919	0.00032796	3.9365	3.9908	1.9769	1.9967
0.025	0.00010773	8.2066e-05	3.9839	3.9964	1.9942	1.9987
0.0125	2.696e-05	2.0524e-05	3.9960	3.9985	1.9986	1.9994
0.00625	6.7409e-06	5.132e-06	3.9994	3.9993	1.9998	1.9997

We can observe for both of the equations an order of convergence close to 2 and a rate close to 4. In the next chapter we will compare the computational times of these schemes.

5.2.5 Monolithic Picard scheme

In this section we will present the results of the *Monolithic-Picard scheme*. Such method aim to solve the 2 equations simultaneously, considering them as a system of the form

$$\theta_\Psi \frac{\Psi^{n+1,j+1}-\Psi^n}{\Delta t} + \theta_c \frac{c^{n+1,j+1}-c^n}{\Delta t} - \frac{\partial}{\partial x}(K(\Psi^{n+1,j}, c^{n+1,j})\frac{\partial}{\partial x}\Psi = H(x,t)$$
$$(\theta_\Psi \frac{\Psi^{n+1,j+1}-\Psi^n}{\Delta t} + \theta_c \frac{c^{n+1,j+1}-c^n}{\Delta t})c^{n+1,j} + \theta \frac{c^{n+1,j+1}-c^n}{\Delta t}$$
$$+ \frac{\partial}{\partial x}(-\theta(\Psi^{n+1,j+1}, c^{n+1,j})\frac{\partial c}{\partial x} + u_w * c) = H_c(x,t)$$

We will again solve the space derivatives using the two point flux approximations, obtaining the final expression:

$$\theta_\Psi \frac{\Psi^{n+1,j+1}-\Psi^n}{\Delta t} + \theta_c \frac{c^{n+1,j+1}-c^n}{\Delta t} +$$
$$- \frac{1}{\Delta x}(\frac{K(\Psi_{i+1}^{n+1,j}, c_i^n) + K(\Psi_i^{n+1,j}, c_{i-1}^n)}{2}\frac{\Psi_{i+1}^{n+1,j}-\Psi_i^{n+1,j}}{\Delta x} +$$
$$- \frac{K(\Psi_i^{n+1,j}, c_i^n) + K(\Psi_{i-1}^{n+1,j}, c_{i-1}^n)}{2}\frac{\Psi_i^{n+1,j}-\Psi_{i-1}^{n+1,j}}{\Delta x}) = H(x,t)$$
$$(\theta_\Psi \frac{\Psi^{n+1,j+1}-\Psi^n}{\Delta t} + \theta_c \frac{c^{n+1,j+1}-c^n}{\Delta t})c^{n+1,j} + \theta \frac{c^{n+1,j+1}-c^n}{\Delta t} +$$
$$+ \frac{1}{\Delta x}(-\frac{\theta(\Psi_{i+1}^{n+1}, c_{i+1}^{n+1,j}) + \theta(\Psi_i^{n+1}, c_i^{n+1,j})}{2}\frac{c_{i+1}^{n+1,j+1}-c_i^{n+1,j+1}}{\Delta x})$$
$$+ u_{w,i+\frac{1}{2}} * \frac{c_{i+1}^{n+1,j+1}+c_i^{n+1,j+1}}{2} + \frac{\theta(\Psi_i^{n+1}, c_i^{n+1,j}) + \theta(\Psi_{i-1}^{n+1}, c_{i-1}^{n+1,j})}{2}$$
$$\frac{c_i^{n+1,j+1}-c_{i-1}^{n+1,j+1}}{\Delta x}) - u_{w,i-\frac{1}{2}} * \frac{c_i^{n+1,j+1}+c_{i-1}^{n+1,j+1}}{2}) = H_c(x,t)$$

We can now present the usual table (5.18) with errors and orders of convergence

Table 5.18: Computational errors and Orders of Convergence for Newton scheme

$\Delta x = \Delta t$	Ψ er	c er	r_p	r_c	o_p	o_c
0.1	0.0016895	0.0013088	-	-	-	-
0.05	0.00042919	0.00032796	3.9365	3.9908	1.9769	1.9967
0.025	0.00010773	8.2066e-05	3.9839	3.9964	1.9942	1.9987
0.0125	2.696e-05	2.0524e-05	3.9960	3.9985	1.9986	1.9994
0.00625	6.7409e-06	5.132e-06	3.9991	3.9993	1.9997	1.9998

It is possible to observe that the results obtained are extremely close to the one we got applying the *Modified Picard*. This interesting result is due to the fact that the linearizations used in the monolithic scheme are the same as the one used into the Picard. In the next section we will also compare the computational times and, if possible, the condition numbers of the systems.

5.2.6 Monolithic L-scheme

Here the *Monolithic L scheme* will be analysed, solving the space and time derivatives of the two equations and studying them as a singular system, we obtain:

$$\theta(\Psi^{n+1,j}, c^n) - \theta(\Psi^n, c^n) + L_1(\Psi^{n+1,j+1} - \Psi^{n+1,j}) + L_2(c^{n+1,j+1} - c^{n+1,j})$$
$$- \frac{1}{\Delta x}\left(\frac{K(\Psi_{i+1}^{n+1,j}, c_i^n) + K(\Psi_i^{n+1,j}, c_{i-1}^n)}{2}\frac{\Psi_{i+1}^{n+1,j} - \Psi_i^{n+1,j}}{\Delta x} + \right.$$
$$\left. - \frac{K(\Psi_i^{n+1,j}, c_i^n) + K(\Psi_{i-1}^{n+1,j}, c_{i-1}^n)}{2}\frac{\Psi_i^{n+1,j} - \Psi_{i-1}^{n+1,j}}{\Delta x}\right) = H(x,t)$$
$$[\theta(\Psi^{n+1}, c^{n+1,j}) - \theta(\Psi^n, c^n) + L_1(\Psi^{n+1,j+1} - \Psi^{n+1,j}) + L_2(c^{n+1,j+1} - c^{n+1,j})] * c^{n+1,j}$$
$$+ \theta * (c^{n+1,j+1} - c^n) + \frac{1}{\Delta x}\left(-\frac{\theta(\Psi_{i+1}^{n+1}, c_{i+1}^{n+1,j}) + \theta(\Psi_i^{n+1}, c_i^{n+1,j})}{2}\frac{c_{i+1}^{n+1,j+1} - c_i^{n+1,j+1}}{\Delta x}\right)$$
$$+ u_{w,i+\frac{1}{2}} * \frac{c_{i+1}^{n+1,j+1} + c_i^{n+1,j+1}}{2} + \frac{\theta(\Psi_i^{n+1}, c_i^{n+1,j}) + \theta(\Psi_{i-1}^{n+1}, c_{i-1}^{n+1,j})}{2}$$
$$\left.\frac{c_i^{n+1,j+1} - c_{i-1}^{n+1,j+1}}{\Delta x}\right) - u_{w,i-\frac{1}{2}} * \frac{c_i^{n+1,j+1} + c_{i-1}^{n+1,j+1}}{2}) = H_c(x,t)$$

The results will be again presented with the following table 5.19.

Table 5.19: Computational errors and Orders of Convergence for Monolithic-L scheme

$\Delta x = \Delta t$	Ψ er	c er	r_p	r_c	o_p	o_c
0.1	0.001506	0.0012696	-	-	-	-
0.05	0.00036128	0.00030926	4.1685	4.1052	2.0595	2.0374
0.025	8.1772e-05	7.3399e-05	4.4181	4.2135	2.1434	2.0750
0.0125	1.7829e-05	1.6923e-05	4.5865	4.3373	2.1974	2.1168
0.00625	5.4571e-06	4.2134e-06	3.2671	4.0164	1.7080	2.0059

Chapter 6

Comparison of different iterative schemes for coupled flows and transport in porous media: an academic example

In this chapter we will present different approaches to the coupled problem given by the Richards and transport equations. Each approach will treat the coupling aspect of the system in a different way. Until this moment, for each linearization scheme, we presented a monolithic and a non monolithic method, here other alternatives will be investigated. Each approach will differ from the others on how the accuracy requirement given by ϵ is achieved.

We will observe as the non monolithic approach used in this thesis (later listed as *method 2*) has some interesting advantages compared to the common solving formulation (*method 4*).

6.1 Five Different Approaches to the Coupled Problem

We will here list five different approaches to the coupled problem given by the Richards and transport equations:

- *Method 1*: *Monolithic case*, the two equations are treated as a system of the form
$$\begin{cases} F_1(\Psi^{n+1,j+1}, c^{n+1,j+1}) = H(x,t) \\ F_2(\Psi^{n+1,j+1}, c^{n+1,j+1}) = H_c(x,t) \end{cases}$$
a singular while loop is built requiring that both Ψ and c converge to the solution.

- *Case 2*: *Iterative fully implicit*, the equations are solved separately, *one iteration* of the linearization scheme is applied on each equation, the Ψ obtained from the Richards equation is used into the transport and the c obtained here will be used into the next Richards, such chain will stop when both Ψ and c

satisfy the convergence criteria given by ϵ. We have again just a while loop, using the expressions above, they can be reformulated as follow:

$$F_1(\Psi^{n+1,j+1}, c^{n+1,j}) = H(x,t) \;\leftrightarrows\; F_2(\Psi^{n+1,j+1}, c^{n+1,j+1}) = H_c(x,t)$$

- *Method 3*: *Semi-implicit*, the two equations are solved separately, each of them will have its own while loop, which will stop when the solution satisfies the convergence requirement. We solve, for example, the Richards equation using each time the Ψ obtained at the previous step without updating the concentration c, precisely:

$$\overset{loop_j}{\underset{\curvearrowright}{\bigcirc}} \qquad \overset{loop_i}{\underset{\curvearrowright}{\bigcirc}}$$

$$F_1(\Psi^{n+1,j+1}, c^n) \;=\; H(x,t) \qquad F_2(\Psi^{n+1}, c^{n+1,i+1}) \;=\; H_c(x,t)$$

- *Method 4*: this approach is more complex than the previous ones and it does require the introduction of two new loops, precisely we will have

$$\Psi^{n+1,j+1,k_1+1} \underset{k_1}{\longrightarrow} \Psi^{n+1,j+1} \qquad c^{n+1,j+1,k_2+1} \underset{k_2}{\longrightarrow} c^{n+1,j+1}$$

  plus the usual loop given by

$$\Psi^{n+1,j+1} \underset{j}{\rightarrow} \Psi^{n+1} \qquad c^{n+1,j+1} \underset{j}{\rightarrow} c^{n+1}$$

  this means that we are going to solve the Richards equation until we obtain the convergence for the loop given by k_1, we will use then the values $\Psi^{n+1,j+1}$ into the transport. We will again iterate until the convergence for k_2 is obtained, the computed concentration will be then implemented into the Richards and so on until the loop defined trough j converges.

- *Method 5*: *no while loop*, this case is for sure the fastest but also the less accurate, the linearization scheme is applied only once without requiring any convergence criteria to be satisfied.

We can try to show the difference between such formulations, presenting part of the Matlab codes used in the cases listed above, especially for *Methods 2,3,4* and *5*. We will start from the fourth method which is the most complex and we will show as all the other three approaches are only simplification of it.

Method 4

Observing the code listed below we can notice the *while loop* into *line 1* which requires that both $\Psi^{n+1,j+1}$ and $c^{n+1,j+1}$ satisfy the convergence requirement given by ϵ. In *line 5* we can find the next loop regarding the Richards equation, such loop requires the convergence of the quantity $\Psi^{n+1,j+1,k_1+1}$ and its outcome will be the value $\Psi^{n+1,j+1}$ then used into the transport equation. In *line 19* the third and last *while loop* is defined and it requires the convergence of $c^{n+1,j+1,k_2+1}$to $c^{n+1,j+1}$.

```
1  while norm_Psi>ep || norm_c>ep
2  % norm for Psi(n+1,j+1) and c(n+1,j+1),
3  % after convergence we will have Psi(n+1) and c(n+1)
4
5    while norm_1>ep   % k_1 while loop, after convergence we will have Psi(n+1,j+1)
6
7     for i= 2:N   % let's build the coeff a_i and b_i to solve
8                  % the linear system related to Richards' eq
9           ...
10
11    end
12   % Solve the system
13    Psi_k_new = (linsolve(A,b'))';
14    norm_1 = norm(Psi_k-Psi_k_new);
15    Psi_k = Psi_k_new;
16   end
17 Psi_new = Psi_k;     %Psi(n+1,j+1)
18
19   while norma_2>ep    % k_2 while loop,  after convergence we will have c(n+1,j+1)
20
21    for i= 2:N   % let's build the coeff a_i and b_i to solve
22               % the linear system related to transport eq
23       ...
24
25    end
26   % Solve the system
27    c_k_new = (linsolve(B,d'))';
28    norm_2 = norm(c_k-c_k_new);
29    c_k = c_k_new;
30   end
31 c_new = c_k;    %c(n+1,j+1)
32
33 norm_Psi = norm(Psi_new-Psi);
34 Psi = Psi_new;
35
36 norm_c = norm(c_new-c);
37 c = c_new;
38 % if both norms satisfy convergence conditions then
39 % we obtain Psi(n+1) and c(n+1)
40
41 end
```

Method 2

As said before the second method, the one used in our thesis, is only a simplification of the more commonly used *Method 4*. To obtain *Method 2* from *Method 4* is enough to comment the while loops in the lines *5* and *19*, in this way we use, as required, only one iteration on each equation, but, thanks to the while loop in line *1* we still couple the two equations.

Method 3

The third approach is characterized by the absence of an external loop coupling the two equations. We will solve the Richards and use the Ψ obtained into the transport equation but, the concentration then computed, will not be used into the next Richards because the code will stop when such concentration is obtained. Clearly this formulation is a simplification of *Method 4*, it is enough to comment the while loop into the first line to obtain the method described.

Method 5

The last and less accurate method is the simplest of all, in such approach there won't be any iteration, we will apply the linearization schemes only once on both Richards and transport equations and the values obtained will be our solutions. Referring to the code listed above we need to comment all the while loops.

We can now concentrate on the *L-scheme*, *modified Picard* and *Newton* lineariza-

tion schemes, we will plot the computational times and errors obtained using the five methods listed above to solve the coupled problem, Richards plus transport, defined in the previous chapter.

In this thesis we have used the monolithic scheme and the *Method 2* described above, we want to show as such approaches represent a valid choice for our problem. The scheme more commonly used is *Method 4*.

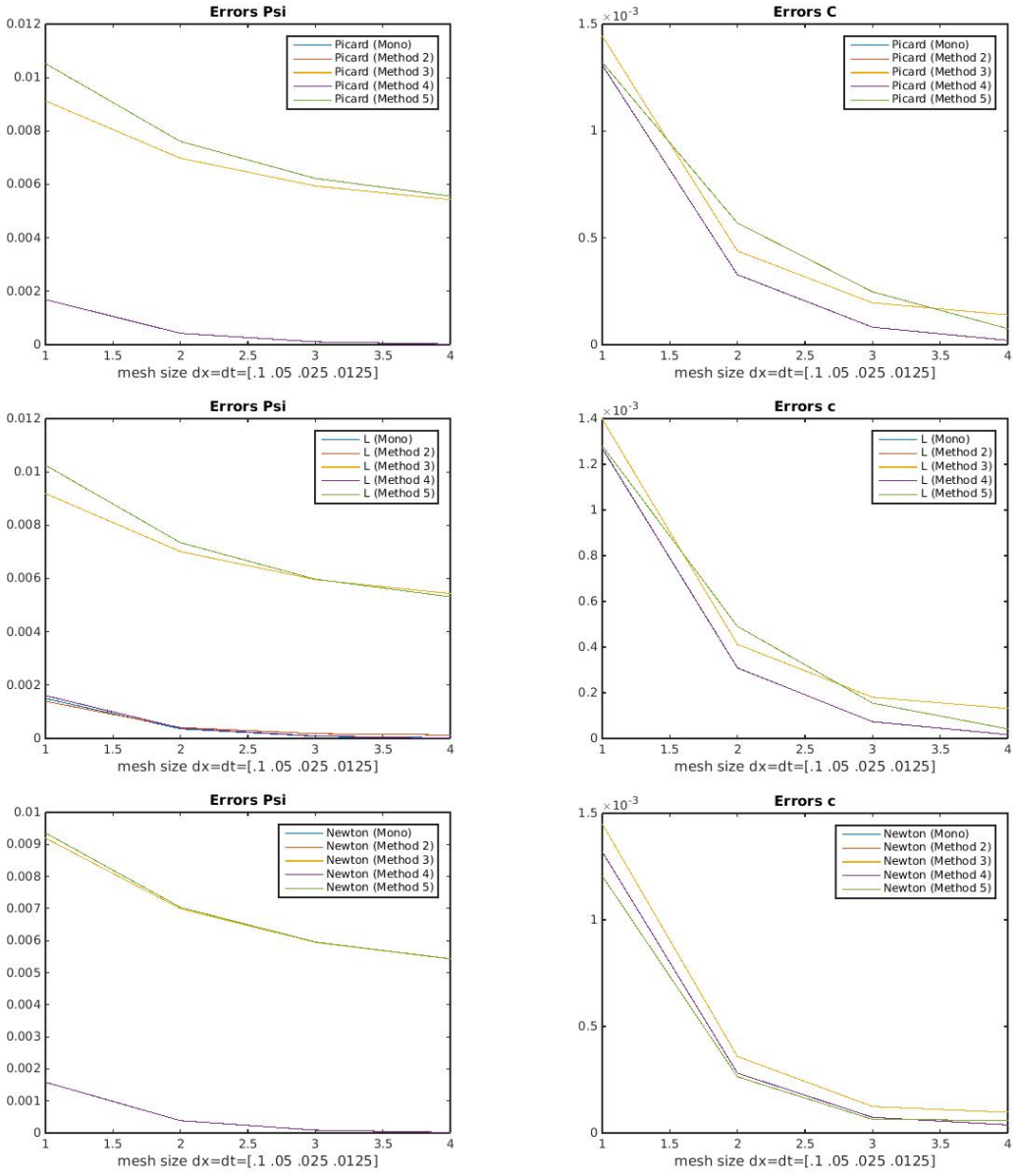


Figure 6.1: Comparison of different Methods through the Numerical Error

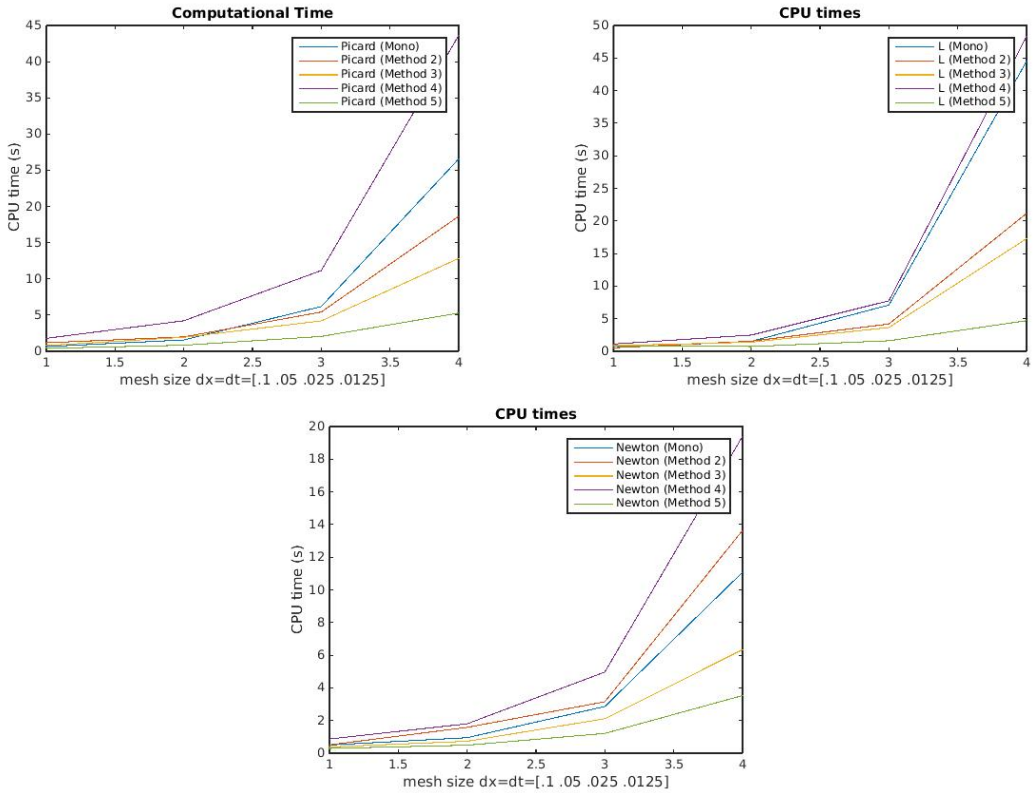


Figure 6.2: Comparison of different Methods through Computational Time

The plots into Fig 6.1 show as our choices, *Methods 1* and *2*, have been appropriate. Observing the numerical errors we can conclude that *Method 5* is the worst followed by *Method 3*. More interesting is the comparison between the *Methods 2* and *4*, our aim is to show that our approach can represent a valid alternative to the most commonly used *Method 4*. Looking at the computational time it is evident as our scheme is faster than the other one, this is justify by the fact that the linearization schemes are applied only once on each equation to obtain the next values $\Psi^{n+1,j+1}$ and $c^{n+1,j+1}$.

To better understand the different CPU times it is useful to observe the number of iterations for both of the schemes, such results are presented into Tab 6.1, 6.2 and 6.3. The third and sixth columns represent the number of iteration used by the while loop, defined into *line 1* of the code showed above, for both methods 2 and 4. The last two columns are instead the number of iteration for the while loops for the Richards and transport equation, loops present only in Method 3 and Method 4 and defined in the *lines 5* and *19*. Without these two last columns one could conclude that Method 4 should be faster than Method 2, because the number of iterations in the common loop is clearly smaller. Considering instead also all the other iterations it is evident why Method 2 results faster than Method 4.

Table 6.1: Number iterations for Picard Methods

$\Delta x = \Delta t$	Mono	Method 2	Ψ Method 3	c Method 3	Method 4	Ψ Method 4	c Method 4
.1	10	8	8	6	5	24	16
.05	9	8	7	5	6	27	16
.025	9	8	7	5	6	27	16
.0125	9	8	7	4	7	29	17

Table 6.2: Number iterations for L Methods

$\Delta x = \Delta t$	Mono	Method 2	Ψ Method 3	c Method 3	Method 4	Ψ Method 4	c Method 4
.1	9	8	8	6	5	24	16
.05	8	8	7	5	6	27	16
.025	8	8	7	4	6	27	15
.0125	7	7	6	4	7	28	17

Table 6.3: Number iterations for Newton Methods

$\Delta x = \Delta t$	Mono	Method 2	Ψ Method 3	c Method 3	Method 4	Ψ Method 4	c Method 4
.1	4	5	4	3	5	12	11
.05	4	6	3	3	6	14	12
.025	4	6	3	3	6	14	12
.0125	4	7	3	3	7	16	14

The most interesting results is the error, form the plots above we can not notice any difference between the two methods. Tab 6.4 and 6.5, presenting the precise values of the computational errors, can help us, we can observe as the difference between the two approaches is barely notable and for this reason can be neglected.

Table 6.4: Pressure errors for Different Methods

$\Delta x = \Delta t$	Picard Method 2	Picard Method 4	L Method 2	L Method 4
.1	.001689508458540	.001689507598692	.001397533279725	.001609887270517
.05	.000429190791296	.000429190507025	.000412102931704	.000390000719968
.025	.000107731266710	.000107730585812	.000180405943387	.000089590395172
.0125	.000026959688117	.000026959443881	.000131403037815	.000019905332965

$\Delta x = \Delta t$	Newton Method 2	Newton Method 4
.1	0.001628048541982	0.001628048955889
.05	0.000397125821946	0.000397126102005
.025	0.000092181128907	0.000092181085043
.0125	0.000020097948905	0.000020098480960

Table 6.5: Concentration errors for Different Methods

$\Delta x = \Delta t$	Picard Method 2	Picard Method 4	L Method 2	L Method 4
.1	.001308837171125	.001308837110575	.001270844956482	.001270844938159
.05	.000327964630603	.000327964609374	.000309533778124	.000309533781492
.025	.000082065700205	.000082065690940	.000073449412170	.000073449419705
.0125	.000020524253931	.000020524256372	.000016927235965	.000016927236357

$\Delta x = \Delta t$	Newton Method 2	Newton Method 4
.1	0.001319521562785	0.001319521563571
.05	0.000281011734219	0.000281011734207
.025	0.000073309367797	0.000073309368151
.0125	0.000038163304848	0.000038163305318

For this particular example *Method 2* seems to be a valid alternative to *Method 4*, it is faster and their accuracy is extremely close.

A singular example can not be enough to conclude that the code used in this thesis is actually better, but we can try to consolidate such supposition presenting a second example. The second problem, again regarding the Richards and transport equations has been obtained using two different analytical solutions, precisely: $\Psi_a(x,t) = c_a(c,t) = x^2(1-x)^2t^2$. For this new problem we will again present the different results obtained applying each of the approaches presented in this chapter.

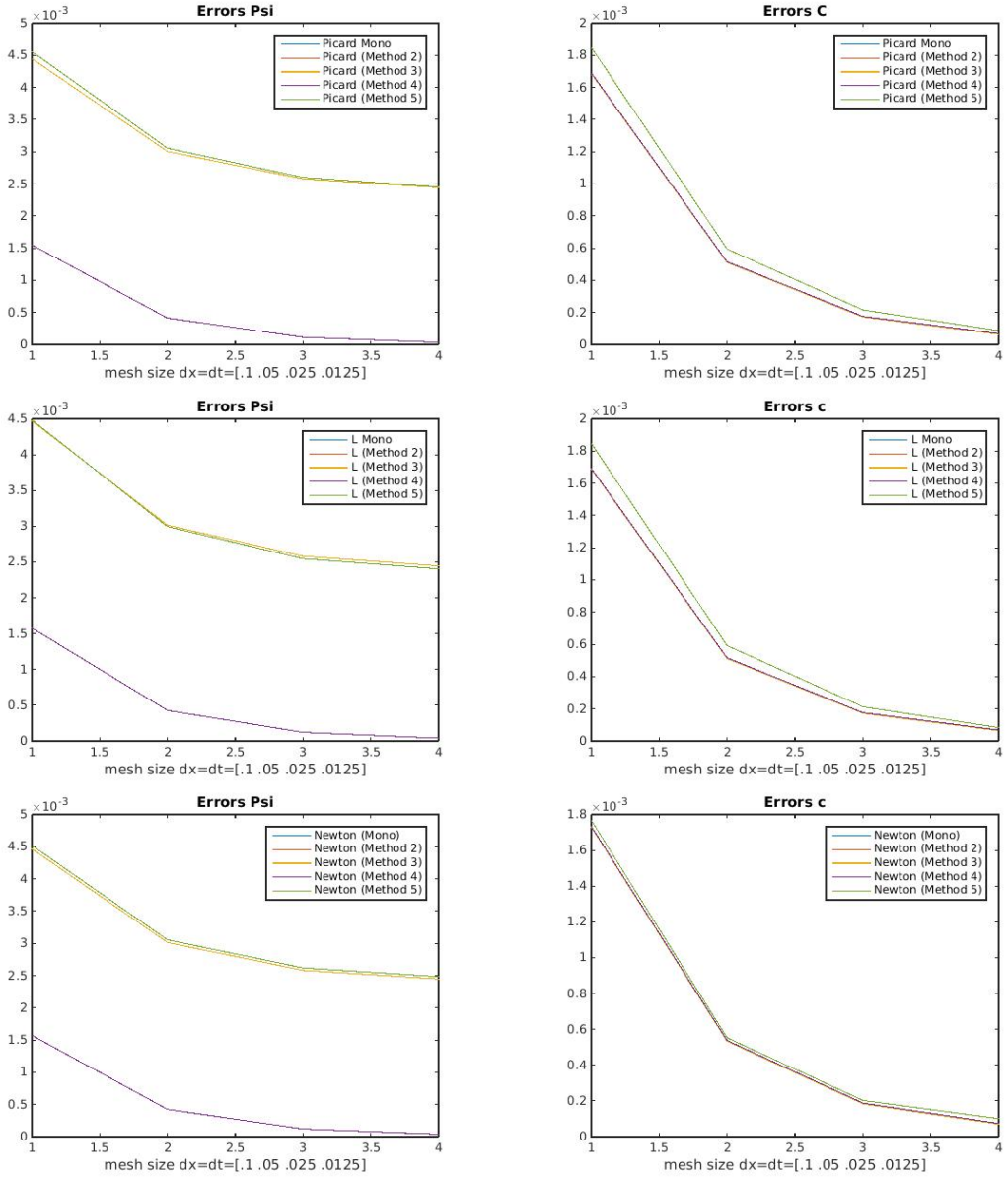


Figure 6.3: Comparison of different Methods through Numerical Error (Example 2)

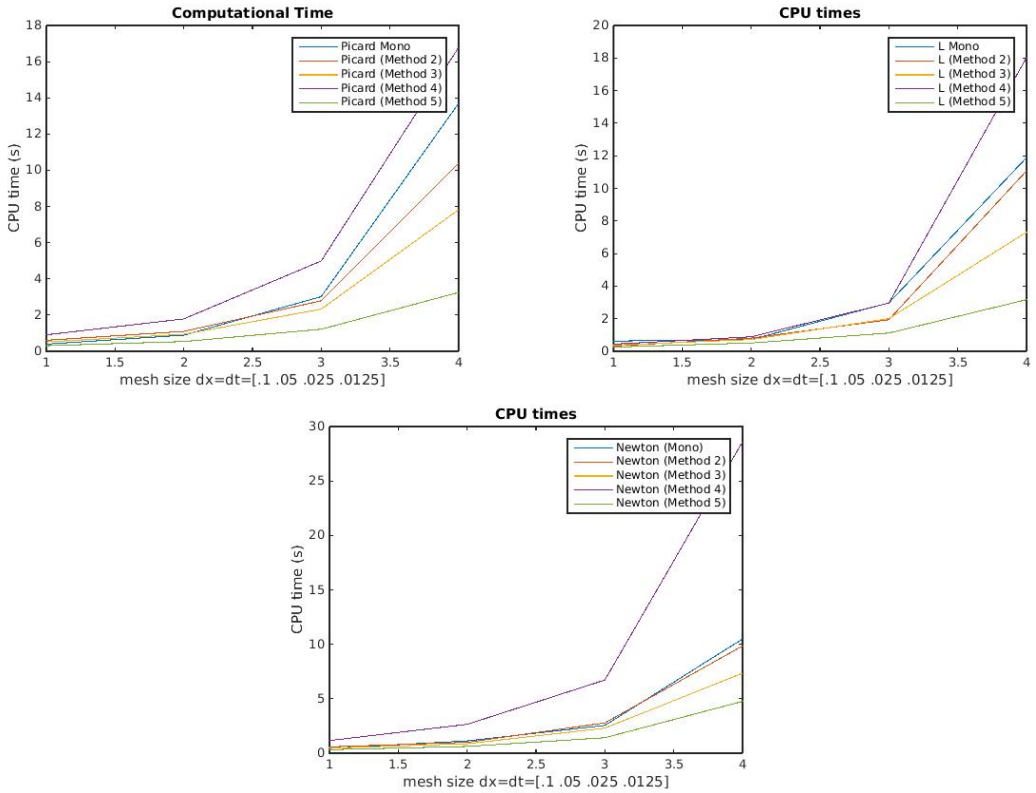


Figure 6.4: Comparison of different Methods through Computational Time (Example 2)

Fig 6.3 shows results coherent with the one obtained for the first example. We have again that the monolithic approach and Method 2 represent a valid alternative to Method 4. The errors in the tables 6.6 and 6.7 can again be used to show as the results obtained with the *Method 4* are very close to the ones obtain with *Method 2* and the differences can be neglected.

Table 6.6: Pressure errors for Different Methods Example 2

$\Delta x = \Delta t$	Picard Method 2	Picard Method 4	L Method 2	L Method 4
.1	.001551174795712	.001551174782674	.001579181331281	.001579181319024
.05	.000413789641805	.000413789584797	.000428131005687	.000428131060814
.025	.000115007345564	.000115007355729	.000122524066147	.000122524054468
.0125	.000034938701511	.000034938700777	.000038861891204	.000038861866367

$\Delta x = \Delta t$	Newton Method 2	Newton Method 4
.1	0.001573282460211	0.001573282433022
.05	0.000425269518171	0.000425269198661
.025	0.000121162339339	0.000121162310885
.0125	0.000038248710819	0.000038248662678

Table 6.7: Concentration errors for Different Methods Example 2

$\Delta x = \Delta t$	Picard Method 2	Picard Method 4	L Method 2	L Method 4
.1	.001690485882842	.001690485879390	.001693122882090	.001693122880102
.05	.000515222420786	.000515222419505	.000516530604538	.000516530603638
.025	.000176041269641	.000176041269288	.000176718107111	.000176718107139
.0125	.000068360288033	.000068360287948	.000068708491932	.000068708491654

$\Delta x = \Delta t$	Newton Method 2	Newton Method 4
.1	0.001733945476034	0.001733945475316
.05	0.000537733928626	0.000537733925132
.025	0.000187879385127	0.000187879378699
.0125	0.000074486028210	0.000074486034264

We can again observe as the number of iteration for the main loop coupling the two equations is smaller for the fourth approach but, as before, we must consider also all the iteration used on each equation and these make *Method 2* faster.

Table 6.8: Number iterations for Picard Methods Example 2

$\Delta x = \Delta t$	Mono	Method 2	Ψ Method 3	c Method 3	Method 4	Ψ Method 4	c Method 4
.1	6	6	5	4	4	14	9
.05	6	6	5	4	4	14	9
.025	5	6	5	4	5	16	11
.0125	5	6	5	4	5	16	11

Table 6.9: Number iterations for L Methods Example 2

$\Delta x = \Delta t$	Mono	Method 2	Ψ Method 3	c Method 3	Method 4	Ψ Method 4	c Method 4
.1	6	6	5	4	4	14	9
.05	6	6	5	4	4	14	9
.025	5	6	5	4	5	16	11
.0125	5	7	5	4	5	19	11

Table 6.10: Number iterations for Newton Methods Example 2

$\Delta x = \Delta t$	Mono	Method 2	Ψ Method 3	c Method 3	Method 4	Ψ Method 4	c Method 4
.1	4	5	3	4	4	9	9
.05	4	5	3	4	4	9	9
.025	3	6	3	5	5	11	12
.0125	3	6	3	5	5	11	12

Considering the results of these two different academic examples we can officially conclude that the Method 2 is a valid alternative to Method 4, and for this reason, in the following we will proceed to compare more into the details the monolithic approach and the fully implicit approach (Method 2) for the different linearization schemes used in the thesis. The results presented in the following regard the first academic example.

For all the computations we have used *Matlab* running on a Dell laptop with the following specs: Quad-core, Core i5 Processor with 8.00 GB of Ram.

6.2 Computational times

The graph 6.5 presents the different CPU times required by each linearization scheme to solve our problem, such results have been obtained using the CPUtime command of *Matlab*, we also present the precise values of the CPU time in the table 6.11.

It is possible to observe as the results, presented in the plot 6.5 and the table 6.11, show that the *Monolithic L-scheme* is the slowest and the *Newton method* is the fastest.

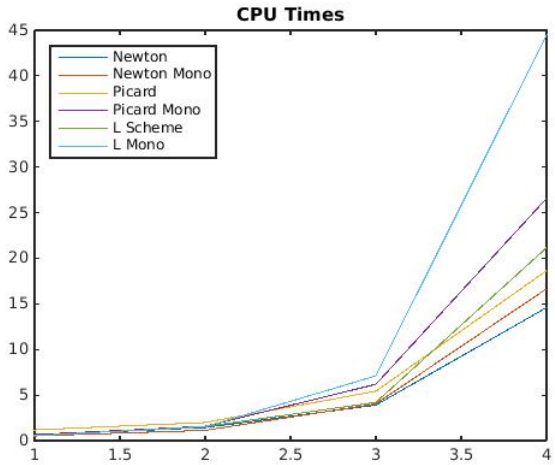


Figure 6.5: figure
CPU times obtained for the meshes given by dx=dt=[$\frac{1}{10}$, $\frac{1}{20}$, $\frac{1}{40}$, $\frac{1}{80}$]

Table 6.11: CPU times in seconds for each scheme, Mono=Monolithic

$\Delta x = \Delta t$	Newton	Newton Mono	Picard	Picard Mono	L Scheme	L Mono
.1	.6000	.5200	1.1800	.6500	.5400	.5500
.05	14400	1.1300	1.9900	1.5700	1.5500	1.5200
.025	3.900	4.0300	5.4200	6.1800	4.1900	7.1000
.0125	14.5900	16.6500	18.6400	26.5600	21.1500	44.5000

6.3 Number of Iterations

In this subsection we will compare the total number of iterations for each scheme having a better comprehension of how the codes work. Such results will be presented in the table 6.12. We can observe as the Newton method presents the lowest number of iteration for each loop, this justify, at least in part, the velocity of the scheme. From the data of the table we can also notice as the number of iterations for the L scheme is smaller than the one for the Picard, as seen before this doesn't make the scheme faster.

Table 6.12: Number iterations]

$\Delta x = \Delta t$	Newton	Newton Mono	Picard	Picard Mono	L Scheme	L Mono
.1	5	4	8	10	9	8
.05	6	4	8	9	8	8
.025	6	4	8	9	8	8
.0125	7	4	8	9	7	7

6.4 Condition numbers

It is now time to present the condition numbers for the monolithic schemes, for each of them we can build a matrix and observe its condition number. High condition numbers give unstable systems and this slows down the code, in some cases gives even wrong results. The condition numbers are presented into Figure 6.6

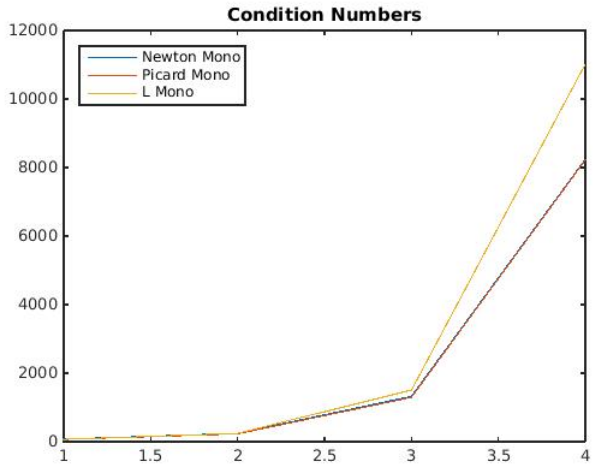


Figure 6.6: figure
Condition Numbers for Newton, Picard-Mono and L-Mono obtained for the meshes given by dx=dt=$[\frac{1}{10}, \frac{1}{20}, \frac{1}{40}, \frac{1}{80}]$

From this plot we can observe as the L scheme presents the highest condition number, this justify the fact that, even requiring less iterations than the Picard, it results slower. We can also notice as the condition numbers of the systems related to the Newton and Picard methods are extremely similar,we can not notice any evident difference from the graph. This similitude means that roughly each iteration is computed at the same time for both of the scheme and, having Newton less iteration, it results clearly faster.

Chapter 7

A real case study: the salinity problem

In this chapter we are going to use all the codes developed for the *Richards* and *transport equations* applying them to a real case, the *salinity problem*. We will also compare our results with the one in [1], using it as a benchmark, to ensure that our codes are correct. To start we need to look back at the Richards formulation and slightly modify it, introducing more realistic parameters, one of such quantities is the *diffusion-dispersion coefficient* D, considered constant in the previous formulations. A more realistic expression for θD is:

$$\theta D = \xi u + D_0 \frac{\theta^{10/3}}{\Phi}$$

with ξ the dispersivity, D_0 the free solution diffusion coefficient and Φ the porosity. Values for these and the others parameters have been taken from [1] and [32], in the second, in particular, it is studied how different kind of salts can influence the surface tension of the water.

When the surface tension is not constant the pressure head Ψ must be rescaled, precisely by the factor $\gamma_0(c_0)/\gamma(c)$ [1], it is then fundamental to find an expression that relates γ to c. Many article cover this subject, we based our formulation on [32] where a linear relation is expressed as follow

$$\gamma(c) = \zeta c + \gamma_0 \tag{7.1}$$

with ζ a fitting parameter and γ_0 the reference surface tension. It has been observed that the presence of the salt into the water phase brings to an increase of the surface tension, such increase is linearly proportional to the concentration and the parameter ζ changes considering different salts. We are going to study two different salts, precisely the *sodium perchlorate* ($NaClO_4$) and the *sodium sulfate* (Na_2SO_4).

From a physical point of view the salinity problem plays an important role in agriculture where the salt dissolved in the water, both irrigation and rain water, infiltrates in the soil. An high concentration of salt makes more difficult for plants to absorb nutrient and also, we have that an higher surface tension will slow down the flow of the water in the ground risking to create, after a while, marshes.

7.1 Physical studies

We give four different examples, in the first the surface tension will be considered constant, we can use this case for comparison with the following two where the different salts will be studied. The fourth and last case will be taken from [1] and will be used to verify that our code is correct.

Let's proceed giving the parameters for our examples into the table 7.1, it is possible to observe that not only the constants are different from the one in the previous chapters but also the domains, studying a physical problem a more realistic domain is required. We are modeling the first meter of soil under the surface observing how the flow evolves during ten hours. Boundaries and initial conditions

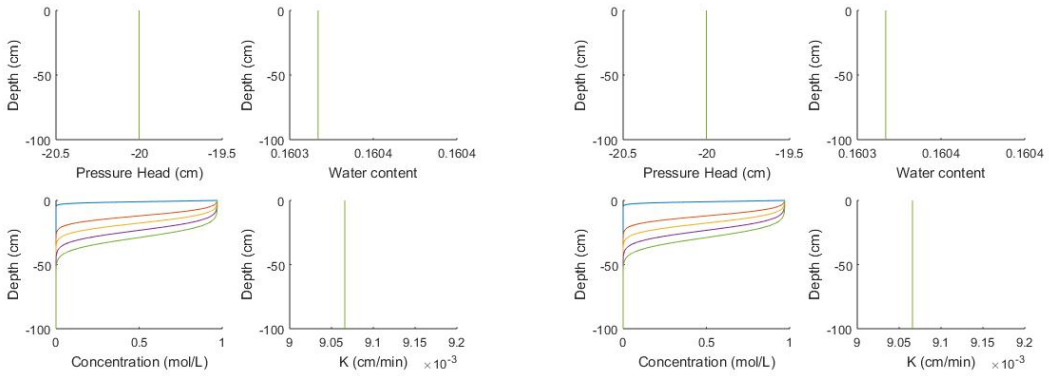
Table 7.1: Parameters

	Case 1	Case 2	Case 3	Case 4
T_{max}	10 h	10 h	10 h	10 h
Δt	.2 min	.2 min	.2 min	.2 min
Ω	[-100cm ,0cm]	[-100cm,0cm]	[-100cm,0cm]	[-100cm,0cm]
Δx	.5 cm	.5 cm	.5 cm	.5 cm
Van Genucthen Parameters				
θ_s	.3	.3	.3	.3
θ_r	.05	.05	.05	.05
n	2.43	2.43	2.43	2.43
l	.31	.31	.31	.31
α	.0551	.0551	.0551	.0551
Surface tension parameters				
ζ	-	0.5971	2.4901	-
$\sigma 0$	72.62 mN/m	72.98 mn/m	72.98 mn/m	72.62mN/m
a	-	-	-	.5936
b	-	-	-	.4745
K_s	6.0e-02 cm/min	6.0e-02 cm/min	6.0e-02 cm/min	6.0e-02 cm/min
ξ	1.0 cm	1.0 cm	1.0 cm	1.0 cm
D_0	6.0e-04	6.0e-04	6.0e-04	6.0e-04
Convergent coefficient				
ϵ	e-07	e-07	e-07	e-07

are also taken from [1], we require initial pressure $\Psi(x,0) = -20cm$ and same value for both inlet and outlet. Regarding the concentration we have initial condition $c(x,0) = 0\ mol/L$, inlet $c(0,t) = .97\ mol/L$ and for the outlet a *Neumann* condition is required. In the rest of the thesis we always had *Dirichlet* boundary conditions, a precise value, for example for the concentration, was required; in this case instead we want the gradient of c to be equal to zero at the outlet, precisely $\nabla c\ |_{(-100,t)} = 0$.

Case 1

This example has been used to present a steady flow base case in which there will be no changes in the surface tension, for this reason we will use the non scaled expression of θ and the parameters presented in the first column of 7.1. As expected, the results in the figure 7.1, show constant values for: pressure, water content and hydraulic conductivity in both space and time.



(a) Picard

(b) L Scheme

(c) Picard Mono

(d) L Mono

Figure 7.1: Linearization Schemes Case 1

We plotted the profile of such quantities at different times, precisely $t = 10min$, $t = 50min$, $t = 100min$, $t = 200min$, $t = 300min$, $t = 400min$, $t = 500min$ and $t = 10h$. It is not possible to observe any difference between the results obtained for the different linearizations schemes. The results obtained for the case 1 can be compared with [1].

These are the results we obtain if we neglect the influence of the concentration on the surface tension, we can now use these plots comparing them with the next examples.

Case 2

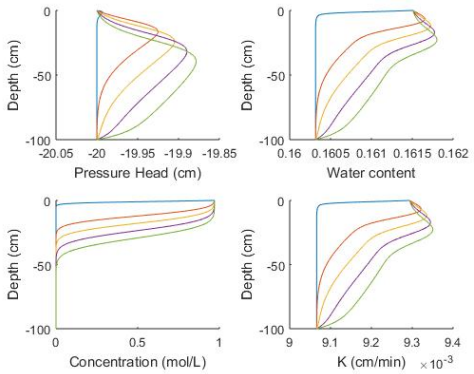
We will now consider the main problem of this chapter, introducing a changing surface tension due to the presence of an external substance, in this case, *sodium perchlorate* ($NaClO_4$). To study these changes we must recall the equation 7.1 and use the parameters from the second column of the table 7.1. We then obtain that the changing rate for the salinity problem is:

$$\frac{\sigma}{\sigma_0} = \frac{\varsigma}{\sigma_0} c + 1 \tag{7.2}$$

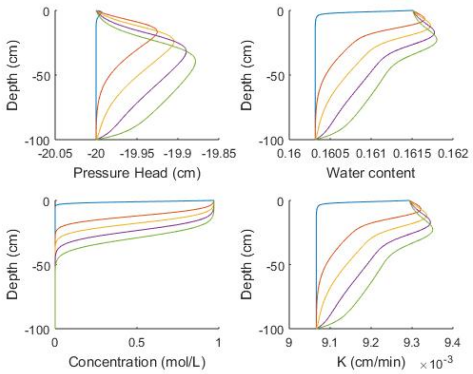
substituting the equation above into 4.1 we obtain the new expression for the water content

$$\Theta = \frac{\Theta_s - \Theta_r}{(1+ \mid \alpha(\frac{\varsigma}{\sigma_0}c + 1)^{-1}\Psi \mid^n)^m} + \Theta_r \tag{7.3}$$

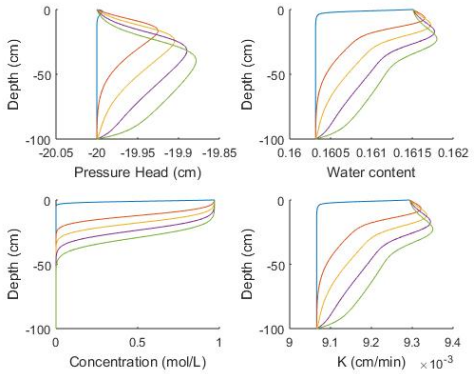
Using now the same initial and boundary conditions for both Ψ and c as in the first case we obtain the results presented into Figure 7.2. It is interesting a comparison with the results from the first case, it is evident as an increase in the surface tension has complicated the problem, we do not have any more constant values for all Ψ, θ and K but we can now observe clear changes in all of them. These results are coherent with what we were expecting, it is known that with an higher surface tension we obtain higher pressure. The next example will be used to study a different salt for which the increase in the surface tension is more evident.



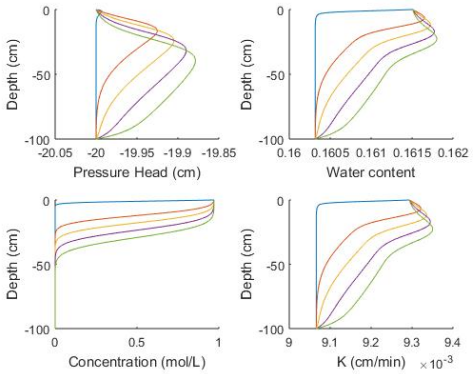
(a) Picard



(b) L Scheme



(c) Picard Mono



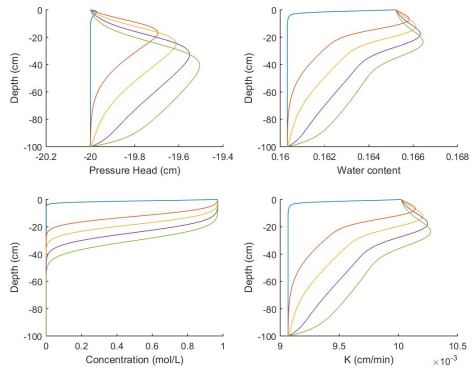
(d) L Mono

Figure 7.2: Linearization Schemes Case 2

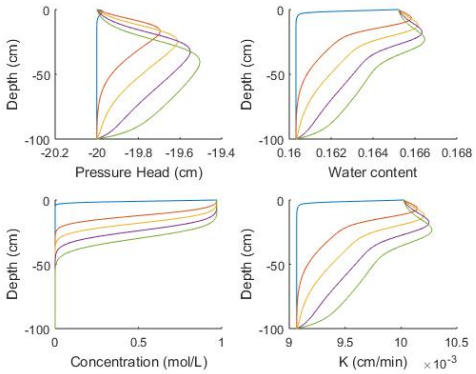
Case 3

In this example the *sodium sulfate* (Na_2SO_4) has been studied. As for the previous case the surface is linearly dependent from the concentration so that the same equation 7.3 can be implemented but different coefficients ζ and σ_0 are required. The results are again presented through the Figure 7.3 where no evident difference

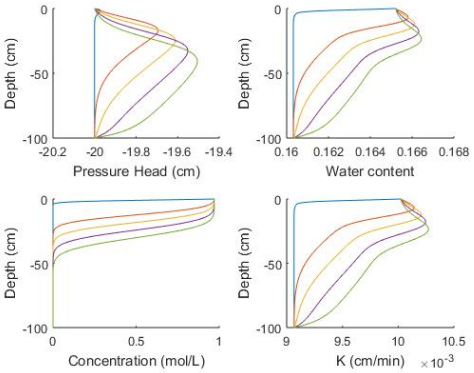
between the linearization schemes can be observed. It is evident an increase in the pressure, as said before the changes in the surface tension are now more substantial.



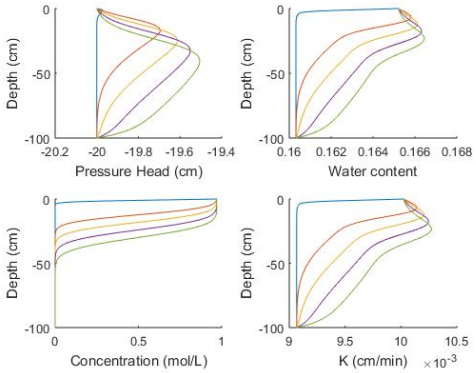
(a) Picard



(b) L Scheme



(c) Picard Mono



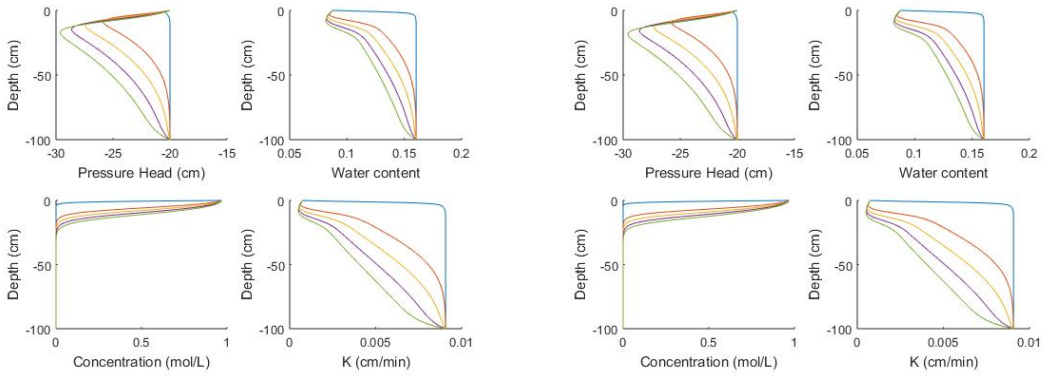
(d) L Mono

Figure 7.3: Linearization Schemes Case 3

Case 4

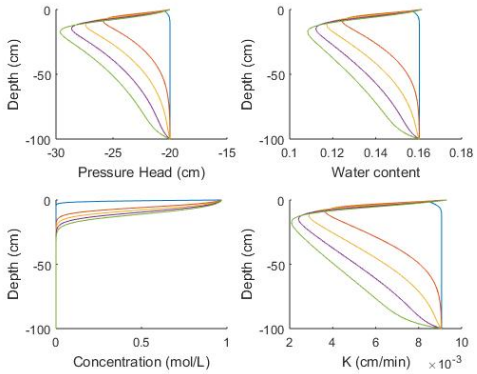
This last example has been taken from [1], and it has been used as benchmark to verify that the codes developed were correct. We ran such codes and we obtained the same results as in [1], this gives us the confidence to say that the codes are working properly. For this case a decreasing surface tension, due to the presence of butanol in the water phase, is studied and a more complex expression is given:

$$\sigma = \sigma_0 - \sigma_0 b \ln\left(\frac{c}{a} + 1\right) \tag{7.4}$$

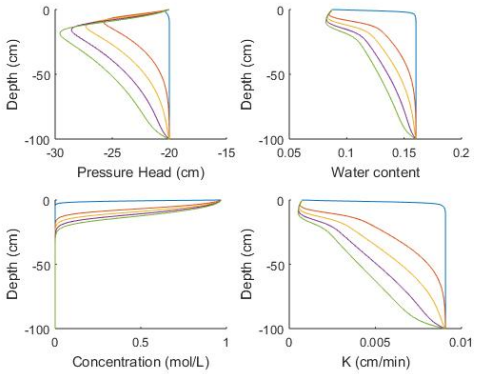


(a) Picard

(b) L Scheme



(c) Picard Mono



(d) L Mono

Figure 7.4: Linearization Schemes Case 4

such equation can be derived from [1].

Looking at Figure 7.4 we can observe results completely different compared to the ones obtained in the previous cases. The other examples were studying two different salts which were increasing the surface tension, for this case, instead, the surface tension is reduced.

7.2 Comparison of different iterative schemes

As in the previous chapter we are going to compare the five different approaches to the linearization scheme presented in these cases. We will plot the different computational times obtained with the five schemes for the Picard and L schemes applied to the salinity problem. This section will be a further confirmation that the approaches used in this thesis are, not only valid, but also better than the one usually implemented for this kind of studies. Unfortunately for this problem we do not have an analytical solution and we can not study the numerical errors, we can still consider the results to be correct because they have been visually compared with the one of [1].

Let's now proceed presenting the computational times, Fig 7.5, and the number of iterations, Tab 7.2 and 7.3.

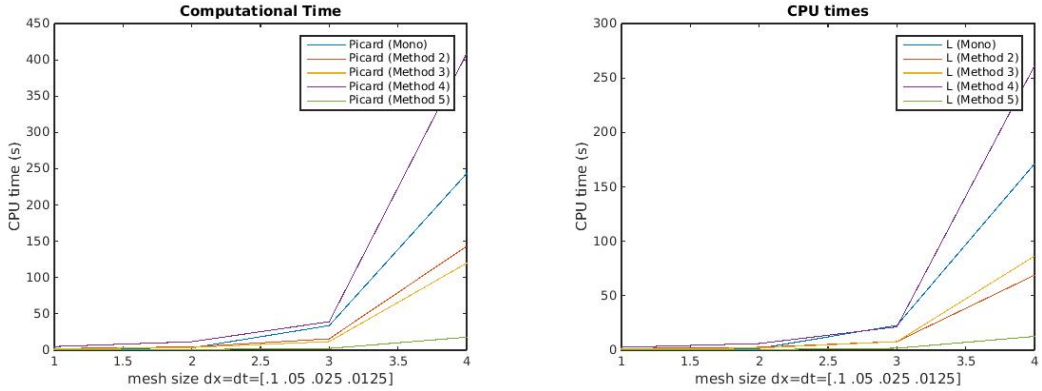


Figure 7.5: Comparison of different Methods through Computational Time

Table 7.2: Number iterations for Picard Methods

$\Delta x = \Delta t$	Mono	Method 2	Ψ Method 3	c Method 3	Method 4	Ψ Method 4	c Method 4
.1	6	5	4	4	4	14	10
.05	5	5	4	4	4	14	10
.025	8	6	6	3	4	21	9
.0125	14	6	10	3	4	32	8

Table 7.3: Number iterations for L Scheme

$\Delta x = \Delta t$	Mono	Method 2	Ψ Method 3	c Method 3	Method 4	Ψ Method 4	c Method 4
.1	6	5	4	4	4	14	10
.05	4	5	3	4	4	14	10
.025	8	5	6	3	3	20	6
.0125	14	3	10	3	3	31	6

Exactly as for the cases studied in the previous chapter the approach number *2* remains a valid alternative to *method 4*, especially for the evident improvement into the computational time. It is interesting also to observe as the number of iteration

in the main loop, coupling the Richards and transport equations, is smaller for the *Method 4*. We must again consider all the other iterations, due to the 2 separate loops presented in the last columns of both Tab 7.2 and 7.3, which justify the slowness of the fourth approach.

Chapter 8

Conclusions

In this thesis we considered first two-phase flow and transport in porous media and after a specific case of groundwater flow described by the Richards and transport equations. We deeply investigated the coupled system of non-linear *PDEs*, given by the second problem, studying different linearization schemes, precisely: the *L-scheme*, the *modified Picard* and the *Newton method*.

We also concentrated on how to implement such schemes in an efficient way, considering the coupling aspect of our problem. For each equation we implemented an inner loop to ensure that the non-linear terms were treated properly. We presented five different approaches differing on how such loops were defined.

We concluded as the approach mainly used in this thesis, here defined as the *fully implicit iterative* method, represents a valid alternative to the most common solving formulation (presented by *method 4*). The method seems to be equally accurate and faster, with an evident reduction of the number of iterations for each loop being achieved. We numerically compared the different approaches on multiple problems to ensure that our results are realistic. We used three academic examples, two academic examples, admitting an analytical solutions and a real case study.

For the real case study, we used a benchmark problem, the salinity problem. It is a classical benchmark for the one dimensional Richards and transport problems. The iterative methods performed similarly for the real case study and the academic problem. We concluded that, in our opinion, the best choice for solving implicitly, fully coupled flow and transport in porous media is Method 2 (called the fully implicit method).

Bibliography

[1] J. Smith, R. Gillham, *The effect of concentration-dependent surface tension on the flow of water and transport of dissolved organic compounds: A pressure head-based formulation and numerical model, Water Resources Research 31(3):795*, 1994.

[2] J. Nordbotten, M. Celia, *Geological Storage of CO_2, Wiley*, 2012.

[3] K. Choi, M. Jackson, G. Hampson, A. Jones, A. Reynolds, *Predicting the impact of sedimentological heterogeneity on gas–oil and water–oil displacements: fluvio-deltaic Pereriv Suite Reservoir, Azeri–Chirag–Gunashli Oilfield, South Caspian Basin, Petroleum Geoscience 17(2):143-163*, 2011.

[4] Petro wiki - capillary pressure, `http://petrowiki.org/Capillary_pressure`.

[5] S. Hassanizadeh, M. Celia, H. Dahle, *Dynamic Effect in the Capillary Pressure–Saturation Relationship and its Impacts on Unsaturated Flow, Vadose Zone Journal 1:38–57* , 2001.

[6] A. Manda, M. Gross, *Estimating aquifer-scale porosity and the REV for karst limestones using GIS-based spatial analysis, GSA Special Papers 404:177-189* , 2006.

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

[8] K. Skiftestad, *Numerical Modelling of Microbial Enhanced Oil Recovery with Focus on Dynamic Effects: An Iterative Approach, Master Thesis UiB*, 2015.

[9] R. Eymard, *Finite volume method, Scholarpedia 5(6):9835*, 2010.

[10] J. Stoer, R. Bulirsch, *Introduction to numerical analysis, Springer*, 1993.

[11] M. Celia, E. Bouloutas, *A General Mass-Conservative Numerical Solution for the Unsaturated Flow Equation, Water Resources Research 26(7):1483-1496*, 1900.

[12] P. Knabner, S. Bitterlich, R. Iza Teran, A. Prechtel, E. Schneid, *Influence of Surfactants on Spreading of Contaminants and Soil Remediation, Springer*, 2003.

[13] F. Radu, W. Wang, *Convergence analysis for mixed finite element scheme for flow in strictly unsaturated porous media, Nonlin. Anal. Serie B: Real World Appl., DOI:10.1016/j.nonrwa.2011.05.003*, 2011.

[14] MathWorks - Documentation, http://it.mathworks.com/help/matlab/matlab_prog/measure-performance-of-your-program.html.

[15] F. List, F. Radu, *A study on iterative methods for solving Richards' equation, Comput. Geosci. 20:341*, 2015.

[16] I. Pop, F. Radu, P. Knabner, *Mixed finite elements for the Richards' equation: linearization procedure, Journal of computational and applied mathematics 168(1):365-373*, 2004.

[17] I. Pop, F. Radu, P. Knabner, *On the convergence of the Newton method for the mixed finite element discretization of a class of degenerate parabolic equation. Numerical Mathematics and Advanced Applications, A. Bermudez de Castro et al.(editors), Springer*, 2006.

[18] I. Pop, F. Radu, K. Kumar, *Convergence analysis for a conformal discretization of a model for precipitation and dissolution in porous media, Numerische Mathematik 127(4):715*, 2014.

[19] I. Pop, F. Radu, K. Kumar, *Convergence analysis of mixed numerical schemes for reactive flow in a porous medium, SIAM Journal on Numerical Analysis 51(4):2283-2308*, 2013.

[20] I. Pop, F. Radu, S. Attinger, *Analysis of an Euler implicit - mixed finite element scheme for reactive solute transport in porous media, Numerical Methods for Partial Differential Equations 26(2):320-344*, 2010.

[21] F. Lehman, P. Ackerer, *Comparison of iterative methods for improved solutions of the fluid flow equation in partially saturated porous media, P. Transport in Porous Media 31:275*, 1998.

[22] J. Smith, R. Gillham, *Effects of solute concentration-dependent surface tension on unsaturated flow: Laboratory sand column experiments, Water Resource Research 35(4):973–982*, 1999.

[23] N. Bergamashi, M. Putti, *Mixed finite elements and Newton-type linearizations for the solution of Richards' equation, Int. J. Num. Meth. Engng. 45:1025-1046*, 1900.

[24] J. Reddy,, *An Introduction to the finite Element Method, McGraw-Hill Mechanical Engineering*, 2006.

[25] F. Radu, J. Nordbotten, I. Pop, 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*, 2015.

[26] F. Radu, *Convergent mass conservative schemes for flow and reactive solute transport in variably saturated porous media, arXiv:1512.08387*, 2013.

[27] F. Radu, *Mixed finite element discretization of Richards' equation: error analysis and application to realistic infiltration problems, PhD Thesis, University of Erlangen-Nürnberg, Germany*, 2014.

[28] F. Radu, S. Attinger, M. Bause, A. Prechtel, *A mixed hybrid finite element discretization scheme for reactive transport in porous media, Numerical mathematics and advanced applications, 513-520*, 2008.

[29] F. Radu, J. Nordbotten, I. Pop, K. Kumar, *A convergent mass conservative numerical scheme based on mixed finite elements for two-phase flow in porous media, arXiv:1512.08387* , 2015.

[30] M. Slodicka, *A robust and efficient linearization scheme for doubly non-linear and degenerate parabolic problems arising in flow in porous media, SIAM J. Sci. Comput., 23(5), 1593–1614.*, 2002.

[31] M. van Genuchten, *A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils, Soil Science Society of America Journal 44(5)*, 1980.

[32] D. Husseini, *Effects of Anions acids on Surface Tension of Water, Undergraduate Research at JMU Scholarly Commons*, 2015.

[33] T. Arbogast, M. Wheeler, N. Zhang, *A nonlinear mixed finite element method for a degenerate parabolic equation arising in flow in porous media, SIAM J. Numer. Anal. 33:1669–1687*, 1996.

[34] T. Arbogast, M. Obeyesekere, M. Wheeler, *Numerical methods for the simulation of flow in root-soil systems, SIAM J. Num. Anal. 30:1677-1702*, 1993.

[35] I. Pop, *Error estimates for a time discretization method for the Richards' equation, Comput. Geosci. 6:141–160*, 2002.

[36] I. Pop, M. Sepulveda, F. Radu, O. Villagran, *Error estimates for the finite volume discretization for the porous medium equation, J. Comput. Appl. Math. 234: 2135–2142*, 2010.

[37] C. Dawson, V. Aizinger, *Analysis of an upwind-mixed finite element method for nonlinear contaminant transport equations, SIAM J. Numer. Anal. 35:1709–1724*, 1998.

[38] C. Dawson, V. Aizinger, *Upwind-mixed methods for transport equations, Comput. Geosci. 3:93-110*, 1999.

[39] R. Eymard, M. Gutnic, D. Hilhorst, *The finite volume method for Richards equation, Comput. Geosci. 3:256–294*, 1999.

[40] R. Eymard, D. Hilhorst, M. Vohral IK, *A combined finite volume-nonconforming/mixed-hybrid finite element scheme for degenerate parabolic problems, Numer. Math. 105:73–131*, 2006.

[41] F. Brunner, F.Radu, M. Bause, P. Knabner, *Optimal order convergence of a modified BDM1 mixed finite element scheme for reactive transport in porous media, Adv. Water Resour. 35:163–171*, 2012.

[42] F. Brunner, F.Radu, P. Knabner, *Analysis of an upwind-mixed hybrid finite element method for transport problems, SIAM J. Num. Anal. 52 (1):83-102*, 2014.

[43] R. Klausen, F. Radu, P. Knabner, *Optimal order convergence of a modified BDM1 mixed finite element scheme for reactive transport in porous media, Int. J. for Numer. Meth. Fluids 58:1327–1351*, 2008.

[44] E. Schneid, P. Knabner, F. Radu, *A priori error estimates for a mixed finite element discretization of the Richards' equation, Numer. Math. 98:353–370*, 2004.

[45] F. Radu, I. Pop, P. Knabner, *Order of convergence estimates for an Euler implicit, mixed finite element discretization of Richards' equation, SIAM J. Numer. Anal. 42:1452–1478*, 2004.

[46] F. Radu, I. Pop, P. Knabner, *Error estimates for a mixed finite element discretization of some degenerate parabolic equations, Numer. Math. 109:285–311*, 2008.

[47] F. Radu, N. Suciu, J. Hoffmann, A. Vogel, O. Kolditz, C-H Park, S. Attinger, *Accuracy of numerical simulations of contaminant transport in heterogeneous aquifers: a comparative study, Adv. Water Resour. 34:47–61*, 2011.

[48] F. Radu, A. Muntean, I. Pop, N. Suciu, O. Kolditz, *A mixed finite element discretization scheme for a concrete carbonation model with concentration-dependent porosity, J. Comput. Appl. Math. 246:74-85*, 2013.

[49] M. Vohralik, *A posteriori error estimates for lowest-order mixed finite element discretizations of convection-diffusion-reaction equations, SIAM J. Numer. Anal. 45:1570–1599*, 2007.

[50] C. Woodward, C. Dawson, *Analysis of expanded mixed finite element methods for a nonlinear parabolic equation modeling flow into variably saturated porous media, SIAM J. Numer. Anal. 37:701-724*, 2000.