Robust solvers for fully coupled transport and flow in saturated/unsaturated porous media

Davide Illiano

Thesis for the degree of Philosophiae Doctor (PhD)
University of Bergen, Norway
2021

UNIVERSITY OF BERGEN

Robust solvers for fully coupled transport and flow in saturated/unsaturated porous media

Davide Illiano

Thesis for the degree of Philosophiae Doctor (PhD)
at the University of Bergen

Date of defense: 29.03.2021

Year: 2021

Title: Robust solvers for fully coupled transport and flow in saturated/unsaturated porous media

Name: Davide Illiano

Print: Skipnes Kommunikasjon / University of Bergen

Preface

This dissertation is submitted as a partial fulfillment of the requirements for the degree of Doctor Philosophy (PhD) at the University of Bergen. The advisory committee has consisted of Florin Adrian Radu (University of Bergen), Iuliu Sorin Pop (University of Hasselt) and Knut-Andreas Lie (SINTEF).

The PhD project has been financially supported by VISTA, a collaboration between the Norwegian Academy of Science and Letters and Equinor, project number 6367, project name: *Adaptive model and solver simulation of enhanced.*

Acknowledgments

I have read many acknowledgments, and I have realized that it isn't easy to thank all the people who helped me during these years within a few lines. Anyhow I will try to do my best. First and foremost, I want to thank Florin, my main supervisor, and my guide during these years. I started working with you already during my master's, and in these six years, you have been of incredible help. Not only have you guided me in my work, but you have been able to calm me down whenever obstacles appeared along the road. For this and for much more, I want to say thank you.

I want to thank also Sorin, who, even though in Hassel and thus far away, has been a great guide and has often engaged with me in exciting discussions. Thanks to the Sintef research group. In particular, thanks to Olav and Knut-Andreas, who have shown me many of the secrets of MRST. Thanks to the Vista Program, which has supported my work in these years.

I was fortunate to be part of the Porous Media Group here in Bergen. Collaborating with you has been a fantastic experience; everyone has been incredibly supportive inside and outside the University. I want to thank every one of you, including those who have started a new chapter in Oslo. In particular, thanks to Erlend, Jakub, Manuel, David, Abay, Jhabriel, you have not been merely great colleagues but friends.

Grazie alla mia famiglia e ai miei amici in Italia, se oggi sono qui è anche grazie a voi. Lasciare la mia casa e tutti voi non è stato facile, ma sono felice di aver intrapreso tale percorso. So di avervi sempre accanto sebbene a innumerevoli chilometri di distanza. In particolare, grazie a Mamma e Papà, se non fosse stato per voi, e per l'importanza che avete sempre dato alla cultura e all'educazione, oggi non avreste tra le mani la mia tesi di Dottorato. Grazie mille, tutto questo è iniziato con il vostro aiuto.

Thanks to my Norwegian family and friends. Starting a new life in a different country is never easy, but you have helped me each step of the way. During these years, I have met some incredible people, and I found a new larger family. I should have probably written this in Norwegian, but we all know how it would have turned out.

Last but not least, I want to thank my wife Siri. During these years, you have been my rock and my source of light in the dark moments. You have heard me talking for hours about linearization schemes and solving algorithms. Even though I believe you have no idea of what they are, you have always been listening and supportive. I never met anyone as positive and optimistic as you. Without out you always by my side, these years

would have been much darker.

It has been an incredible journey, and now it has come to an end. Thank you all for having been with me along this path.

Davide Illiano

Abstract

Numerical simulations and laboratory studies are our main tools to comprehend better processes happening in the subsurface. The phenomena are modeled thanks to systems of partial differential equations (PDEs), which are extraordinarily complex to solve numerically due to their often highly nonlinear and tightly coupled character. After decades of research on new and improved solving algorithms, there is still a need for accurate and robust schemes. In this work, we investigate linearization schemes and splitting techniques for fully coupled flow and transport in porous media.

A particular case of multiphase flow in porous media, the study of water flow in variably saturated porous media, modeled by the Richards equation, is studied here. An external component, e.g., a surfactant, is transported by the water phases. The resulting system of equations is fully coupled and nonlinear.

In this work, we investigate three different linearization schemes, the *classical Newton method*, commonly used throughout the industry, the *modified Picard method*, and the *L-scheme*. Even though only linearly convergent, the latter appears to be the most robust scheme for some particular cases. The convergence of the L-scheme has also been studied theoretically. Both the Newton method and modified Picard are faster, in terms of numbers of iterations, but fail to converge in cases of unsaturated-saturated porous media or complex phenomena such as hysteresis effects.

The rate of convergence of the L-scheme can be improved by combining it with the Newton method or by using the Anderson acceleration. The scheme resulting by combining the L-scheme and the Newton method appears practically to be both quadratically and globally convergent. It requires fewer iterations than the L-scheme, it is more robust than the Newton method, and it also converges for larger time steps. Using larger time steps can considerably decrease the total number of iterations over the full simulation. Alternatively, by applying the Anderson acceleration, one avoids the computation of any derivatives, and thus the implementation of the algorithm is less invasive. The rate of convergence of the L-scheme depends on user-defined parameters. Finding the optimal L values can be tedious. Optimizing the Anderson acceleration is more straightforward, and the improvements obtained are remarkable.

The equations investigated in this work are not only characterized by nonlinear terms but are also fully coupled. The external component, dissolved into the water phase, directly influences the flow. We investigate two splitting approaches, the canonical nonlinear splitting, and an alter-

nate linearized splitting. We compare them in terms of the numbers of iterations required to achieve the convergence, and the condition numbers of the systems to be solved within each iteration. After all, the latter appears to be a better alternative; it requires fewer iterations and achieves equally accurate results. A monolithic or fully implicit formulation is also investigated. The solution algorithm is computationally slower than the splitting ones but appears to be more robust.

Finally, a global random walk approach for solving the coupled nonlinear problem is also studied. The scheme results in being free of numerical diffusion. Using vast numbers of particles, almost as many as the molecules involved in the reaction, it produces an intuitive representation of the process. The global random walk algorithms are explicit and thus often more straightforward than the typical finite volume/element schemes.

The models investigated in this work represent a particular case of two-phase flow in porous media. Still, we are confident that similar results, concerning the linearization schemes and solving algorithms, can also be achieved in the case of more canonical models.

List of papers

A **Iterative schemes for surfactant transport in porous media**
D. Illiano, I. S. Pop, F. A. Radu, *Computational Geosciences 2020,*
DOI: https://doi.org/10.1007/s10596-020-09949-2

B **An efficient numerical scheme for fully coupled flow and reactive transport in variably saturated porous media including dynamic capillary effects**
D. Illiano, I. S. Pop, F. A. Radu, *accepted author manuscript, Numerical Mathematics and Advanced Applications ENUMATH 2019,*
DOI: https://doi.org10.1007/978-3-030-55874-1_55

C **Efficient solvers for nonstandard flow and transport in unsaturated porous media**
D. Illiano, J. W. Both, I. S. Pop, F. A. Radu, *submitted manuscript*
arXiv:2012.14773

D **Random walk methods for flow and transport in unsaturated/saturated porous media**
N. Suciu, D. Illiano, A. Prechtel, F. A.Radu *submitted manuscript*
arXiv:2011.12889

Contents

Part II: Scientific results

Part I

Background

Chapter 1

Introduction

Modeling flow in porous media is a classic example of a mathematical approach used to represent complex physical phenomena. Numerous processes, which play fundamental roles in our society, are instances of flow in porous media. Enhanced oil recovery, soil remediation, and medical diffusion in biological tissues are a few examples. The problems span entirely different fields, from energy production to preservation of natural habitats to medicine. Later on, we will observe that such porous structures are present in various materials, both natural and manufactured.

All the aforementioned phenomena are complex physical processes and are remarkably challenging to model and simulate. In applied mathematics, engineering, and physics, one aims to simplify the processes as much as possible to be able to investigate them while at the same time, obtain accurate simulations. This can be achieved in various ways, e.g., by neglecting some secondary effects, but also by assuming that some quantities remain constant, and more. The description will then not be completely accurate, but the problems are otherwise too complex to solve. The errors here obtained are defined as modeling errors due to the simplifications and assumptions taken into account. Furthermore, one incurs in numerical errors due to inaccuracies in the numerical computations.

Partial differential equations (PDEs) are one of the mathematical tools used to study all these processes. Even though these equations have been investigated for decades, there is still a need for accurate and robust solvers due to their complexity. We will concentrate mainly on how to treat the nonlinearities that often characterized such equations. Furthermore, when investigating systems of PDEs, the equations involved can be coupled together. We will study different solution algorithms that either solve the

equations all together or split them.

A central focus of this work is the flow of water in variably saturated porous media, modeled by the Richards equation [12, 33, 55]. Unsaturated/saturated domains are found, for example, in the section of the soil closer to the surface. In the upper part of the soil, air and water phases coexist; thus, the domain is unsaturated. Below the water table, only the water phase is found, and thus the domain becomes fully saturated. Furthermore, an external component, e.g., a surfactant, is dissolved and transported in the water phase. The process is modeled by a reactive transport equation [3, 21, 35, 43, 47, 49, 65, 70, 80]. Soil remediation is a classical physical phenomenon that can be studied thanks to this set of equations. The surfactant is dissolved in the water phase flowing in the variably saturated porous medium. Of large interest is, for example, to understand which regions have already been reached by the substance. Furthermore, it allows us to know if the reservoirs below the water tables have been contaminated. Surfactants are chemical compounds commonly used in our everyday life, from detergency to emulsification. Moreover, they are also used in enhanced oil recovery (EOR) processes. They are frequently used to remove contaminants from underground reservoirs. They are considered safe at low concentrations [20], but their accumulation in water reservoirs can results in natural disasters [86]. Thus one needs to pay particular attention to the transportation of such substances.

The equations used here to describe the flow and transport are fully coupled, nonlinear, possibly degenerate PDEs. The resulting system is extraordinarily complex to solve numerically. When investigating flow and transport, the equations are commonly one-way coupled. The flow influences the transport, but the external component's effects on the flow are neglected for ease of the study. In this work, we will include such effects, and thus the equation will be fully coupled.

To solve numerically the system of PDEs obtained, one needs first to discretize in time and space the equations involved. We refer to [28] for a practical review of numerical methods for the Richards equation. Due to the low regularity of the solution and the need of relatively large time steps, the backward Euler method is an optimal candidate for the time discretization. There exist numerous spatial discretization techniques, to name a few the Galerkin Finite Element Method (*FEM*) [9, 46, 61, 73], the Discontinuous Galerkin Method (*DGM*) [7, 52], the Mixed Finite Element Method (*MFEM*) [6, 13, 30, 66, 67, 83, 87], the Finite Volume Method (*FVM*) [16, 25, 26] and the Multi-Point Flux Approximation (*MPFA*) [1, 8, 10, 45], which is an example of *FVM*.

Since the time discretization, given by the backward Euler, is implicit, the resulting fully discretized problem is a sequence of nonlinear equations. Nonlinear quantities characterize, in fact, both Richards and transport equations. Thus, one needs to linearize the resulting discretized nonlinear problems. There exist numerous linearization techniques, each with its advantages and disadvantages. For example, one can use the classical Newton method, here presented in Section 3.2.1, which is remarkably fast but not very robust. The scheme can fail to converge for especially complex problems. A possible alternative, here thoroughly investigated and first presented in Section 3.2.3, is the L-scheme [55, 64]. The scheme is usually slower in terms of the number of iterations than the Newton method, but it is more robust. Also, there exist various techniques that can be used to accelerate the scheme. In this work, we test two different approaches; first, we combined the Newton method and the L-scheme, obtaining a so-called mixed scheme. The linearization technique appears to be both globally and quadratically convergent. Moreover, we implement the Anderson acceleration (Section 3.4), a powerful post-processing tool that can drastically improve the rate of convergence of linearly convergent schemes [5].

Furthermore, either monolithic (Section 3.3.1) or splitting solvers (Sections 3.3.2 and 3.3.3) are implemented to solve the coupled problem. The former are often easier to construct and more stable. The latter give more freedom in the solving process. One can combine different solvers for the flow and the transport, respectively. We presented two different splitting schemes, the canonical nonlinear splitting, and an alternate linearized spitting. The latter appears to be a valid alternative to the more commonly used nonlinear approach.

1.1 Main results

This thesis's main contributions are novel linearization schemes and solving algorithms for coupled Richards and transport equations. Four different papers compose this thesis. In each one of them, we studied different models, but the conclusions were often similar. The original flow and transport in variably saturated porous media are extended by adding the dynamic capillarity effects and later the hysteresis effects. The L-scheme, already studied in various previous works, e.g., [55, 64], is here extended to this particular coupled problem. The Anderson acceleration is used to improve the rate of convergence of the linearly convergent scheme. Moreover, given the coupled nature of the problem, we investigate different solving algo-

rithms. We propose a linearized splitting solver as an alternative to the canonical nonlinear splitting. Finally, we also investigate a global random walk (GRW) solution algorithm.

1. **To develop solvers for fully coupled Richards and transport equations.** In Papers A and C, we design and numerically assess three different algorithms' performance: a monolithic solver and two splitting ones. The alternate linearized splitting proposed here appears to be a better alternative to the canonical nonlinear splitting. It results faster in terms of the number of iterations required to obtain the solution. Furthermore, the two solvers are equally accurate.

2. **Rigorous convergence analysis for the L-scheme applied to the coupled Richards and transport equations.** In Paper A, we prove the convergence of the L-scheme, in its monolithic formulation, and discuss the convergence of modified Picard and Newton methods.

3. **To propose an improved linearization scheme.** In Paper B, we observed that a combination of the L-scheme and the Newton method results in being a globally and quadratically convergent linearization scheme. The model studied here includes the dynamic capillarity effects.

4. **To study the Anderson acceleration for the fully coupled nonstandard problem.** In Paper C, we implement the Anderson acceleration, observing how such a tool can drastically improve linearly convergent schemes, e.g., the L-scheme. The model investigated includes both dynamic capillarity and hysteresis effects.

5. **To develop a global random walk solver.** In Paper D, we propose and investigate a GRW solver for different sets of equations. The solution algorithm is first built upon a one-dimensional Richards equation, and it is finally extended to the fully coupled flow and transport investigated in the other papers. The resulting technique is free of numerical diffusion and presents considerable improvements compared to the standard two-point flux approximation.

Chapter 2

Mathematical modeling

Conservation of mass, momentum and energy are examples of conservation laws, forming the basis for many physical models, including flow in porous media. The governing equations are then completed by including constitutive laws. Such are based on experimental observations and often take into account many simplifications. Darcy's law is a classic example of constitutive law. It describes the flow of a single fluid through a porous medium. It was stated after Henry Darcy performed numerous experiments on columns of sands. To close a model, finally, boundary conditions have to be stated.

In the following sections, we will present the fundamental elements on which models for flow in porous media are built. We will start with a brief discussion regarding what is a porous medium and its properties. Then, thanks to Darcy's law and the mass conservation law, we present the model for single-phase flow in porous media. From there, we extend the model to two-phase flows including the nonstandard effects, such as dynamic capillary pressure and hysteresis. A particular case of two-phase flow, a so called one a-half phase flow, modelled by the Richards equation is the model mainly investigated in this work. An external component is dissolved in the water phase and a transport equation is also taken into account. All in all, this chapter provides a brief introduction of the mathematical models which have been the basis of this work.

2.1 Porous media

Porous media are materials presenting pores through which fluid can flow and be stored. They can be found all around us. We can observe them in the soil, in biological tissues and also in man-made materials, e.g., cement. In this work, we mainly concentrate on underground water reservoirs, but similar models can be used for different media. Two properties are fundamental when describing a porous medium: its porosity and its permeability. The first describes the ability to store fluid and depends on the size and number of pores. The latter describes the ability to transmit a fluid and thus depends on the connection between the pores.

When studying flow in porous media one need to establish which modeling scale is most appropriate. One could investigate the individual pores and pore channels, thus studying the flow on a so called pore-scale. The domains considered in reservoir simulations extend often for kilometers and thus it is not convenient to use such scale. One needs to individuate an appropriate representative elementary volume (REV). REVs are the smallest volumes over which reservoir properties, such as porosity and permeability, are representative of the whole domain.

2.1.1 Porosity

The porosity is defined as the fraction of pore space within a REV [54, 62], precisely

$$\phi = \frac{V_p}{V_b} \tag{2.1}$$

where V_p is the volume of the pores, and V_b is the bulk volume. By its definition, ϕ is a fraction between 0 and 1. Porosity equal to 0 means that no pores are included in the domain. In this work, we consider a rigid or nondefomable domain; thus, Φ is a non-changing value that can be measured in the absence of fluid. In the case of deformable domains, the porosity depends on the pressure of the fluid and the rock compressibility.

2.1.2 Permeability

As previously stated, the permeability describes the ability to transmit fluids, [54, 62]. It is commonly indicated with K and it is a proportionality factor between the flow rate and the applied pressure. The SI-unit is m^2, but it is more often measured in darcy, $1D \approx 0.987 \cdot 10^{-12} m^2$. The permeability of reservoirs usually studied ranged from 0.1 mD (millidarcy) to 20 D [54].

For ease of presentation, one often assumes that the permeability is a scalar; more generally, it is a tensor of the form

$$\boldsymbol{K} = \begin{bmatrix} K_{xx} K_{xy} K_{xz} \\ K_{yx} K_{yy} K_{yz} \\ K_{zx} K_{zy} K_{zz} \end{bmatrix}. \tag{2.2}$$

The elements on the diagonal of $\boldsymbol{K}$ represent how the pressure drop in one direction directly influences the flow rate in the same axial direction. The remaining terms describe the relationship between flow rate in one direction and pressure drop in the perpendicular directions.

A parameter, such as the permeability, is defined as anisotropic when it changes values depending on the direction in which it is considered. Alternatively, in case of no directional differences, it is defined as isotropic [62]. Whenever the full tensor $\boldsymbol{K}$ is needed, the porous medium is defined as anisotropic. Opposite to that, if a diagonal representation is sufficient, it is called isotropic. Furthermore, the full tensor K is always symmetrical and positive definite [54].

2.1.3 Heterogeneous porous media

The soil under our feet and many other porous media contain complex structures and have heterogeneous properties. A property is defined as homogeneous if its values do not depend on the position in the domain. Conversely, it is defined as heterogeneous if it is a function of spatial location. To describe heterogeneous porous media, one often uses statistical tools [2, 38]. MRST, the Matlab Reservoir Simulation Toolbox [54], for example, includes two simplified methods to generate geostatistical relations, the *gaussianField* and the *logNormLayers* functions. Such tools were briefly used in Paper A, when we investigated a more realistic porous medium, with a heterogeneous permeability. Furthermore, in Paper D, we study domains with permeability modeled as a log-normal space random function with fixed variance and correlation lengths. An example of such domains is given by Fig. 2.1. The permeability is obtained thanks to a Kraichnan field generator [79].

2.2 Mass conservation and Darcy's law

The mass conservation law states that the temporal changes in mass within a volume must equal the fluxes flowing through its boundary and any

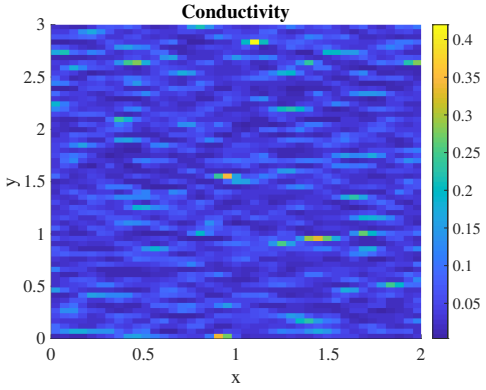


Figure 2.1: Example of heterogeneous conductivity studied in Paper D.

source and sink within the volume itself. Formally this becomes:

$$\int_{\omega} \frac{\partial(\phi\rho)}{\partial t} dV = -\int_{\partial\omega} \rho \boldsymbol{u} \cdot \boldsymbol{n} dA + \int_{\omega} \rho q dV, \tag{2.3}$$

where ω is a general volume, contained within the porous media Ω, $\partial\omega$ its boundary, ϕ the porosity of the porous medium, ρ the density of the fluid, $\boldsymbol{u}$ the flux, $\boldsymbol{n}$ the outward pointing normal vector and q any source or sink inside the domain ω. The divergence theorem can be applied to the surface integral, and considering that the volume ω is arbitrary one can conclude that the single-phase fluid must satisfy the continuity equation:

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = \rho q, \qquad \text{in } \Omega. \tag{2.4}$$

As previously stated, constitutive laws are also required to complete the model. Henry Darcy, in 1856, studied how water flows in a column of sand. The main result out of his numerous experiments is the Darcy's law for single-phase flow:

$$\boldsymbol{u} = -\frac{\boldsymbol{K}}{\mu}\big(\nabla p - \rho \boldsymbol{g}\big), \tag{2.5}$$

where $\boldsymbol{u}$ is the volumetric flux, $\boldsymbol{K}$ is the permeability introduced in the previous section, μ the dynamic viscosity, p the pressure of the fluid, ρ its density, and $\boldsymbol{g}$ the gravitational acceleration. Note that $\boldsymbol{g} = (0, 0, -9.81)^T = -9.81\nabla z$

In the case of incompressible fluid and matrix, the porosity and the density are pressure independent and constant in time. Thus, one can

simplify the continuity equation (2.4), and combining it with Darcy's law (2.5), it results in:

$$-\nabla \cdot \Big(\frac{\boldsymbol{K}}{\mu}\big(\nabla p - \rho \boldsymbol{g}\big)\Big) = q \qquad \text{in } \Omega. \tag{2.6}$$

2.3 Initial and boundary conditions

To close the system of equations introduced in the previous section, so that a solution is defined, one needs to define the boundary conditions. They determine the behavior on the boundary of the domain. Furthermore, in the case in which the fluid or the rock are compressible, one studies a parabolic equation and thus needs an initial condition. When studying incompressible fluid and rock, equation (2.6) is elliptic, and no initial condition is required.

Considering the domain Ω and the time interval $[0, T]$, with T the final time step, one is often interested in describing a closed flow system; no fluid can flow across the external boundary of the domain. This is achieved by imposing a homogeneous Neumann condition,

$$\boldsymbol{u}(x,t) \cdot \boldsymbol{n} = 0 \qquad \text{for } x \in \partial\Omega \text{ and } t \in (0, T]. \tag{2.7}$$

Using the boundary condition (2.7), also called no-flow condition, any pressure solution of (2.6) is defined up to a constant. One must prescribe a value along the boundary. The reservoir investigated is often connected, along at least one side, to a more extensive porous medium which can provide additional pressure support. One fixes the pressure on such a boundary by imposing a Dirichlet condition

$$p(x,t) = p_{Diric}(x,t) \qquad \text{for } x \in \Gamma \subset \partial\Omega \text{ and } t \in (0, T]. \tag{2.8}$$

Finally,a non-vanishing normal flux could be prescribed on a part of the boundary, which is included by a so-called inhomogeneous Neumann condition

$$\boldsymbol{u}(x,t) \cdot \boldsymbol{n} = \boldsymbol{u}_{Neu}(x,t) \qquad \text{for } x \in \Gamma \subset \partial\Omega \text{ and } t \in (0, T]. \tag{2.9}$$

As previously stated, in the case of a parabolic equation, one also needs to impose an initial condition of the form

$$p(x,0) = p_0(x) \qquad \text{for } x \in \Omega. \tag{2.10}$$

2.4 Two-phase flow

We will now extend the formulation previously presented for single-phase flow to two-phase flow. When studying two-phase flow, two fluids are transmitted within the reservoir. We can always individuate a wetting phase and a nonwetting one. The wetting phase is the fluid that is more easily attracted by the medium (*wettability*) [54], while the other is the nonwetting one. A typical example can be water and oil, the first being the wetting phase, the latter being the nonwetting one. Considering that two phases are now present in the reservoir, one can define the saturation. The saturation of the phase α, S_α, is the fraction of the pore volume occupied by the phase α. We always assume that the fluids fill the entire void space in the porous medium, so that:

$$\sum_\alpha S_\alpha = 1. \tag{2.11}$$

Given the pressures of the two phases, we define the macroscopic capillary pressure as the difference between the nonwetting (p_n) and the wetting (p_w) pressures:

$$p_{cap} = p_n - p_w. \tag{2.12}$$

The capillary pressure plays a fundamental role in the modeling of two-phase flows. Thanks to numerous experiments [19, 71, 72], it has been observed that the capillary pressure depends on the saturation of the wetting phase, i.e. $p_{cap} = P_c(S_w)$. Such holds true for steady states flow under static condition. A typical example of the capillary pressure curve is presented in Fig. 2.2. There, we can observe two curves, the drainage and the imbibition curves. The former describes the process of a nonwetting fluid displacing the wetting one. The latter describes the opposite; the wetting phase displaces the nonwetting one.

When discussing single-phase flows, we defined the permeability. In the case of two-phase flows, one needs to introduce the effective permeability K^e_α, which describes the capacity of the rock to transmit the phase α. K^e_α is always smaller than K; the second fluid β can be seen as an additional obstacle. Thus, one can define the relative permeability as

$$k_{r\alpha} = K^e_\alpha / K. \tag{2.13}$$

Relative permeabilities are usually functions of the saturations, in case of two-phase flow we have

$$k_{rn} = k_{rn}(S_n) \qquad \text{and} \qquad k_{rw} = k_{rw}(S_w), \tag{2.14}$$

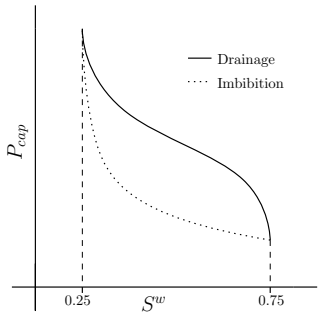


Figure 2.2: The hysteresis loop, figure inspired by the study done in [57].

where the index n indicates the nonwetting component and w the wetting one.

Finally, given the relative permeabilities $k_{r\alpha}$ we can write the conservation equations and generalized Darcy's law for two-phase flows

$$\begin{aligned} \frac{\partial}{\partial t}\big(\phi\rho_\alpha S_\alpha\big) + \nabla\cdot(\rho_\alpha \boldsymbol{u_\alpha}) &= \rho_\alpha q_\alpha, \\ \mathbf{u}_\alpha = -\frac{Kk_{r\alpha}}{\mu_\alpha}\big(\nabla p_\alpha - \rho_\alpha \boldsymbol{g}\big), \end{aligned} \tag{2.15}$$

for $\alpha = w, n$ and every domain Ω.

2.4.1 Richards equation

The Richards equation is commonly used to study the water flow in variably saturated porous media, [12, 33, 55]. The vadose or unsaturated zone extends from the ground surface to the underground water table. In this region, water and air, classical examples of two immiscible phases, coexist. Many simplifications are often taken into account, reducing the two-phase flow to a so called one-a half phase [54]. One central assumption for this is that the inactive phase, the air phase, is connected to the atmosphere. Thus the air pressure is set equal to the atmospheric one, due to the fact that air is much less viscous than water. Furthermore, the atmospheric pressure is set equal to zero, obtaining $p_n = p_{air} = 0$ and $p_{cap} = p_n - p_w = -p_w$. Based on these assumptions, the equations commonly used to model two-phase flows can be reduced to the simpler Richards equation:

$$\frac{\partial\theta(\Psi)}{\partial t} - \nabla\cdot\Big(K(\theta(\Psi))\nabla(\Psi + z)\Big) \;=\; H, \tag{2.16}$$

where θ is the water content (ΦS_w), expressed as a function of the pressure head Ψ, K is the conductivity, z denotes the vertical coordinate of $\mathbf{x}$, opposite to the gravity, and H is an external force or source term involved. The pressure head Ψ is defined as: $\Psi = \frac{p}{\rho \mathbf{g}}$, with p the pressure of the water phase. The conductivity K is a quantity that depends from both the medium and the fluid. Given the permeability $\widetilde{K}$, the conductivity is $K := \rho g \widetilde{K}/\mu$. In the following, if not otherwise specified, K will always refer to the conductivity, not the permeability. The Richards equation is a degenerate parabolic PDE, whose solution has typically low regularity [4].

One commonly used set of expressions for θ and K is given by the van Genuchten-Mualem model [82], precisely

$$\theta(\Psi) = \begin{cases} \theta_r + (\theta_s - \theta_r)\left(1 + (-\beta\Psi)^m\right)^{-\frac{m-1}{m}}, & \Psi \leq 0 \\ \theta_s, & \Psi > 0, \end{cases} \tag{2.17}$$

$$K(\theta(\Psi)) = \begin{cases} K_s \theta_e(\Psi)^{\frac{1}{2}} \left[1 - \left(1 - \theta_e(\Psi)^{\frac{m}{m-1}}\right)^{\frac{m-1}{m}}\right]^2, & \Psi \leq 0 \\ K_s, & \Psi > 0, \end{cases} \tag{2.18}$$

where θ_r and θ_s represent the values of the residual and saturated water content, $\theta_e = (\theta - \theta_r)/(\theta_s - \theta_r)$ is the effective water content, K_s is the saturated conductivity and β and m are model parameters dependent on the soil [33].

2.5 Dynamic capillarity and hysteresis

Previously, we have introduced the concepts of drainage and imbibition. These processes can repeat themselves, one after the other. For instance, one can start by injecting water (waterflooding) in a reservoir filled with oil, until a certain oil saturation is reached (residual saturation). This value represents the quantity of oil that remains trapped in the pores and cannot be extracted by merely injecting more water. The process will create an imbibition curve, like the one presented in Fig. 2.2. If then water is displaced by injecting oil, the drainage curve and a residual saturation for the water are obtained. It is evident that the two curves are completely different, thus the cycle is defined hysteretical [11, 36, 57, 74].

Moreover, a new imbibition process, results in a curve different from the primary one. The reason for this is a so-called dynamic capillary pressure. The behavior of the hysteretical system depends not only on its current state but also on its history. Such dynamic effects have been observed in

laboratory experiments and have already been studied in many papers, e.g., [22, 29, 31, 34, 58, 75, 90]. The complex structure of the pores forming the medium, and thus the droplets of oil trapped in it, are part of the causes for this phenomenon. Any point of the area bounded by the primary drainage and imbibition curves of Fig. 2.2 could represent an equilibrium point. In Paper A, both hysteresis and dynamic capillary pressure effects were neglected. This formulation is the easiest and thus the most commonly used. In Articles B and C, we use the play-type hysteresis and dynamic capillary effects as introduced in [11]. More precisely,

$$p_n - p_w \in P_c(S_w) - \tau(S_w)\partial_t S_w - \gamma \text{sign}(\partial_t S_w) \tag{2.19}$$

where $\tau(S_w) > 0$ denotes the dynamic capillary coefficient and $\text{sign}(\partial_t S_w)$ is the sign graph

$$\text{sign}(\partial_t S_w) = \begin{cases} 1 & \text{for } \partial_t S_w > 0, \\ [-1,1] & \text{for } \partial_t S_w = 0, \\ -1 & \text{for } \partial_t S_w < 0, \end{cases} \tag{2.20}$$

used to describe the hysteresis effects, with γ the width of the primary hysteresis cycle. By choosing $\tau(S_w) = 0$, the dynamic effect are neglected. Similarly, if $\gamma = 0$, there is no difference between the drainage and imbibition curves and thus the hysteresis effects are not taken into account. The resulting model is investigated in Paper B, only the dynamic capillary effects are included.

2.6 Surfactant transport

The final model studied in this work is the reactive transport equation. It describes the movement of an external component, dissolved in one of the phases. We will consider, e.g., a surfactant transported by the water phase in a variably saturated domain. Surfactants are substances that can reduce the forces occurring between two phases inside a porous medium. For instance, surfactants can be used to reduce the interfacial tension between oil and water, enabling them to combine. Thus, they are commonly used in enhanced oil recovery because they can improve the production, allowing the extraction of trapped oil [53]. Furthermore, surfactants are commonly used for remediation of contaminated soil [60]. Water is injected in contaminated reservoirs (soil washing or flushing), and the pollutant is thus removed. The water solubility is the controlling removing mechanism; surfactants and other additives can accelerate the process. Moreover, it has

been observed that pretreatment of contaminated soil with surfactant can enhance these contaminants' biodegradation.

All the aforementioned processes, involving the injection, and thus the transportation of a surfactant in porous media, can be described by the following equation.

$$\frac{\partial\big(\theta(\Psi)c\big)}{\partial t} - \nabla\cdot(D\nabla c - \mathbf{u_w}c) + R(c) \;=\; H_2, \tag{2.21}$$

where c is the concentration of the external component, D is the diffusion / dispersion coefficient, assumed to be constant for ease of presentation, $\mathbf{u_w}$ is the water flux, $R(c)$ is the reaction term, expressed as a function of the concentration, and finally H_2 is an external force or source term involved in the process. The effects due to the diffusion and the dispersion of the component are also taken into account. When studying flow in porous media, and particularly oil recovery, they are often neglected because the advection effects dominate the transport.

2.7 Fully coupled systems for reactive transport in variably saturated porous media

Combining the Richards (2.16) and the transport (2.21) equations defined in the previous sections, one obtains the coupled system:

$$\begin{aligned} \frac{\partial\theta(\Psi)}{\partial t} - \nabla\cdot(K(\theta(\Psi))\nabla(\Psi+z)) \;&=\; H_1,\\ \frac{\partial\big(\theta(\Psi)c\big)}{\partial t} - \nabla\cdot(D\nabla c - \mathbf{u_w}c) + R(c) \;&=\; H_2. \end{aligned} \tag{2.22}$$

Initial conditions and homogeneous Dirichlet boundary conditions for both pressure and concentration close the system. The formulation is restricted to the homogeneous boundary conditions for sake of simplicity. The system (2.22) is not fully coupled, but only one-way coupled. The concentration of the external component does not influence the water flow. System (2.22) represents a simplified formulation which is often valid.

A more precise formulation is obtained by including the effects of the external component on the water flow. The external component, dissolved in the water phase, influences the surface tension ζ, which becomes a function of the concentration, and this results in a rescale of the pressure [78]

$$\theta(\Psi, c) := \theta\Big(\zeta(c)\Psi\Big), \qquad \text{with} \qquad \zeta(c) = \frac{1}{1 - b\log(c/a+1)}. \tag{2.23}$$

The water content θ becomes itself a function of both pressure and concentration. The parameters a and b depend on the fluid and the medium. We refer to [37, 77, 78] for details regarding (2.23). The equation above is used to model the influence of a surfactant on the water tension but similar equations can be used for different substances. One should not neglect such effects when studying the transport of surfactants, which are injected in the reservoir for their abilities to modify the surface tension. Given the double dependency of θ on both Ψ and c, from (2.23) one naturally obtains the fully coupled system:

$$\begin{aligned} \frac{\partial \theta(\Psi,c)}{\partial t} - \nabla\cdot\Big(K\big(\theta(\Psi,c)\big)\nabla(\Psi+z)\Big) &= H_1, \\ \frac{\partial\big(\theta(\Psi,c)c\big)}{\partial t} - \nabla\cdot(D\nabla c - \mathbf{u_w}c) + R(c) &= H_2. \end{aligned} \tag{2.24}$$

The van Genuchten-Mualem formulation for the water content and the conductivity becomes:

$$\theta(\Psi,c) = \begin{cases} \theta_r + (\theta_s-\theta_r)\Big(1+\big(-\beta(\frac{1}{1-b\log(c/a+1)})\Psi\big)^m\Big)^{\frac{1-m}{m}} & \Psi\le 0, \\ \theta_s & \Psi > 0, \end{cases}$$

$$K(\theta(\Psi,c)) = \begin{cases} K_s\theta_e(\Psi,c)^{\frac{1}{2}}\left[1-\big(1-\theta_e(\Psi,c)^{\frac{m}{m-1}}\big)^{\frac{m-1}{m}}\right]^2 & \Psi\le 0, \\ K_s & \Psi > 0. \end{cases}$$

Once more homogeneous boundary conditions and initial conditions for Ψ and c close the system. We will refer to system (2.24) as the *standard fully coupled model*. This system is studied in Paper A and Paper D. In [67], a similar fully coupled problem is studied. The reaction term, in the transport equation, produces water and thus the problem results being fully coupled.

A more complete formulation is obtained when dynamic capillary pressure and hysteresis effects are taken into account. The water content is not expressed as a function of the unknown pressure and concentration, but it becomes itself an unknown. Including the nonstandard effects into the coupled system (2.24), we obtain:

$$\begin{aligned} \partial_t\theta - \nabla\cdot(K(\theta)\nabla(\Psi+z)) &= H_1, \\ \Psi + p_{cap}(\theta,c) - \tau(\theta)\partial_t\theta &\in \gamma(\theta)\mathrm{sign}(\partial_t\theta), \\ \partial_t(\theta c) - \nabla\cdot(D\nabla c - \mathbf{u_w}c) + R(c) &= H_2, \end{aligned} \tag{2.25}$$

with θ, Ψ and c the three unknowns. The sign graph can be regularized by using

$$\Xi(\xi) = \begin{cases} \text{sign}(\xi) & \text{if} \quad |\xi| \geq \delta, \\ \xi/\delta & \text{if} \quad |\xi| < \delta, \end{cases} \tag{2.26}$$

setting $\xi = \partial_t \theta$ and choosing a positive δ, the system representing the *nonstandard formulation* becomes

$$\begin{aligned} \partial_t \theta - \nabla \cdot (K(\theta)\nabla(\Psi + z)) &= H_1, \\ \Psi + p_{cap}(\theta, c) - \tau(\theta)\partial_t \theta &= \gamma \Xi(\partial_t \theta), \\ \partial_t(\theta c) - \nabla \cdot (D\nabla c - \mathbf{u_w} c) + R(c) &= H_2. \end{aligned} \tag{2.27}$$

We investigated the system (2.27) in Paper C. In Paper B we have studied the special case $\gamma = 0$, i.e. a system characterized by dynamic capillary pressure but no hysteresis.

Chapter 3

Numerical methods

In this chapter, we present different techniques that can be used to numerically solve the equations described in the previous chapter. First, we introduce time and space discretizations. The discretized equations cannot directly be solved due to the nonlinearities of the terms involved. We therefore investigate different linearization schemes. Finally, we present both monolithic and splitting schemes to solve the obtained fully coupled, discretized and linearized systems. For ease of presentation, we consider here the standard model (2.24) where dynamic capillary pressure and hysteresis effects are neglected. The discretization techniques, the linearization schemes and the solving algorithms can easily be extended to the nonstandard model (2.25), as in Papers B and C.

3.1 Discretization in space and time

Before introducing the discretization, we discuss the weak formulation of system (2.24), serving as foundation for the following discussion. We start by presenting some spaces commonly used in functional analysis. We denote by $L^2(\Omega)$ the space of real valued, squared integrable functions defined on Ω, and $H^1(\Omega)$ its subspace, containing the functions having also the first order weak derivatives in $L^2(\Omega)$. $H_0^1(\Omega)$ is the space of functions belonging to $H^1(\Omega)$ and having a vanishing trace on $\partial\Omega$. Furthermore, we denote by $< \cdot, \cdot >$ the standard $L^2(\Omega)$ scalar product (and by $\|\cdot\|$ the corresponding norm) and the pairing between $H_0^1(\Omega)$ and its dual $H^{-1}(\Omega)$. Finally, by $L^2(0,T;X)$, for some $T > 0$, we mean the Bochner space of functions taking values in the Banach-space X, the extension to $H^1(0,T;X)$ being

straightforward. For a more detailed introduction we refer to the textbook [23].

We state the weak formulation of the problem related to (2.24), as presented in Paper A:

Problem P: Find $\Psi, c \in L^2(0,T;H_0^1(\Omega)) \cap H^1(0,T;H^{-1}(\Omega))$ such that

$$< \partial_t \theta(\Psi,c), v > + < K\big(\theta(\Psi,c)\big)\nabla(\Psi+z), \nabla v > = < H_1, v >, \qquad (3.1)$$

and

$$< \partial_t \big(\theta(\Psi,c)c\big), w > + < D\nabla c + \mathbf{u_w} c, \nabla w > = < H_2, w >, \qquad (3.2)$$

hold for all $v, w \in H_0^1(\Omega)$ and for almost every $t \in (0,T]$. For ease of presentation, we neglected the reaction term $R(c)$.

Due to the low regularity of the solution [4] and the need for relatively large time steps, the backward Euler method is the best candidate for the time discretization. The process studied in reservoir simulations can take place over considerable time intervals, thus the need for large time steps. We discretize Problem P by combining the backward Euler method with linear Galerkin finite elements as spatial discretization.

Let $N \in \mathbb{N}$ be a strictly positive natural number and the time step $\Delta t := T/N$. Thus, the discrete times are $t_n := n\Delta t$ $(n \in \{0,1,\ldots,N\})$. Furthermore, let $\mathcal{T}_h$ be a regular decomposition of Ω into d-dimensional simplices, $\overline{\Omega} = \underset{\mathcal{T}\in\mathcal{T}_h}{\cup} \mathcal{T}$, with h denoting the mesh diameter. The finite element space $V_h \subset H_0^1(\Omega)$ is defined by

$$V_h := \{v_h \in H_0^1(\Omega) \; s.t. \; v_{h|\mathcal{T}} \in \mathbb{P}_1(\mathcal{T}), \text{ for any } \mathcal{T} \in \mathcal{T}_h\}, \qquad (3.3)$$

where $\mathbb{P}_1(\mathcal{T})$ denotes the space of linear polynomials on $\mathcal{T}$ and $v_{h|\mathcal{T}}$ the restriction of v_h onto $\mathcal{T}$.

For the fully discrete counterpart of Problem P we let $n \geq 1$ be fixed and assume that $\Psi_h^{n-1}, c_h^{n-1} \in V_h$ are given. The solution pair at time t_n solves:

Problem $\mathbf{P}_n$: Find $\Psi_h^n, c_h^n \in V_h$ such that for all $v_h, w_h \in V_h$ there holds

$$\begin{aligned} &< \theta(\Psi_h^n, c_h^n) - \theta\big(\Psi_h^{n-1}, c_h^{n-1}\big), v_h > \\ &+\Delta t < K(\theta(\Psi_h^n, c_h^n))(\nabla \Psi_h^n + \mathbf{e_z}), \nabla v_h > = \Delta t < H_1, v_h >, \end{aligned} \qquad (3.4)$$

and

$$\begin{aligned} &< \theta(\Psi_h^n, c_h^n)c_h^n - \theta(\Psi_h^{n-1}, c_h^{n-1})c_h^{n-1}, w_h > \\ &+\Delta t < D\nabla c_h^n + \mathbf{u_w^n} c_h^n, \nabla w_h > = \Delta t < H_2, w_h >, \end{aligned} \qquad (3.5)$$

where $\mathbf{e_z}$ denotes the unit vector in the direction opposite to gravity. Note that Problem P_n is a coupled system of two elliptic, nonlinear equations. The water content θ and the conductivity K are both expressed as functions of the unknown pressure Ψ_h^n and concentration c_h^n.

The space discretization presented above is based on linear Galerkin finite elements. Finite element methods are the most common discretization techniques used for these sets of equations. The linearization schemes and solving algorithms, already investigated in previous papers, e.g., [55, 64], are built on them. For this reason, Problem P_n will serve as a basis for the presentation of the linearization schemes and solving algorithms in the following sections.

Finite elements methods are not the only discretization techniques available. There exist also a large number of finite-difference and finite-volume methods. The standard discretization used in MRST and also throughout the industry is the two-point flux approximation scheme (TPFA). TPFA is a finite-volume method which is easy and computationally cheap to implement. We refer to [54] for both a presentation of the method and description of the implementation in MRST. Different discretization schemes can be taken into consideration; the linearization schemes do not depend on them.

3.2 Linearization schemes

The models presented in the previous chapter contain several nonlinearities. In particular, the water content θ and the conductivity K, defined by the van Genuchten formulation, are highly nonlinear. Furthermore, the reaction term, often included in the transport equation, can be nonlinear.

There exist numerous techniques to linearize nonlinear quantities. In this work, we investigate mainly the classical Newton method, the L-scheme, presented as an alternative to the former, and the modified Picard method. All of them are characterized by the introduction of a linearization loop, with iteration index j, starting at $j = 1$. At the time step n and iteration $j + 1$ we will indicate with $\Psi_h^{n,j+1}$, the unknown pressure, similarly we can define the other unknowns. One needs to specify an initial guess for $j = 1$. The most natural choice is setting $\Psi_h^{n,1} = \Psi_h^{n-1}$. For globally converging schemes, such as the L-scheme, one could use a different value for the pressure, but such choice is necessary when using the Newton method. This may induce severe restrictions on the time step size as observed in [69].

We decided to investigate three linearization schemes because each has

its advantages and disadvantages; a perfect scheme has not been found yet. The Newton method is quadratically but only locally convergent. The L-scheme is generally globally but only linearly convergent. Thus, the former, when converging, is faster in term of numbers of iterations. The latter is more robust and often converges when the Newton method fails. The modified Picard method is only linearly and locally convergent but it is more robust than the Newton method.

In the following, we introduce the three different linearization schemes of interest, by defining a single iteration. To illustrate these, we exemplary utilize Richards equation, part of the coupled Problem P_n.

3.2.1 Newton method

The Newton method is the most commonly used linearization scheme, due to its potential quadratic convergence. However, it is only locally convergent. Thus, sufficiently small time steps are required to obtain the numerical solution. It is conveniently implemented in numerous programs, e.g., it is the standard solver in MRST. The scheme requires the computation of the Jacobian matrix, which can be challenging and time-consuming. In MRST, automatic differentiation is implemented. In the following chapter, we will present, through snapshots of the code, how to use the Newton solver already implemented in MRST.

Taking into consideration the Richards equation, its linearization thanks to the Newton method for finding $\Psi_h^{n,j+1}$, given $\Psi_h^{n,j}$ and Ψ_h^{n-1} is:

$$\begin{aligned}
&< \theta(\Psi_h^{n,j}) - \theta(\Psi_h^{n-1}), v_h > + < \theta'(\Psi_h^{n,j})(\Psi_h^{n,j+1} - \Psi_h^{n,j}), v_h > \\
&+ \Delta t < K\big(\theta(\Psi_h^{n,j})\big)(\nabla \Psi_h^{n,j+1} + \mathbf{e_z}), \nabla v_h > \\
&+ \Delta t < K'\big(\theta(\Psi_h^{n,j})\big)\theta'(\Psi_h^{n,j})(\nabla \Psi_h^{n,j} + \mathbf{e_z})(\Psi_h^{n,j+1} - \Psi_h^{n,j}), \nabla v_h > \\
&= \Delta t < H_1, v_h > .
\end{aligned} \tag{3.6}$$

We observed in numerous simulations, that the Newton method as here defined, can fail to converge [40, 41, 39]. For this reason, there is a need for alternative linearization schemes which, even though only linearly converging, are more robust. In particular, when modeling flows in porous media, there is a need for large time steps. The phenomena investigated can take place on considerable intervals of times and thus using a too fined time step can become computationally unpractical.

It should be stressed that there are several modifications of the Newton scheme improving the aforementioned aspects as, to name a few, line-search and trust-region methods, or Anderson acceleration techniques, as

discussed in [28, 42, 48, 51, 84, 85, 89]. Nevertheless, considering their complexities, we believe that the modified Picard method and especially the L-scheme are better alternatives due to their lower computational cost.

3.2.2 Modified Picard method

The modified Picard method, introduced by Celia in 1990 [18], can be interpreted as a reduced version of the Newton method. Anyhow, the scheme appears to be more robust. Instead of computing first order Taylor approximations, particular nonlinearities are merely approximated at the previous iteration value. The Richards equation, linearized by the modified Picard method, at iteration $j+1$, reads:

$$\begin{aligned} &< \theta(\Psi_h^{n,j}) - \theta(\Psi_h^{n-1}), v_h > + < \theta'(\Psi_h^{n,j})(\Psi_h^{n,j+1} - \Psi_h^{n,j}), v_h > \\ &+ \Delta t < K\big(\theta(\Psi_h^{n,j})\big)(\nabla \Psi_h^{n,j+1} + \mathbf{e_z}), \nabla v_h > = \ \Delta t < H_1, v_h > . \end{aligned} \tag{3.7}$$

No derivatives of the conductivity term K are computed. Only the derivative of the water content θ is taken into account. The resulting scheme is linearly convergent and thus, in general, slower, in term of numbers of iterations than the Newton method, yet more robust in practice. Still, as for the Newton method, the modified Picard method can fail to converge for particularly complex problems [41]. Still the results can be improved by investigating smaller time steps. This can be time-consuming and can result in overall slow solving algorithms.

3.2.3 L-scheme

The L-scheme is an ever more robust method than the modified Picard method. It does not require the computation of any derivative and thus is extremely easy to implement. For the Richards equation, the L-scheme linearized formulation reads:

$$\begin{aligned} &< \theta(\Psi_h^{n,j}) - \theta(\Psi_h^{n-1}), v_h > + L < \Psi_h^{n,j+1} - \Psi_h^{n,j}, v_h > \\ &+ \Delta t < K\big(\theta(\Psi_h^{n,j})\big)(\nabla \Psi_h^{n,j+1} + \mathbf{e_z}), \nabla v_h > = \Delta t < H_1, v_h >, \end{aligned} \tag{3.8}$$

where L is a positive user-defined parameter, on which only mild restriction are applied [41, 55]. To ensure the converge of the scheme, the quantities involved must satisfy some conditions, e.g., the function θ must be Lipschitz continuous and monotonically increasing. The global, linear convergence of the scheme, for Richards equation, is proved in [55]. In Paper A, we extended the proof the the coupled problem (2.24).

In all the numerical examples investigated in this work, the scheme has appeared to be particularly robust and it ensured the convergence to the solution even when both the Newton and the modified Picard methods have failed to converge. Due to its linear order of convergence, the scheme can result to be slower, in terms of numbers of iterations, than the Newton method. Anyhow, because it does not require the computation of the Jacobian, it can be overall computationally faster. Furthermore, it converges also for larger time steps, compared to the Newton method, which can result in fewer total iterations on the full simulation.

Finally, in Paper B, we propose a scheme obtained by combining the Newton method and the L-scheme, in the hope to achieve a quadratically and globally convergent scheme. First, a few iterations are applied using the L-scheme, until a close enough approximation of the solution is achieved. Then, the Newton method is applied and thanks to its quadratic order of convergence, the solution is obtained within a few extra iterations. The scheme had already been investigated, for the Richards equation, in [55]. In paper B, we test it on the fully coupled problem (2.24), including the dynamic capillary effects. We have observed that such an L-scheme/Newton method can converge for larger time steps than the classical Newton method and in fewer iterations than the L-scheme. The scheme appears to be a clear improvement compared to both the Newton method and the L-scheme. The disadvantage is that the user must choose of when to switch between the methods and must compute the Jacobian.

3.3 Solvers for coupled problems

The systems of equations presented in the previous chapters are characterized not only by their nonlinearities but also by their coupled nature. Considering the fully coupled Richards and reactive transport formulation, the water content θ is a function of both unknowns, the pressure Ψ and the concentration c. Among several, there exist two approaches for solving coupled equations, either a fully implicit solver, also called *monolithic*, or a splitting/sequential approach. In this work, we investigate the monolithic solver and two different splitting algorithms. The former is usually more robust but requires the study of larger matrices and in case of fine meshes can be computationally expensive. The latter formulation gives more freedom to the reader. Solving the equation separately, one can choose different solvers for each of them. The solvers handle matrices half the size compared to the ones investigated in the monolithic formulation.

In the following, for each approach, we present how the different lin-

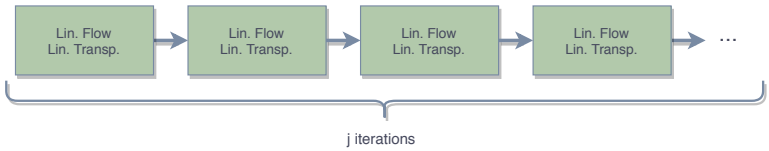


Figure 3.1: Monolithic scheme.

earization schemes are implemented.

3.3.1 Monolithic solver

We start with the fully implicit or monolithic solver. Given the standard formulation (2.24), at the time step n and iteration $j+1$, the two equations are expressed as functions of both the unknown pressure $\Psi_h^{n,j+1}$ and the unknown concentration $c_h^{n,j+1}$.

$$\begin{cases} F_1^{lin}(\Psi_h^{n,j+1}, c_h^{n,j+1}) & = 0, \\ F_2^{lin}(\Psi_h^{n,j+1}, c_h^{n,j+1}) & = 0, \end{cases} \tag{3.9}$$

with F_1 the Richards equation and F_2 the transport. Such equations are characterized by nonlinear quantities and thus, a linearization scheme is required. We indicate with F^{lin}, the linearized formulation of F obtained by the Newton method, the modified Picard method, or the L-scheme. Having two equations and two unknowns, we can solve the system at once. In Figure 3.1, we report a schematized version of this approach.

We will present now how the different linearization schemes can be implemented together with the monolithic solver.

Monolithic Newton method

The Newton method in its monolithic formulation (Mono-Newton) reads: Let $\Psi_h^{n-1}, c_h^{n-1}, \Psi_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1}, c_h^{n,j+1} \in V_h$ such that

$$
\begin{aligned}
&< \theta(\Psi_h^{n,j}, c_h^{n,j}) - \theta(\Psi_h^{n-1}, c_h^{n-1}), v_h > \\
&+ < \frac{\partial \theta}{\partial \Psi}(\Psi_h^{n,j}, c_h^{n,j})(\Psi_h^{n,j+1} - \Psi_h^{n,j}), v_h > \\
&+ < \frac{\partial \theta}{\partial c}(\Psi_h^{n,j}, c_h^{n,j})(c_h^{n,j+1} - c_h^{n,j}), v_h > \\
&+ \Delta t < K\big(\theta(\Psi_h^{n,j}, c_h^{n,j})\big)\big(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}\big), \nabla v_h > \\
&+ \Delta t < K'\big(\theta(\Psi_h^{n,j}, c_h^{n,j})\big)\frac{\partial \theta}{\partial \Psi}(\Psi_h^{n,j}, c_h^{n,j})\big(\nabla(\Psi_h^{n,j}) \\
&+ \mathbf{e_z}\big)(\Psi_h^{n,j+1} - \Psi_h^{n,j}), \nabla v_h > \\
&+ \Delta t < K'\big(\theta(\Psi_h^{n,j}, c_h^{n,j})\big)\frac{\partial \theta}{\partial c}(\Psi_h^{n,j}, c_h^{n,j})\big(\nabla(\Psi_h^{n,j}) \\
&+ \mathbf{e_z}\big)(c_h^{n,j+1} - c_h^{n,j}), \nabla v_h >= \Delta t < H_1, v_h >, \\
&\\
&< \theta(\Psi_h^{n,j}, c_h^{n,j}) c_h^{n,j+1} - \theta(\Psi_h^{n-1}, c_h^{n-1}) c_h^{n-1}, w_h > \\
&+ < \frac{\partial \theta}{\partial \Psi}(\Psi_h^{n,j}, c_h^{n,j})(\Psi_h^{n,j+1} - \Psi_h^{n,j}) c_h^{n,j}, w_h > \\
&+ < \frac{\partial \theta}{\partial c}(\Psi_h^{n,j}, c_h^{n,j})(c_h^{n,j+1} - c_h^{n,j}) c_h^{n,j}, w_h > \\
&+ \Delta t < D \nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla w_h > \\
&+ \Delta t < R(c^{n,j}), w_h > + \Delta t < R'(c^{n,j})(c_h^{n,j+1} - c_h^{n,j}), w_h > \\
&= \Delta t < H_2, w_h >,
\end{aligned}
\tag{3.10}
$$

hold true for all $v_h, w_h \in V_h$.

Remark 1 *Observe that* $\mathbf{u_w^{n-1}} := -K(\theta_h^{n-1}, c_h^{n-1})\nabla(\Psi_h^{n-1} + z)$ *appears in the advection term of the linearized transport equation. This is done for the ease of presentation. Nevertheless, the calculations carried out in this work were also performed using* $\mathbf{u_w^n} := -K(\theta_h^{n,j}, c_h^{n,j})\nabla(\Psi_h^{n,j+1} + z)$. *The differences in the results were marginal.*

Monolithic modified Picard

The monolithic modified Picard (Mono-Picard) approach for solving (2.24) reads as:

Let $\Psi_h^{n-1}, c_h^{n-1}, \Psi_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1}, c_h^{n,j+1} \in V_h$ such that

$$\begin{aligned}
&< \theta(\Psi_h^{n,j}, c_h^{n,j}) - \theta(\Psi_h^{n-1}, c_h^{n-1}), v_h > \\
&+ < \frac{\partial \theta}{\partial \Psi}(\Psi_h^{n,j}, c_h^{n,j})(\Psi_h^{n,j+1} - \Psi_h^{n,j}), v_h > \\
&+ < \frac{\partial \theta}{\partial c}(\Psi_h^{n,j}, c_h^{n,j})(c_h^{n,j+1} - c_h^{n,j}), v_h > \\
&+ \Delta t < K\big(\theta(\Psi_h^{n,j}, c_h^{n,j})\big)(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_h > \\
&= \Delta t < H_1, v_h >, \\
&< \theta(\Psi_h^{n,j}, c_h^{n,j}) c_h^{n,j+1} - \theta(\Psi_h^{n-1}, c_h^{n-1}) c_h^{n-1}, w_h > \\
&+ < \frac{\partial \theta}{\partial \Psi}(\Psi_h^{n,j}, c_h^{n,j})(\Psi_h^{n,j+1} - \Psi_h^{n,j}) c_h^{n,j}, v_h > \\
&+ < \frac{\partial \theta}{\partial c}(\Psi_h^{n,j}, c_h^{n,j})(c_h^{n,j+1} - c_h^{n,j}) c_h^{n,j}, v_h > \\
&+\Delta t < D \nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla w_h > + \Delta t < R(c^{n,j}), w_h > \\
&= \Delta t < H_2, w_h >,
\end{aligned} \tag{3.11}$$

hold true for all $v_h, w_h \in V_h$.

We note that no derivative of the nonlinear conductivity K and the reaction term R have been computed. Instead, the quantities are linearized by simply using the values of pressure and concentration obtained from the previous iteration j.

Monolithic L-scheme

The monolithic L-scheme (Mono-LS) approach for solving (2.24) reads as: Let $\Psi_h^{n-1}, c_h^{n-1}, \Psi_h^{n,j}, c_h^{n,j} \in V_h$ be given, L_1 and L_2 are free to be chosen

parameters, find $\Psi_h^{n,j+1}, c_h^{n,j+1} \in V_h$ such that

$$\begin{aligned}
&< \theta(\Psi_h^{n,j}, c_h^{n,j}) - \theta(\Psi_h^{n-1}, c_h^{n-1}), v_h > \\
&+ < L_1(\Psi_h^{n,j+1} - \Psi_h^{n,j}), v_h > \\
&+ \Delta t < K\big(\theta(\Psi_h^{n,j}, c_h^{n,j})\big)(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_h > \\
&= \Delta t < H_1, v_h >, \\
&< \theta(\Psi_h^{n,j}, c_h^{n,j}) c_h^{n,j+1} - \theta(\Psi_h^{n-1}, c_h^{n-1}) c_h^{n-1}, w_h > \\
&+ < L_2(c_h^{n,j+1} - c_h^{n,j}), v_h > \\
&+\Delta t < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla w_h > + \Delta t < R(c^{n,j}), w_h > \\
&= \Delta t < H_2, w_h >,
\end{aligned} \tag{3.12}$$

hold true for all $v_h, w_h \in V_h$. L_1 and L_2 should be large enough in order to ensure the convergence of the scheme [41].

The L-scheme does not involve the computations of any derivatives, the linear systems to be solved within each iteration are better conditioned compared to the ones given by Newton or Picard methods [40, 41, 39, 55] and it is globally (linearly) convergent.

3.3.2 The nonlinear splitting approach (NonLinS)

An iterative splitting approach involves successively solving the flow and the transport equations separately, iterating between the two. The main advantage of such an algorithm is that two matrices just half the size of the one used in the monolithic formulation have to be handled. Furthermore, solving the problem sequentially allows for combining tailored different solvers for each equation.

We investigate two splitting schemes: the nonlinear splitting (NonLinS) and the alternate linearized splitting (AltLinS), presented in Fig. 3.2 and Fig. 3.3, respectively. The former reads:
Let $c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1}$ such that

$$F_1(\Psi_h^{n,j+1}, c_h^{n,j}) = 0, \tag{3.13}$$

and then find $c_h^{n,j+1} \in V_h$ such that

$$F_2(\Psi_h^{n,j+1}, c_h^{n,j+1}) = 0. \tag{3.14}$$

Multiple iterations are required to solve the nonlinear equations. The pressure $\Psi_h^{n,j+1}$ computed is incorporated in the transport $F_2(\Psi_h^{n,j+1}, c_h^{n,j+1})$,

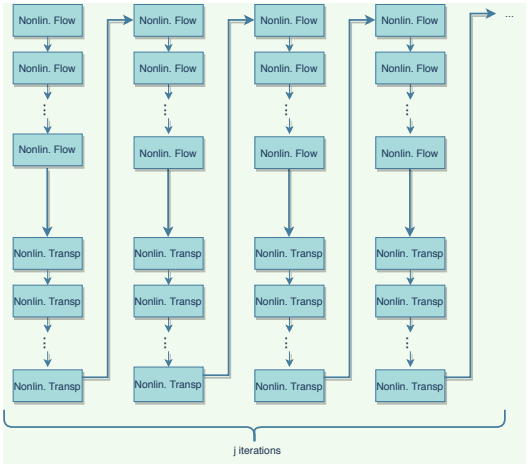


Figure 3.2: Nonlinear splitting scheme (NonLinS).

which becomes a function of only the unknown concentration $c_h^{n,j+1}$. The resulting F_1 and F_2, being nonlinear, are linearised using the Newton method, the modified Picard method or the L-scheme.

The nonlinear splitting approach for solving (2.24) reads as:
Let $\Psi_h^{n-1}, c_h^{n-1}, \Psi_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1} \in V_h$ such that

$$\begin{aligned} &< \theta(\Psi_h^{n,j+1}, c_h^{n,j}) - \theta(\Psi_h^{n-1}, c_h^{n-1}), v_h > \\ +\Delta t &< K(\Psi_h^{n,j+1}, c_h^{n,j})(\nabla \Psi_h^{n,j+1} + \mathbf{e_z}), \nabla v_h > \\ &= \Delta t < H_1, v_h > \end{aligned} \tag{3.15}$$

hold true for all $v_h \in V_h$, and next
find $c_h^{n,j+1} \in V_h$ such that

$$\begin{aligned} &< \theta(\Psi_h^{n,j+1}, c_h^{n,j+1}) c_h^{n,j+1} - \theta(\Psi_h^{n-1}, c_h^{n-1}) c_h^{n-1}, w_h > \\ &+\Delta t < D \nabla c_h^{n,j+1} + \mathbf{u_w^{n,j+1}} c_h^{n,j+1}, \nabla w_h > \\ +\Delta t &< R(c_h^{n,j+1}), w_h > \; = \; \Delta t < H_2, w_h > \end{aligned} \tag{3.16}$$

hold true for all $w_h \in V_h$. In Fig. 3.2, we illustrate the schematized version of the nonlinear splitting scheme.

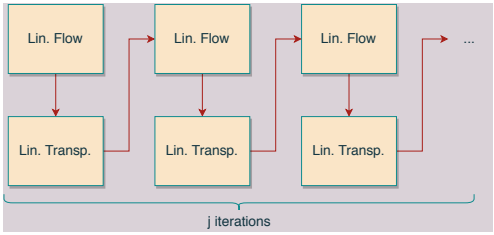


Figure 3.3: Alternate linearised splitting scheme (AltLinS).

3.3.3 The alternate linearized splitting approach (AltLinS)

Finally, the alternate linearized splitting scheme performs only one linearisation step per iteration, see Fig. 3.3. The alternate splitting scheme can be written as:
Let $c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1} \in V_h$ such that

$$F_1^{lin}(\Psi_h^{n,j+1}, c_h^{n,j}) = 0, \tag{3.17}$$

and then $c_h^{n,j+1}$ such that

$$F_2^{lin}(\Psi_h^{n,j+1}, c_h^{n,j+1}) = 0. \tag{3.18}$$

Depending on which linearisation is used, we refer to alternate splitting Newton (AltLinS-Newton), alternate splitting Picard (AltLinS-Picard) or alternate splitting L-scheme (AltLinS-LS). All of the schemes are presented in details below.

The alternate linearized Newton method (AltLinS-Newton)

The alternate Newton method applied to (2.24) reads as
Let $\Psi_h^{n-1}, c_h^{n-1}, \Psi_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1} \in V_h$ such that

$$\begin{aligned}
&< \theta(\Psi_h^{n,j}, c_h^{n,j}) - \theta(\Psi_h^{n-1}, c_h^{n-1}), v_h > \\
&+ < \frac{\partial \theta}{\partial \Psi}(\Psi_h^{n,j}, c_h^{n,j})(\Psi_h^{n,j+1} - \Psi_h^{n,j}), v_h > \\
&+ \Delta t < K\big(\theta(\Psi_h^{n,j}, c_h^{n,j})\big)(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_h > \\
&+ \Delta t < K'\big(\theta(\Psi_h^{n,j}, c_h^{n,j})\big)\frac{\partial \theta}{\partial \Psi}(\Psi_h^{n,j}, c_h^{n,j})\big(\nabla \Psi_h^{n,j} \\
&+ \mathbf{e_z}\big)(\Psi_h^{n,j+1} - \Psi_h^{n,j}), \nabla v_h >= \Delta t < H_1, v_h >
\end{aligned} \tag{3.19}$$

hold true for all $v_h \in V_h$. Next,
find $c_h^{n,j+1} \in V_h$ such that

$$\begin{aligned} &< \theta(\Psi_h^{n,j+1}, c_h^{n,j}) c_h^{n,j+1} - \theta(\Psi_h^{n-1}, c_h^{n-1}) c_h^{n-1}, w_h > \\ &+ < \frac{\partial \theta}{\partial c}(\Psi_h^{n,j+1}, c_h^{n,j})(\Psi_h^{n,j+1}, c_h^{n,j})(c_h^{n,j+1} - c_h^{n,j}), v_h > \\ &+ \Delta t < D \nabla c_h^{n,j+1} + \mathbf{u_w^{n,j+1}} c_h^{n,j+1}, \nabla w_h > + \Delta t < R(c^{n,j}), w_h > \\ &+ \Delta t < R'(c_h^{n,j})(c_h^{n,j+1} - c_h^{n,j}), w_h > = \ \Delta t < H_2, w_h > \end{aligned} \tag{3.20}$$

hold true for all $w_h \in V_h$.

The alternate linearized Picard (AltLinS-Picard)

The alternate Picard method applied to (2.24) is formulated as follows:
Let $\Psi_h^{n-1}, c_h^{n-1}, \Psi_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1} \in V_h$ such that

$$\begin{aligned} &< \theta(\Psi_h^{n,j}, c_h^{n,j}) - \theta(\Psi_h^{n-1}, c_h^{n-1}), v_h > \\ &+ < \frac{\partial \theta}{\partial \Psi}(\Psi_h^{n,j}, c_h^{n,j})(\Psi_h^{n,j+1} - \Psi_h^{n,j}), v_h > \\ &+ \Delta t < K\big(\theta(\Psi_h^{n,j}, c_h^{n,j})\big)\big(\nabla \Psi_h^{n,j+1} + \mathbf{e_z}\big), \nabla v_h > \\ &= \ \Delta t < H_1, v_h > \end{aligned} \tag{3.21}$$

hold true for all $v_h \in V_h$. Next
find $c_h^{n,j+1} \in V_h$ such that

$$\begin{aligned} &< \theta(\Psi_h^{n,j+1}, c_h^{n,j}) c_h^{n,j+1} - \theta(\Psi_h^{n-1}, c_h^{n-1}) c_h^{n-1}, w_h > \\ &+ < \frac{\partial \theta}{\partial c}(\Psi_h^{n,j+1}, c_h^{n,j})(c_h^{n,j+1} - c_h^{n,j}) c_h^{n,j}, w_h > \\ &+ \Delta t < D \nabla c_h^{n,j+1} + \mathbf{u_w^{n,j+1}} c_h^{n,j+1}, \nabla w_h > + \Delta t < R(c_h^{n,j}), w_h > \\ &= \ \Delta t < H_2, w_h > . \end{aligned} \tag{3.22}$$

hold true for all $w_h \in V_h$.

The alternate linearized L-scheme (AltLinS-L)

The alternate L-scheme for solving (3.1–3.2) reads as
Let $\Psi_h^{n-1}, c_h^{n-1}, \Psi_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1} \in V_h$ such that

$$\begin{aligned} &< \theta(\Psi_h^{n,j}, c_h^{n,j}) - \theta(\Psi_h^{n-1}, c_h^{n-1}), v_h > + L_1 < \Psi_h^{n,j+1} - \Psi_h^{n,j}, v_h > \\ &\Delta t < K\big(\theta(\Psi_h^{n,j}, c_h^{n,j})\big)\big(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}\big), \nabla v_h > = \Delta t < H_1, v_h > \end{aligned} \tag{3.23}$$

hold true for all $v_h \in V_h$. Next
find $c_h^{n,j+1} \in V_h$ such that

$$\begin{aligned} &< \theta(\Psi_h^{n,j+1}, c_h^{n,j}) c_h^{n,j+1} - \theta(\Psi_h^{n-1}, c_h^{n-1}) c_h^{n-1}, w_h > \\ &+ L_2 < c_h^{n,j+1} - c_h^{n,j}, w_h > + \Delta t < D \nabla c_h^{n,j+1} + \mathbf{u_w^{n,j+1}} c_h^{n,j+1}, \nabla w_h > \\ &+ \Delta t < R(c_h^{n,j}), w_h > = \ \Delta t < H_2, w_h > \end{aligned} \tag{3.24}$$

hold true for all $w_h \in V_h$.

Remark 2 *(Stopping criterion) For both monolithic and splitting schemes, one stops the iterations when*

$$\left\| \Psi_h^{n,j+1} - \Psi_h^{n,j} \right\| \leq \epsilon_1, \tag{3.25}$$

$$\left\| c_h^{n,j+1} - c_h^{n,j} \right\| \leq \epsilon_2, \tag{3.26}$$

where, in our numerical studies, we choose $\epsilon_1 = \epsilon_2 = 10^{-6}$.

Remark 3 *We have presented here the linearization schemes and the solving algorithms for the standard model, we have neglected the dynamic capillarity and the hysteresis effects. Similarly, one can apply the schemes to the nonstandard formulation including these effects. The resulting solving algorithms are presented in Paper C.*

3.4 Anderson acceleration

The Anderson acceleration (AA) is a powerful post-processing tool which can drastically improve linearly convergent fixed point schemes. The L-scheme is clearly a perfect candidate. In previous studies [55, 59, 64, 76, 88], it has been observed that the scheme is more robust than the Newton method and can be considered a valid alternative. The L-scheme is only linearly convergent, and the rate of convergence depends on the linearizing parameters. Tuning the parameters to obtain optimal results, in terms of numbers of iterations, and thus computational times, can be tedious and time-consuming. The AA can drastically reduce the numbers of iterations, especially in case of non-optimal L parameters. In Paper C we observed how simpler it is to tune the AA instead of finding the optimal L.

D. G. Anderson introduced the acceleration tool in 1965 [5], and since then it has been investigated in multiple works, to list a few [15, 24, 84].

We recall here the definition of AA, presented in [84], formulated for a general fixed point problem, of the form: given $g : \mathbb{R}^n \to \mathbb{R}^n$, solve the system $x = g(x)$,

Algorithm 1 Classical Fixed-Point iteration

1: Given x_0
2: **for** $k = 0, 1,...$ until convergence **do**
3: $\quad x_{k+1} = g(x_k)$
4: **end for**

In the AA one utilizes previously computed iterates and combines their contribution to obtain a new iterate. In the following, we denote with AA(m) the Anderson acceleration, where $m + 1$ previously computed iterates are taken into account. AA(0) correspond to the nonaccelerated formulation.

Algorithm 2 Anderson Acceleration(m)

1: Given x_0
2: **for** $k = 1, 2...$ until convergence **do**
3: $\quad$ Set $m_k = min\{m, k - 1\}$
4: $\quad$ Define the matrix $F_k = (f_{k-m_k-1}, \cdots, f_{k-1})$, where $f_i = g(x_i) - x_i$
5: $\quad$ Find $\alpha \in \mathbb{R}^{m_k+1}$ that solves

$$\min_{\alpha=(\alpha_0,\cdots,\alpha_{m_k})^T} \|F_k\alpha\| \text{ s.t. } \sum_{i=0}^{m_k} \alpha_i = 1.$$

6: $\quad$ Define $x_k := \sum_{i=0}^{m_k} \alpha_i g(x_{k-m_k+i-1})$
7: **end for**

The original formulation presented in [5] allows a more general step,

$$x_k := \beta_k \sum_{i=0}^{m_k} \alpha_i g(x_{k-m_k+i-1}) + (1 - \beta_k) \sum_{i=0}^{m_k} \alpha_i x_{k-m_k+i-1},$$

for a user-defined relaxation parameter $\beta_k \in (0, 1]$. We considered the simplified formulation, obtained with $\beta_k = 1$, because we achieved no improvements in the numerical results when using the extended one.

Summarizing observations from Paper C, large values for the depth m can result in an instability of the solution algorithm. When implementing the Anderson acceleration, one has to tune such parameter properly. A

small m could produce only a small reduction in the numbers of iterations, too large m could result in a non-converging algorithm, as also previously reported in [27].

The definition of the nonlinear splitting solvers allows for different ways to apply the AA. We study three different loops, the coupling one and two linearizing ones, one for each equation. We can apply the Anderson acceleration to each of them. Two different parameters, m and m_{lin}, are defined. The former is used for the AA on the coupling loop, the latter for the implementation on the linearisation ones. For simplicity, the same m_{lin} will be used for the loop regarding the flow equation and for the one regarding the transport.

3.5 Global random walk algorithm

In Paper D, a global random walk (GRW) solution algorithm [14, 50, 63, 81], for solving flow and transport in variably saturated porous media, is presented. The solution is approximated by moving large numbers of computational particles on regular lattices, according to a specific random walk rule. The GRW can use vast numbers of particles, almost as many as the molecules involved in the reaction, resulting in an intuitive representation of the process. GRW algorithms are explicit and thus often more straightforward than typical finite volume/element schemes. Finally, the resulting scheme is practically free of numerical diffusion.

We investigated the GRW for different sets of equations, from one-dimensional Richards equation to fully coupled two-dimensional flow and transport in variably saturated porous media. For illustration, we will present here the GRW solver for the one-dimensional Richards equation in the space-time domain $[0,1] \times [0,T]$, for a through investigation of global random walk solvers we refer to the aforementioned papers.

We start by implementing the staggered finite difference scheme and the backward Euler method, which approximate (2.16) at the position $z_i = i\Delta z$, $i = 1, \cdots, 1/\Delta z$ and time $t = n\Delta t$, $n = 1, \cdots, T/\Delta t$, as

$$\begin{aligned} \theta(\psi_i^n) - \theta(\psi_i^{n-1}) = \frac{\Delta t}{\Delta z^2}\{[K(\psi_{i+1/2}^n)(\psi_{i+1}^n - \psi_i^n) \\ -K(\psi_{i-1/2}^n)(\psi_i^n - \psi_{i-1}^n)] + \left(K(\psi_{i+1/2}^n) - K(\psi_{i-1/2}^n)\right)\Delta z\}, \end{aligned} \tag{3.27}$$

with Δz the mesh diameter and Δt the time step. Once more, we use the L-scheme to linearize the water content θ and the conductivity K. The

resulting equation is

$$\begin{aligned}\psi_i^{n,j+1} = &\left[1 - (r_{i+1/2}^{n,j} + r_{i-1/2}^{n,j})\right] \psi_i^{n,j} + r_{i+1/2}^{n,j}\psi_{i+1}^{n,j} \\ &+ r_{i-1/2}^{n,j}\psi_{i-1}^{n,j} + (r_{i+1/2}^{n,j} - r_{i-1/2}^{n,j})\Delta z \\ &- \left(\theta(\psi_i^{n,j}) - \theta(\psi_i^{n-1})\right)/L,\end{aligned} \tag{3.28}$$

where

$$r_{i\pm1/2}^{n,j} = K(\psi_{i\pm1/2}^{n,j})\Delta t/(L\Delta z^2). \tag{3.29}$$

The solution $\psi_i^{n,j}$ is further represented by the distribution of $\mathcal{N}$ computational particles at the sites of the one-dimensional lattice, i.e. $\psi_i^{n,j} \approx n_i^{n,j}a/\mathcal{N}$, with a being a constant equal to a unit length, and the L-scheme (3.28) becomes

$$\begin{aligned}n_i^{n,j+1} = &\left[1 - \left(r_{i+1/2}^{n,j} + r_{i-1/2}^{n,j}\right)\right] n_i^{n,j} + r_{i+1/2}^{n,j}n_{i+1}^{n,j} \\ &+ r_{i-1/2}^{n,j}n_{i-1}^{n,j} + \lfloor \mathcal{N} f^s \Delta t \rfloor,\end{aligned} \tag{3.30}$$

where the source term is defined as

$$f^s = \left(r_{i+1/2}^{n,j} - r_{i-1/2}^{n,j}\right)\Delta z - \left[\theta(n_i^{n,j}) - \theta(n_i^{n-1}\right]/L, \tag{3.31}$$

and $\lfloor \cdot \rfloor$ denotes the floor function.

Remark 4 *One can extend the GRW solver presented above to solve the fully coupled flow and transport problem investigated in this work. The scheme is presented, in its detail, in Paper D.*

Chapter 4

Implementation in MRST

In this chapter, through snapshots of the code, we will explain how to implement in MRST [54] the equations from Chapter 2, as well as the linearization schemes and solving techniques from Chapter 3. MRST, Matlab Reservoir Simulation Toolbox, is a comprehensive collection of functions used to investigate different flow problems in porous media. Two powerful utilities are the *ad-core*, which is a collection of all the functions necessary to solve systems of equations using automatic differentiation, and *ad-blackoil*. The latter is used to study black oil problems with automatic differentiation. The flow problem, coupled with reactive transport, investigated in this work, is a special case of the more complete black-oil problem.

We will illustrate how the functions and tools available in *ad-blackoil* can be used to numerically model the Richards equations and to apply the different linearization schemes for the numerical solution. Many of the functions and tool already implemented in MRST will be considered as known to the reader. For a more through guide and basic use of MRST we refer to the book [54].

First, we present how the Richards equation and its linearized formulations are defined. We then move our attention to the coupled problem studying the monolithic solvers. Finally, we briefly introduce how the given formulation can be used to obtain a splitting algorithm.

4.1 Richards equation

At the start of this thesis, the Richards equation (2.16) was not explicitly implemented in any official distribution of MRST. We present here how to define the quantities involved, and then how to solve the equation obtained. Precisely, the equation is solved employing to the *simulateScheduleAD* function.

```
[states, report] = simulateScheduleAD(state0, model, schedule);
```

The argument `state0` represents the initial pressure, the equation is defined inside *model* and finally *schedule* includes the boundary conditions and the time step. The outputs of the *simulateScheduleAD* function are the pressures at all discrete times, saved in *states*, and in *report*, among many other useful information, the residuals and the numbers of iterations required to obtain the solution are stored.

Equation (2.16), involves the two nonlinear quantities, $\theta(\Psi)$ and $K(\theta)$, the water content and the conductivity. We define them by:

```
theta = @(p) getTheta(psi, theta_res, theta_sat, alpha, n);
% theta(p) is a function of the pressure
function theta = getTheta(p,theta_R,theta_S,alpha,n)
    theta_neg = theta_res + (theta_sat - theta_res).*(1./(1 + alpha*abs(
    p).^n)).^((n-1)/n);
    theta_pos = theta_sat;
    neg = p <= 0;
    theta = theta_neg.*neg + theta_pos.*~neg;
end
```

and

```
K =@(p, theta(p))  getConductivity(p, theta, theta_res, theta_sat, n) ;
% K(p,theta(p)) is a function of both pressure and water content
function K = getConductivity(p, theta, theta_res, theta_sat, K_sat, n)
    theta_e = (theta - theta_res)/(theta_sat - theta_res)     %
    effective water content
    K_pos = K_sat*theta_e.^(1/2).*(1 - (1 - theta_e.^(n./(n-1))).^((n-1)
    ./n)).^2;
    K_neg = K_sat;
    neg = p <= 0;
    K = K_neg.*neg + K_pos.*~neg;
end
```

Both functions are based on the van Genuchten formulation (2.17)–(2.18). The conductivity K will be later treated as a multiplier of the transmissibility matrix T, already defined within MRST.

Next we move our attention to the Richards equation itself. Its implementation is based on the `TwoPhaseOilWaterModel`, already available in MRST, inside `ReservoirModel`. Here, we report a few snapshots to present how the code has been implemented. The Richards equation can be seen as a special case of a two phase flow in which, one of the phases

is air. Here, the air phase is assumed to be constant, and thus only the water phase is investigated.

```
classdef RichardsEquationFixedPointSchemes < ReservoirModel
    properties        % new defined properties
        Newton
        LScheme
    end

    methods
        function model = RichardsEquationFixedPointSchemes(G, rock,
    fluid, varargin)
            model = model@ReservoirModel(G, rock, fluid);
            model.water = true;      % only water phase
            model.oil = false;
            model.gas = false;

            model = merge_options(model,varargin{:});
        end

        function [problem, state] = getEquations(model, state0, state,
    dt, drivingForces, varargin)

            if model.Newton     % if using the classical Newton method
             [problem, state] = equationsRichardsNewton(state0, state,
    model, dt, drivingForces, varargin{:});
            end
            if model.LScheme   % if using the L-Scheme
                [problem, state] = equationsRichardsLScheme(state0,
    state, model, dt, drivingForces, varargin{:});
            end
        end
  end
```

We observe that each linearization scheme corresponds to one equation. We will now proceed in presenting both of them, starting with the Newton method.

Richards equation linearized by the Newton method

Thanks to the automatic differentiation techniques already implemented within `ad-core`, one only needs to define the Richards equation. The Jacobian and the residuals are computed with the *SimulateScheduleAD* function.
Here ,we report some snapshots from the *equationsRichardsNewton* function. The pressure is the only unknown in this equation.

```
% Properties at current timestep
[p] = model.getProps(state, 'pressure');
% Properties at previous timestep
[p0] = model.getProps(state0, 'pressure');
% Initialize independent variables.
        [p] = initVariablesADI(p);
primaryVars = {'pressure'};
```

The water content is expressed as a function of the unknown pressure, and the conductivity K is a multiplier, also function of p, for the transmissibility matrix T. In addition the water flux `vW` is defined.

```
% Compute transmissibility
theta = fluid.theta;
Kmult = fluid.Kmult(p, theta(p));   % K is a function of p and theta

% Compute pressure gradient
gdz = model.getGravityGradient();   % gravitational contribution
dp = s.Grad(p) - gdz;
upc  = (double(dp)<=0); % for upwind discretizations

T = s.T.*s.faceUpstr(upc, Kmult);
% transmissibility matrix multiplied by the conductivity K
% Finally the water flux is
vW = -T.*dp;
```

Then the nonlinear equation (previously presented as (3.6)) is here defined.

```
% Conservation of mass for water
V = model.G.cells.volumes;
richards = (V./dt).*( theta(p) - theta(p0)) + s.Div(vW);  %Richards
    equation
```

The Newton method is implemented in the framework of `ad-core`.

Richards equation linearized by the L-scheme

We now present how to implement the L-Scheme from equation (3.8) into MRST. We use the same automatic differentiation techniques used for the Newton method but the equation solved at each iteration are linearized by the L-Scheme.
The pressure p is the only unknown

```
% Properties at current timestep
[p_prev] = model.getProps(state, 'pressure');
% Properties at previous timestep
[p0] = model.getProps(state0, 'pressure');
p = p_prev;
% Initialize independent variables.
        [p] = initVariablesADI(p);
primaryVars = {'pressure'};
```

The value of pressure obtained at the previous iteration ($\Psi^{n,j}$ in the notation from Section 3.2) is here used to compute the water content and the conductivity.

```
theta = fluid.theta;
Kmult = fluid.Kmult(p_prev, theta(p_prev));    % given quantities
T = s.T.*s.faceUpstr(upc, Kmult);

gdz = model.getGravityGradient();   % gravitational contribution
dp = s.Grad(p) - gdz;

```

```
upc  = (double(dp)<=0); % for upwind discretizations
T = s.T.*s.faceUpstr(upc, Kmult);
% transmissibility matrix multiplied by the conductivity K
vW = -T.*dp;
```

The linearized Richards equation becomes

```
richards = (V./dt).*( theta(p_prev) - theta(p0) + model.L.*(p - p_prev))
    + s.Div(vW);
```

The equation above is linear, the unknown pressure $\Psi^{n,j+1}$ appears only in the multiplication with the constant L and the water flux. The solution is obtained with an artificial Newton method within the `ad-core` framework, being the equation linear, only one iteration is necessary.

Remark 5 *The modified Picard linearization can be easily obtained from the formulation employing the L-scheme. Instead of using the constant* model.L*, one defines the water content θ at the previous pressure. The AD framework then compute the necessary derivative.*

Remark 6 *The differences between the Richards equation linearized by the L-scheme and the Newton method are minimal, in their MRST implementation. We decided to construct two separate functions for the ease of presentation. Anyhow, everything can be incorporated in a more comprehensive structure.*

4.1.1 The coupled problem

In case of the coupled problem (2.24), a set of two equations is combined. We present here how to implement such a system into MRST, extending the formulation given above for the Richards equation.

We start by defining the new expression for the water content, which becomes now a function of both pressure and concentration, $\theta(\Psi, c)$.

```
  theta = @(p,c) getThetaCoupled(p, c, theta_R, theta_S, alpha, n,a,b)
  function theta = getThetaCoupled(p, c, theta_R, theta_S, alpha, n,a,b)
    theta_neg = theta_R + (theta_S - theta_R).*(1./(1 +   abs(alpha
    *(1./(1-b.*log(c./(a) +1))).^(1).*p)).^n).^((n-1)/n);
    theta_pos = theta_S;
    neg = p <= 0;
    theta = theta_neg.*neg + theta_pos.*~neg;
  end
```

The function presented above is just an example, commonly used for the study of surfactants, but different expressions can be easily defined. For the conductivity K, we use the same function, already implemented for the Richards equation, .

The model used for the coupled, Richards and reactive transport equations, is based on the *blackoil* model already implemented in MRST, typically used to study the transportation of an external component, e.g., a polymer within a oil-water model.
An external component is now included in our study.

```
classdef RichardsTransportEquationFixedPointSchemes < ReservoirModel
  properties
    % External component present (polymer)
    polymer
    Newton
    LScheme
  end
methods
  function model = RichardsTransportEquationFixedPointSchemes(G, rock,
    fluid, varargin)
    model = model@ReservoirModel(G, rock, fluid);
    model.water = true;
    model.oil = false;
    model.gas = false;
    model.polymer = true;    % any external component
    model = merge_options(model,varargin{:});
  end
```

Once more we will define two different functions, one nonlinear, for the linearization by the Newton method, and one customized for the linearized equations obtained thanks to the L-scheme.

```
function [problem, state] = getEquations(model, state0, state, dt,
   drivingForces, varargin)
  if model.Newton
    [problem, state] = equationsRichardsTransport(state0, state, ...
    model, dt, drivingForces, varargin{:});
  end
  if model.LScheme
    [problem, state] = equationsRichardsTransportLScheme(state0,...
    state, model, dt, drivingForces, varargin{:});
  end
end
```

4.1.2 Implementation of the monolithic schemes

We are now going to present how to solve the coupled problem monolithically. We need to express the equations as functions of both the unknown pressure and concentration.

The monolithic Newton method

First, both pressure and concentration are imposed as unknowns of the problem.

```
% Properties at current timestep
[p, c] = model.getProps(state, 'pressure', 'concentration')
```

```
% Properties at previous timestep
[p0, c0] = model.getProps(state0, 'pressure','concentration');
% Initialize independent variables.
        [p, c] = initVariablesADI(p, c);
primaryVars = {'pressure', 'concentration'};
```

Furthermore, the water content θ and the conductivity K are defined as functions of both unknowns and the flux is computed.

```
% Compute transmissibility
theta = fluid.theta;
Kmult = fluid.Kmult(p, theta(p,c));
upc  = (double(dp)<=0); % for upwind discretizations
T = s.T.*s.faceUpstr(upc, Kmult);

dp = s.Grad(p) +  gdz;
vW = -T.*dp;

VcW = s.faceUpstr(upc,c).*vW; % upwind of cell values to face
VcD = -model.D.*s.Grad(c);    % D is the diffusion/dispersion coef.

Vc = VcW + VcD;
```

The Richards and transport equations (3.10) are then obtained as follow

```
R = c./(1+c);       % example of reaction term as function of c
richards = (V./dt).*(theta(p,c) - theta(p0,c0)) + s.Div(vW);
transport = (V./dt).*(theta(p,c).*c - theta(p0,c0).*c0 ) + s.Div(Vc)
        + V .* R;
```

both of them are nonlinear and thanks to *ad-solver*, the Newton method is automatically applied to them.

The monolithic L-scheme

Similarly, we can present how to define the linear equations obtained thanks to the L-Scheme (3.12). We start by defining the unknowns

```
[p_prev,c_prev] = model.getProps(state, 'pressure', 'concentration');
% Properties at previous timestep
[p0, c0] = model.getProps(state0, 'pressure', 'concentration');
p = p_prev;
c = c_prev;
% Initialize independent variables.
        [p, c] = initVariablesADI(p, c);
primaryVars = {'pressure','concentration'};
```

We need the previous values of both pressure and concentration ($\Psi^{n,j}$ and $c^{n,j}$), to compute the water content, the conductivity and the reaction term. Also we define the flux.

```
% Compute transmissibility
theta = fluid.theta;
rock = model.rock;
Kmult = fluid.Kmult(p_prev, theta(p_prev,c_prev));
T = s.T.*s.faceUpstr(upc, Kmult);
```

Finally, we can present the linearized equations

```
R = c./(1+c_prev)      % linearized reaction term
richards = (V./dt).*(theta(p_prev,c_prev) - theta(p0,c0)
      + model.L_p.*(p - p_prev)) + s.Div(bWvW);
transport = ((V./dt).*(theta(p_prev,c_prev).*c - theta(p0,c0).*c0
      + model.L_c.*(c - c_prev)) + s.Div(Vc) + V .* R;
```

The equations `richards` and `transport`, defined above, are linear and the solution is obtained after only one iteration.

4.1.3 Implementation of the splitting solvers

The splitting solvers investigated in this work are based on the *blackoil-sequential* module implemented in MRST. The idea is to define the two equations separately and then solve them iteratively. Two separate models are defined and then combined together.

```
  Richardsmodel = RichardsEquationModelSequential(G, rock, fluid);
  Transportmodel = SequentialTransportEquationModel(G, rock, fluid);
  seqmodel = SequentialRichardsTransportModel(Richardsmodel,
     Transportmodel);  % splitting model
```

Inside the *seqmodel* one finds the *stepFunction*, commonly used in MRST to solve each time step. In our formulation it looks something like this: one solves first the Richards equation and then the transport.

```
function [state, report] = stepFunction(model, state, state0, dt,...
  drivingForces, linsolve, nonlinsolve,...
  iteration, varargin)
  % Solve pressure and transport sequentially
  psolver = model.RichardsNonLinearSolver;
  csolver = model.TransportNonLinearSolver;

  % first solve pressure
  [state, pressureReport] = ...
  psolver.solveTimestep(state0, dt, model.RichardsEquationModel,...
  'initialGuess', state, forceArg{:});
  pressure_ok = pressureReport.Converged
  if pressure_ok  % then solve transport
    [state, transportReport] = ...
    csolver.solveTimestep(state0, dt, model.TransportEquationModel,...
    'initialGuess', state, forceArg{:});
  end
end
```

Remark 7 *In this work, we have investigated two different splitting solvers, the nonlinear splitting and the linearized alternate splitting. The former is the one automatically implemented in MRST. The latter can be obtained by setting the* stepfunction *as linear in the solver associated with the model. This forces the solver to use only one iteration on each equation, before to move to the next one.*

```
model.stepFunctionIsLinear = true;
```

We will now present the equations for the L-scheme splitting solvers. As stated above, the only difference between the two splitting solvers is the step function, not their formulation. The linearization of the equations using the Newton method can be similarly implemented.

The splitting L-scheme

In the splitting formulation, the two equations are defined separately and are expressed as function of only one of the unknowns. We start with the Richards equation, linearized with the L-scheme (3.23)
We take in consideration both pressure and concentration from the initial time step and from the previous iteration but only the pressure `p` is the primary variable and thus the unknown.

```
% Properties at current timestep
[p_prev, c_prev] = model.getProps(state, 'pressure', 'concentration');
% Properties at previous timestep
[p0, c0] = model.getProps(state0, 'pressure', 'concentration');

p = p_prev;
c = c_prev;
% Initialize independent variables.
[p] = initVariablesADI(p, wellVars{:});
primaryVars = {'pressure'};
```

The rest of the equation is identical to the L-scheme formulation presented for the Richards equation at the beginning of the chapter.

The transport equation (3.24) is expressed as function of only the unknown concentration.

```
% Properties at current timestep
[p_prev, c_prev] = model.getProps(state, 'pressure', 'concentration');
% Properties at previous timestep
[p0, c0] = model.getProps(state0, 'pressure','concentration');

p = p_prev;
c = c_prev;
% Initialize independent variables.
[c, wellVars{:}] = initVariablesADI( c, wellVars{:});
primaryVars = { 'concentration',  wellVarNames{:}};
```

The remaining of the function can be easily obtained from the previous section where we presented the monolithic L-scheme for the Richards equation. In the splitting formulation only the transport is taken into consideration.

Remark 8 *One can easily obtain the Newton method linearized equations by combining the splitting L-scheme and the monolithic Newton formulations presented above.*

4.2 Anderson acceleration

In this section, we present how to implement the Anderson acceleration, from Section 3.4, into MRST. A modified *NonLinearSolver* function has been defined: *NonLinearSolverAnderson*. There, we added a few lines to the *solveMiniStep* function, which is used to solve the iteration inside each time step. We have observed in Paper C, that the non-invasive implementation of the AA can produce major improvements of the overall performance of the iterative solvers.

Here we report a snapshot of the code, which presents how the AA, defined in 3.4, can be implemented in MRST.

```
function [state, failure, report] = solveMinistep(solver, model, state,
    state0, dt, drivingForces)
if model.UseAnderson == 1
  m = model.m;  % use depth m for AA
  if m >1
    m_i = min(i,m);  % i is iteration number
    % Save the report of each iteration, there we find the residuals
    AndersonReports(i) = stepReport;
    % Residuals
    res_p(:,m_i) = cell2mat(stepReport.LinearSolver.VectorResiduals.
    pressure);
    res_c(:,m_i) = cell2mat(stepReport.LinearSolver.VectorResiduals.c);
    % Values at previous iteration
    p_prev(:,m_i) = stepReport.LinearSolver.state_prev.pressure;
    c_prev(:,m_i) = stepReport.LinearSolver.state_prev.c;

    % Solve minimisazion problem min_a ||a*residuals|| s.t. norm a=1
    % new updated solution is x_{k+1} = sum_k a_k * g(x_k) .  g is
    vector of previously computed solutions

    a0 = 1/m_i * ones(1,m_i); % initial guess for a

    if m_i == 1
      a_p = 1;
      a_theta = 1;
      a_c = 1;
    else
    % Pressure
      A_p = getF(res_p);  % difference between the residuals [r_1 - r_n,
     r_2 - r_n, ... , r_n-1 - r_n]
      b_p = - res_p(:,size(res_p,2));
      [Q, R] = qr(A_p,0); % use reduced QR factorizations.
      a_p = R^-1 * Q' * b_p;
      a_p(m_i) = 1 - sum(a_p);  % norm a=1

    % Concentration
      A_c = getF(res_c);
      b_c = -res_c(:,size(a_c,2));
      [Q, R] = qr(A_c,0);
      a_c = R^-1 * Q' * b_c;
      a_c(m_i) = 1 - sum(a_c);

  end

```

```
41    % Obtain updated solution using the vecor a and the
42    % previous solutions
43    pressure(:,m_i) = state.pressure;
44    c(:,m_i) = state.c;
45
46    % simplified Anderson formulation, beta=1
47    pressure_sum =  sum(a_p.*pressure,2);% +(1 - beta)*sum(a_p.*p_prev,2);
48    c_sum =  sum(a_c.*c,2);% +(1 - beta) * sum(a_c.*c_prev,2);
49
50    state.pressure = pressure_sum;  % update solutions
51    state.c = c_sum;
52
53      % update matrices of residual and solutions
54      if i >= m
55        for j=1:m_i-1
56          res_p(:,j) = res_p(:,j+1);
57          res_c(:,j) = res_c(:,j+1);
58          pressure(:,j) = pressure(:,j+1);
59          c(:,j) = c(:,j+1);
60        end
61      end
62    end
63  end
```

The formulation above is valid for the Richards and Transport equations solved monolithically. It can be easily reformulated for the splitting solver.

Remark 9 *The implementations presented in this chapter regard the standard model, we neglected the dynamic capillarity and hysteresis effects. One can easily extend the formulation to the nonstandard model by introducing the capillary pressure equation* (2.19)*. In the splitting formulation, the flow equations, the Richards and the capillary pressure equations, are always solved together and expressed as functions of both the pressure and the water content. In the monolithic formulation, three equations and three unknowns* (Ψ, θ, c) *are present.*

Chapter 5

Summary and outlook

We conclude the first part of the thesis by presenting a summary and reporting each article's main results. The chapter ends with a short section, including our final remarks and outlook for future works.

5.1 Summary of the papers

Paper A: Iterative schemes for surfactant transport in porous media

In Paper A, we consider the transport of a surfactant in variably saturated porous media. The Richards equation [12, 33] models the water flow, and it is fully coupled with the transport equation for the surfactant [3, 35, 43, 47, 49, 65]. The coupling is due to the dependence of the surface tension on the concentration of the external component [37, 47, 77]. For the analytical study, the resulting system is discretized, combining a backward Euler method with the linear Galerkin finite elements.

Both equations, Richards and the reactive transport, are characterized by nonlinear quantities. We discuss three linearization techniques: the Newton method, the modified Picard method [18] and the L-scheme [55, 64]. The former appears to be the fastest scheme in terms of the number of iterations. The scheme is quadratically but only locally convergent, and it fails to converge in test cases involving unsaturated-saturated media. The modified Picard appears to be better conditioned but also fails to converge in the case of variably saturated domains. Finally, the L-scheme is the most robust linearization scheme among the three. It is globally convergent

and does not fail in the saturated part of the domain. The scheme is only linearly convergent and can result in being slower then the Newton method, in terms of numbers of iteration. The L-scheme convergence is formally proved, and we discuss the convergence of the other schemes, extending the theory from [55].

Based on the three linearization techniques, monolithic and splitting schemes are proposed, and their convergence is numerically studied. We investigate two different splitting schemes, the canonical nonlinear splitting and an alternate linearized splitting. The latter appears to be a better alternative to the former. It presents equally accurate results requiring fewer iterations. We illustrate the performance of these solvers based on five numerical examples. We report the numbers of iterations and the condition numbers of the linear systems emerging in each iteration. We compare the results concluding that the L-scheme is more robust than the Newton method. Furthermore, the alternate linearized splitting is the optimal splitting technique based on the iteration count. All the numerical simulations are performed in MRST, a Matlab toolbox for reservoir simulations. The classical two points flux approximation (TPFA), already implemented in the program, is used for the space discretization.

Paper B: An efficient numerical scheme for fully coupled flow and reactive transport in variably saturated porous media including dynamic capillary effects

In the second paper, we extend the problem studied in Paper A by including the non-equilibrium nature of the system. Particular attention is paid to the added dynamic capillary effects [22, 29, 31, 75, 90]. Such are often neglected for ease of presentation, but they are clearly observable in the experiments. The problem investigated, involving the Richards and the reactive transport equations, is fully coupled together and characterized by highly nonlinear expressions. In Paper A, we concluded that the L-scheme is the most robust linearization scheme. Being only linearly convergent, it can result in being slower than the Newton method. Furthermore, the modified Picard method is less robust than the L-scheme and slower than the Newton method; thus, it has been neglected here. In this article, we investigate only the Newton method and the L-scheme. We also study a scheme obtained by combining the two linearizations (called mixed scheme). The combined scheme had been already proposed in [55], in case of no external component and no dynamic capillary effects. The idea is to start by using few L-scheme iterations and then switch, using a user-defined criterion, to the Newton method. This results in a scheme that appears to be globally

and quadratically convergent. The initial L-scheme iterations ensure global convergence. Once a good approximation of the solution is achieved, the Newton method has a quadratic order of convergence. Alternatively, one can improve the Newton method's convergence by using smaller time steps [17]. This is computationally expensive, considering that the time domains investigated are often massive. The mixed scheme, as the L-scheme, also converges for larger time steps. Thus, the total number of iterations over the full simulation is reduced. The mixed scheme appears to be the best linearization scheme, based on the consider numerical test.

For ease of presentation, we have investigated only the monolithic and the canonical splitting schemes. Similar results can be expected for the alternate linearized approach presented in the first paper. The numerical example here studied, based on the literature [32], is implemented in MRST.

Paper C: Efficient solvers for nonstandard flow and transport in unsaturated porous media

This paper considers nonstandard flow and transport in variably saturated porous media in extension to Papers A and B. We include both dynamic capillary pressure and hysteresis effects [11, 36, 57]. Mainly due to the latter, the problem becomes particularly complex to solve numerically.

The system of equations is once more fully coupled and highly nonlinear. The Newton method and the L-scheme are again thoroughly numerically investigated. Furthermore, also monolithic, nonlinear splitting, and alternate linearized splitting solvers are studied. We examined five different numerical examples implemented in MRST. We consider four numerical examples with increasing complexity, based on a manufactured solution, inspired by the literature [56], and finally, an example in which the boundary conditions drive the flow.

The conclusions are coherent with the ones made in Papers A and B. The L-scheme is more robust than the Newton method, and the alternate linearized splitting is faster than the nonlinear splitting in terms of the numbers of iterations.

Whenever the hysteresis effects have been taken into account, the Newton method has failed to converge. The L-scheme has ensured the convergence of the algorithms but, being only linearly convergent, it results in being slow in terms of numbers of iterations. The user-defined L parameters strongly influence the rate of convergence. Finding the optimal L can be tedious and time-consuming. Thus, we decide to improve the L-scheme's convergence rate by implementing the Anderson acceleration

(AA) [5]. The AA is a powerful post-processing tool that can drastically improve linearly convergent schemes, e.g., the L-scheme. Furthermore, the AA requires one parameter m, which is easier to optimize than the L used in the linearization. The m represents the depth of the scheme, precisely how many previously computed iterates one uses to obtain the new solution. We show how a suitable m can drastically improve an L-scheme with nonoptimal L parameters. Small m values can result in marginal improvements; too large m can instead give the solver instability. In Paper B, we combined L-scheme and Newton methods obtaining a mixed scheme. The results were promising, but one needed to implement the Newton scheme and thus compute the full Jacobian. This can be challenging and time-consuming. By accelerating the L-scheme using the AA, one avoids the computation of any derivatives. Furthermore, the AA is a cheap post-processing tool, which can be easily implemented. We conclude that the implementation of the AA is a better alternative to the mixed scheme. The results are promising and no derivatives need to be computed.

Paper D: Random walk methods for flow and transport in unsaturated/saturated natural porous media

In Paper D, we investigate a new global random walk (GRW) solver. We test the algorithm on different sets of equations. We study the Richards equation in one and two dimensions, as well as the Richards equation fully coupled with the transport. Once more, the equations are characterized by highly nonlinear quantities, and thus the L-scheme is implemented. We study numerical examples based on manufactured solutions and ones describing physical experiments. Furthermore, we study domains with highly heterogeneous conductivity. For ease of the study, one often considers homogeneous domains but, e.g., the soil and many others, are often highly heterogeneous porous media. We computed the conductivity by using the Kraichnan algorithm presented in [79].

The global random walk schemes here investigated are validated by the finite volume solvers previously implemented in MRST. The advantage of using GRW schemes is that the solvers appear being practically free of numerical diffusion. The finite volume methods used in the previous article can be affected by such diffusion, which is challenging to isolate among the other errors occurring when studying coupled flow and transport.

5.2 Conclusions and outlook

In this thesis, we focused on developing efficient numerical solvers for nonlinear coupled problems in porous media. We investigated different linearization schemes but also different solving algorithms. In the first paper, we studied the Newton method, the modified Picard method, and the L-scheme. We have concluded that the L-scheme appears to be the optimal linearization scheme for our particular set of equations. The Newton method is only locally convergent and, due to the Richards equation's degeneracy, it often failed to converge. Even though only linearly, the L-scheme is globally convergent and results in being the optimal linearization scheme for flow in porous media. We also compared the monolithic (or full implicit) solver and two splitting approaches. The alternate linearized splitting proposed here appears to be a better alternative than the canonical nonlinear splitting. It gives equally accurate results, requiring few iterations.

In the second paper, considering the conclusions from the first one, we decided to combine the L-scheme and the Newton method, obtaining a so-called mixed scheme. It results in being globally and quadratically convergent and thus a clear improvement compared to the other two schemes. The few initial L-scheme iterations ensure the global convergence, while the subsequent Newton ones give the quadratic order of convergence. The mixed scheme requires fewer iterations than the L-scheme and converges for larger time steps than the Newton method.

The main disadvantage of the mixed scheme is the need for the computation of the Jacobian. This can be time-consuming and complicated. Thus, in the third paper, we decided to improve the L-scheme using the Anderson acceleration (AA). The nonstandard model investigated resulted particularly problematic to solve numerically, and the Newton method often failed to converge. The L-scheme rate of convergence strongly depends on the L parameters used in the linearization. Tuning such parameters can result in a tedious job. The AA can drastically improve the scheme without requiring an optimal value for L. One can either aim for the optimal linearization parameter L or depth m for the AA. We observed that finding the optimal m is easier than to tune the L.

Finally, in the fourth paper, we decided to use the code developed in the previous works as a benchmark for a new solving algorithm. A global random walk (GRW) solver is defined and investigated here. We started by studying a one dimensional Richards equation, eventually extending the formulation to the fully coupled standard flow and transport problem

investigated in this work. The results were particularly impressive. The GRW can use vast numbers of computational particles, almost as many as the molecules involved in the reaction, resulting in an intuitive representation of the process. The GRW algorithms are explicit and thus often more straightforward than the typical finite volume/element schemes. Finally, the solvers result in being practically free of numerical diffusion.

In this work, we studied only a particular case of two-phase flow, the one a-half phase flow modeled by the Richards equation. As we have explained in the previous chapters, the equation is the result of numerous simplifications. One could easily extend the work done to the more general two-phase flow problems, coupled with a reactive transport equation. Considering previous works, already investigating the L-scheme for two-phase flows, e.g., [44, 68], we expect that the results obtained here would be valid for the more general problem. Furthermore, there exist numerous techniques used to improve the Newton method. In this work, we neglected such, but one could compare the results obtained here with them. Due to L-scheme and AA's simplicity, we are confident that they will still result in being one of the best solving algorithms. Similarly, we investigated two splitting solvers. The study can be easily extended to a different set of coupled equations.

Considering the current general need for robust and fast solvers for systems of PDEs; we believe that the results obtained in this work can be an important stepping stone for further works.

Bibliography

[1] I. Aavatsmark. "An introduction to multipoint flux approximations for quadrilateral grids". In: *Computational Geosciences* 6.3-4 (2002), pp. 405–432.

[2] P. M. Adler, C. G. Jacquin, and J. A. Quiblier. "Flow in simulated porous media". In: *International Journal of Multiphase Flow* 16.4 (1990), pp. 691–712.

[3] A. Agosti, L. Formaggia, and A. Scotti. "Analysis of a model for precipitation and dissolution coupled with a Darcy flux". In: *Journal of Mathematical Analysis and Applications* 431.2 (2015), pp. 752–781.

[4] H. W. Alt and S. Luckhaus. "Quasilinear elliptic-parabolic differential equations". In: *Mathematische Zeitschrift* 183.3 (1983), pp. 311–341.

[5] D. G. Anderson. "Iterative procedures for nonlinear integral equations". In: *Journal of the Association for Computing Machinery* 12.4 (1965), pp. 547–560.

[6] T. Arbogast and M. F. Wheeler. "A nonlinear mixed finite element method for a degenerate parabolic equation arising in flow in porous media". In: *SIAM journal on numerical analysis* 33.4 (1996), pp. 1669–1687.

[7] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. "Unified analysis of discontinuous Galerkin methods for elliptic problems". In: *SIAM journal on numerical analysis* 39.5 (2002), pp. 1749–1779.

[8] A. Arrarás, F. J Gaspar, L. Portero, and C. Rodrigo. "Multigrid solvers for multipoint flux approximations of the Darcy problem on rough quadrilateral grids". In: *Computational Geosciences* (2020), pp. 1–16.

[9] J. W. Barrett and P. Knabner. "Finite element approximation of the transport of reactive solutes in porous media. Part 1: error estimates for nonequilibrium adsorption processes". In: *SIAM journal on numerical analysis* 34.1 (1997), pp. 201–227.

[10] M. Bause, J. Hoffmann, and P. Knabner. "First-order convergence of multipoint flux approximation on triangular grids and comparison with mixed finite element methods". In: *Numerische Mathematik* 116.1 (2010), pp. 1–29.

[11] A. Y. Beliaev and S. M. Hassanizadeh. "A theoretical model of hysteresis and dynamic effects in the capillary relation for two-phase flow in porous media". In: *Transport in Porous media* 43.3 (2001), pp. 487–510.

[12] M. Berardi, F. Difonzo, M. Vurro, and L. Lopez. "The 1D Richards' equation in two layered soils: a Filippov approach to treat discontinuities". In: *Advances in water resources* 115 (2018), pp. 264–272.

[13] L. Bergamaschi and M. Putti. "Mixed finite elements and Newton-type linearizations for the solution of Richards' equation". In: *International journal for numerical methods in engineering* 45.8 (1999), pp. 1025–1046.

[14] B. Bijeljic and M. J. Blunt. "Pore-scale modeling and continuous time random walk analysis of dispersion in porous media". In: *Water resources research* 42.1 (2006).

[15] J. W. Both, K. Kumar, J. M. Nordbotten, and F. A. Radu. "Anderson accelerated fixed-stress splitting schemes for consolidation of unsaturated porous media". In: *Computers & Mathematics with Applications* 77.6 (2019), pp. 1479–1502.

[16] C. Cances, I. Pop, and M. Vohralik. "An a posteriori error estimate for vertex-centered finite volume discretizations of immiscible incompressible two-phase flow". In: *Mathematics of Computation* 83.285 (2014), pp. 153–188.

[17] X. Cao and I. S. Pop. "Uniqueness of weak solutions for a pseudo-parabolic equation modeling two phase flow in porous media". In: *Applied Mathematics Letters* 46 (2015), pp. 25–30.

[18] M. A. Celia, E. T. Bouloutas, and R. L. Zarba. "A general mass-conservative numerical solution for the unsaturated flow equation". In: *Water resources research* 26.7 (1990), pp. 1483–1496.

[19] J. Chen, J. W. Hopmans, and M. E. Grismer. "Parameter estimation of two-fluid capillary pressure–saturation and permeability functions". In: *Advances in Water Resources* 22.5 (1999), pp. 479–493.

[20] N. Christofi and I. B. Ivshina. "Microbial surfactants and their use in field studies of soil remediation". In: *Journal of Applied Microbiology* 93.6 (2002), pp. 915–929.

[21] C. Dawson. "Analysis of an upwind-mixed finite element method for nonlinear contaminant transport equations". In: *SIAM journal on numerical analysis* 35.5 (1998), pp. 1709–1724.

[22] D. A. DiCarlo. "Experimental measurements of saturation overshoot on infiltration". In: *Water Resources Research* 40.4 (2004).

[23] R. E. Edwards. *Functional analysis: theory and applications*. Courier Corporation, 1995.

[24] C. Evans, S. Pollock, L. G. Rebholz, and M. Xiao. "A Proof That Anderson Acceleration Improves the Convergence Rate in Linearly Converging Fixed-Point Methods (But Not in Those Converging Quadratically)". In: *SIAM journal on numerical analysis* 58.1 (2020), pp. 788–810.

[25] R. Eymard, M. Gutnic, and D. Hilhorst. "The finite volume method for Richards equation". In: *Computational Geosciences* 3.3-4 (1999), pp. 259–294.

[26] R. Eymard, D. Hilhorst, and M. Vohralik. "A combined finite volume–nonconforming/mixed-hybrid finite element scheme for degenerate parabolic problems". In: *Numerische Mathematik* 105.1 (2006), pp. 73–131.

[27] H.-R. Fang and Y. Saad. "Two classes of multisecant methods for nonlinear acceleration". In: *Numerical Linear Algebra with Applications* 16.3 (2009), pp. 197–221.

[28] M. W. Farthing and F. L. Ogden. "Numerical solution of Richards' equation: A review of advances and challenges". In: *Soil Science Society of America Journal* 81.6 (2017), pp. 1257–1269.

[29] R. Fucik, J. Mikyska, T. Sakaki, M. Benes, and T. H. Illangasekare. "Significance of dynamic effect in capillarity during drainage experiments in layered porous media". In: *Vadose Zone Journal* 9.3 (2010), pp. 697–708.

[30] C. Gallo and G. Manzini. "A mixed finite element/finite volume approach for solving biodegradation transport in groundwater". In: *International journal for numerical methods in fluids* 26.5 (1998), pp. 533–556.

[31] S. M. Hassanizadeh, M. A. Celia, and H. K. Dahle. "Dynamic effect in the capillary pressure–saturation relationship and its impacts on unsaturated flow". In: *Vadose Zone Journal* 1.1 (2002), pp. 38–57.

[32] R. Haverkamp, M. Vauclin, J. Touma, P. J. Wierenga, and G. Vachaud. "A comparison of numerical simulation models for one-dimensional infiltration". In: *Soil Science Society of America Journal* 41.2 (1977), pp. 285–294.

[33] R. Helmig. *Multiphase flow and transport processes in the subsurface: a contribution to the modeling of hydrosystems*. Springer-Verlag, 1997.

[34] R. Helmig, A. Weiss, and B. I. Wohlmuth. "Dynamic capillary effects in heterogeneous porous media". In: *Computational Geosciences* 11.3 (2007), pp. 261–274.

[35] E. J. Henry, J. E. Smith, and A. W. Warrick. "Solubility effects on surfactant-induced unsaturated flow through porous media". In: *Journal of hydrology* 223.3-4 (1999), pp. 164–174.

[36] N. T. Hoa, R. Gaudu, and C. Thirriot. "Influence of the hysteresis effect on transient flows in saturated-unsaturated porous media". In: *Water Resources Research* 13.6 (1977), pp. 992–996.

[37] D. Husseini. "Effects of anions acids on surface tension of water". In: *Undergraduate Research at JMU Scholarly Commons* (2015).

[38] J. D. Hyman, P. K. Smolarkiewicz, and C. L. Winter. "Heterogeneities of flow in stochastically generated porous media". In: *Physical Review E* 86.5 (2012).

[39] D. Illiano, J. W. Both, I. S. Pop, and F. A. Radu. "Efficient Solvers for Non-standard Models for Flow and Transport in Unsaturated Porous Media". In: *arXiv preprint arXiv:2012.14773, submitted manuscript* (2020).

[40] D. Illiano, I. S. Pop, and F. A. Radu. "An efficient numerical scheme for fully coupled flow and reactive transport in variably saturated porous media including dynamic capillary effects". In: *arXiv preprint arXiv:1912.06731, accepted author manuscript* (2020).

[41] D. Illiano, I. S. Pop, and F. A. Radu. "Iterative schemes for surfactant transport in porous media". In: *Computational Geosciences* (2020), pp. 1–18.

[42] J. E. Jones and C. S. Woodward. "Newton–Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems". In: *Advances in Water Resources* 24.7 (2001), pp. 763–774.

[43] A. Karagunduz, M. H. Young, and K. D. Pennell. "Influence of surfactants on unsaturated water flow and solute transport". In: *Water Resources Research* 51.4 (2015), pp. 1977–1988.

[44] S. Karpinski, I. S. Pop, and F. A. Radu. "Analysis of a linearization scheme for an interior penalty discontinuous Galerkin method for two-phase flow in porous media with dynamic capillarity effects". In: *International Journal for Numerical Methods in Engineering* 112.6 (2017), pp. 553–577.

[45] R. A. Klausen, F. A. Radu, and G. T. Eigestad. "Convergence of MPFA on triangulations and for Richards' equation". In: *International journal for numerical methods in fluids* 58.12 (2008), pp. 1327–1351.

[46] P. Knabner. *Finite element simulation of saturated-unsaturated flow through porous media.* 1987.

[47] P. Knabner, S. Bitterlich, R. I. Teran, A. Prechtel, and E. Schneid. "Influence of surfactants on spreading of contaminants and soil remediation". In: (2003), pp. 152–161.

[48] D. A. Knoll and D. E. Keyes. "Jacobian-free Newton–Krylov methods: a survey of approaches and applications". In: *Journal of Computational Physics* 193.2 (2004), pp. 357–397.

[49] K. Kumar, I. S. Pop, and F. A. Radu. "Convergence analysis of mixed numerical schemes for reactive flow in a porous medium". In: *SIAM lournal on numerical analysis* 51.4 (2013), pp. 2283–2308.

[50] E. M. LaBolle, G. E. Fogg, and A. F. B. Tompson. "Random-walk simulation of transport in heterogeneous porous media: Local mass-conservation problem and implementation methods". In: *Water Resources Research* 32.3 (1996), pp. 583–593.

[51] S. H. Lee and Y. Efendiev. "C1-Continuous relative permeability and hybrid upwind discretization of three phase flow in porous media". In: *Advances in Water Resources* 96 (2016), pp. 209–224.

[52] H. Li, M. W. Farthing, C. N. Dawson, and C. T. Miller. "Local discontinuous Galerkin approximations to Richards' equation". In: *Advances in water resources* 30.3 (2007), pp. 555–575.

[53] R. F. Li, W. Yan, S. Liu, G. Hirasaki, C. A. Miller, et al. "Foam mobility control for surfactant enhanced oil recovery". In: *SPE Journal* 15.04 (2010), pp. 928–942.

[54] K.-A. Lie. *An introduction to reservoir simulation using MATLAB/GNU Octave: User guide for the MATLAB Reservoir Simulation Toolbox (MRST).* Cambridge University Press, 2019.

[55] F. List and F. A. Radu. "A study on iterative methods for solving Richards' equation". In: *Computational Geosciences* 20.2 (2016), pp. 341–353.

[56] S. B. Lunowa, I. S. Pop, and B. Koren. "Linearized domain decomposition methods for two-phase porous media flow models involving dynamic capillarity and hysteresis". In: *Computer Methods in Applied Mechanics and Engineering* 372 (2020).

[57] J. E. McClure, R. T. Armstrong, M. A. Berrill, S. Schluter, S. Berg, W. G. Gray, and C. T. Miller. "Geometric state function for two-fluid flow in porous media". In: *Physical Review Fluids* 3.8 (2018).

[58] A. Mikelic. "A global existence result for the equations describing unsaturated flow in porous media with dynamic capillary pressure". In: *Journal of Differential Equations* 248.6 (2010), pp. 1561–1577.

[59] K. Mitra and I. S. Pop. "A modified L-scheme to solve nonlinear diffusion problems". In: *Computers & Mathematics with Applications* 77.6 (2019), pp. 1722–1738.

[60] Catherine N Mulligan, RN Yong, and BF Gibbs. "Surfactant-enhanced remediation of contaminated soil: a review". In: *Engineering geology* 60.1-4 (2001), pp. 371–380.

[61] R. H. Nochetto and C. Verdi. "Approximation of degenerate parabolic problems using numerical integration". In: *SIAM journal on numerical analysis* 25.4 (1988), pp. 784–814.

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

[63] C.-H. Park, C. Beyer, S. Bauer, and O. Kolditz. "A study of preferential flow in heterogeneous media using random walk particle tracking". In: *Geosciences Journal* 12.3 (2008), pp. 285–297.

[64] I. S. Pop, F. A. Radu, and P. Knabner. "Mixed finite elements for the Richards' equation: linearization procedure". In: *Journal of computational and applied mathematics* 168.1-2 (2004), pp. 365–373.

[65] A. Prechtel and P. Knabner. "Accurate and efficient simulation of coupled water flow and nonlinear reactive transport in the saturated and vadose zone—application to surfactant enhanced and intrinsic bioremediation". In: *Developments in Water Science* 47 (2002), pp. 687–694.

[66] F. Radu, I. S. Pop, and P. Knabner. "Order of convergence estimates for an Euler implicit, mixed finite element discretization of Richards' equation". In: *SIAM journal on numerical analysis* 42.4 (2004), pp. 1452–1478.

[67] F. A. Radu, A. Muntean, I. S. Pop, N. Suciu, and O. Kolditz. "A mixed finite element discretization scheme for a concrete carbonation model with concentration-dependent porosity". In: *Journal of Computational and Applied Mathematics* 246 (2013), pp. 74–85.

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

[69] F. A. Radu, I. S. Pop, and P. Knabner. "On the convergence of the Newton method for the mixed finite element discretization of a class of degenerate parabolic equation". In: *Numerical Mathematics and Advanced Applications* 42 (2006), pp. 1194–1200.

[70] F. A. Radu, N. Suciu, J. Hoffmann, A. Vogel, O. Kolditz, C.-H. Park, and S. Attinger. "Accuracy of numerical simulations of contaminant transport in heterogeneous aquifers: a comparative study". In: *Advances in water resources* 34.1 (2011), pp. 47–61.

[71] P. C. Reeves and M. A. Celia. "A functional relationship between capillary pressure, saturation, and interfacial area as revealed by a pore-scale network model". In: *Water resources research* 32.8 (1996), pp. 2345–2358.

[72] S. Reitsma and B. H. Kueper. "Laboratory measurement of capillary pressure-saturation relationships in a rock fracture". In: *Water Resources Research* 30.4 (1994), pp. 865–878.

[73] T. F. Russell and M. F. Wheeler. "Finite element and finite difference methods for continuous flows in porous media". In: (1983), pp. 35–106.

[74] B. Schweizer. "The Richards equation with hysteresis and degenerate capillary pressure". In: *Journal of Differential Equations* 252.10 (2012), pp. 5594–5612.

[75] T. Shubao, L. Gang, H. Shunli, and Y. Limin. "Dynamic effect of capillary pressure in low permeability reservoirs". In: *Petroleum exploration and development* 39.3 (2012), pp. 405–411.

[76] M. Slodicka. "A robust and efficient linearization scheme for doubly nonlinear and degenerate parabolic problems arising in flow in porous media". In: *SIAM journal on scientific computing* 23.5 (2002), pp. 1593–1614.

[77] J. E. Smith and R. W. Gillham. "Effects of solute concentration–dependent surface tension on unsaturated flow: Laboratory sand column experiments". In: *Water Resources Research* 35.4 (1999), pp. 973–982.

[78] J. E. Smith and R. W. Gillham. "The effect of concentration-dependent surface tension on the flaw of water and transport of dissolved organic compounds: A pressure head-based formulation and numerical model". In: *Water Resources Research* 30.2 (1994), pp. 343–354.

[79] N. Suciu. *Diffusion in Random Fields: Applications to Transport in Groundwater*. Springer, 2019.

[80] N. Suciu. "Diffusion in random velocity fields with applications to contaminant transport in groundwater". In: *Advances in water resources* 69 (2014), pp. 114–133.

[81] N. Suciu, F. A. Radu, A. Prechtel, F. Brunner, and P. Knabner. "A coupled finite element–global random walk approach to advection-dominated transport in porous media with random hydraulic conductivity". In: *Journal of Computational and Applied Mathematics* 246 (2013), pp. 27–37.

[82] M. T. Van Genuchten. "A closed-form equation for predicting the hydraulic conductivity of unsaturated soils". In: *Soil science society of America journal* 44.5 (1980), pp. 892–898.

[83] M. Vohralik. "A posteriori error estimates for lowest-order mixed finite element discretizations of convection-diffusion-reaction equations". In: *SIAM journal on numerical analysis* 45.4 (2007), pp. 1570–1599.

[84] H. F. Walker and P. Ni. "Anderson acceleration for fixed-point iterations". In: *SIAM journal on numerical analysis* 49.4 (2011), pp. 1715–1735.

[85] X. Wang and H. A. Tchelepi. "Trust-region based solver for nonlinear transport in heterogeneous porous media". In: *Journal of Computational Physics* 253 (2013), pp. 114–137.

[86] Y. Wang et al. "Exploring the effects of different types of surfactants on zebrafish embryos and larvae". In: *Scientific reports* 5 (2015).

[87] C. S. Woodward and C. N. Dawson. "Analysis of expanded mixed finite element methods for a nonlinear parabolic equation modeling flow into variably saturated porous media". In: *SIAM journal on numerical analysis* 37.3 (2000), pp. 701–724.

[88] W.-A. Yong and I. S. Pop. *A numerical approach to porous medium equations*. IWR, 1996.

[89] R. Younis, H. A. Tchelepi, and K. Aziz. "Adaptively Localized Continuation-Newton Method–Nonlinear Solvers That Converge All the Time". In: *Society of Petroleum Engineers Journal* 15.02 (2010), pp. 526–544.

[90] L. Zhuang, C. J. van Duijn, and S. M. Hassanizadeh. "The effect of dynamic capillarity in modeling saturation overshoot during infiltration". In: *Vadose Zone Journal* 18.1 (2019), pp. 1–14.

Part II

Scientific results

A

Paper A

Iterative schemes for surfactant transport in porous media

Davide Illiano, Iuliu Sorin Pop, Florin Adrian Radu

Iterative schemes for surfactant transport in porous media. Comput Geosci, D. Illiano, I. S. Pop, F. A. Radu, (2020)
DOI:https://doi.org/10.1007/s10596-020-09949-2

Computational Geosciences
https://doi.org/10.1007/s10596-020-09949-2

ORIGINAL PAPER

Iterative schemes for surfactant transport in porous media

Davide Illiano[1] · **Iuliu Sorin Pop**[1,2] · **Florin Adrian Radu**[1]

Received: 31 May 2019 / Accepted: 21 February 2020

Abstract
In this work, we consider the transport of a surfactant in variably saturated porous media. The water flow is modelled by the Richards equations and it is fully coupled with the transport equation for the surfactant. Three linearization techniques are discussed: the Newton method, the modified Picard, and the L-scheme. Based on these, monolithic and splitting schemes are proposed and their convergence is analyzed. The performance of these schemes is illustrated on five numerical examples. For these examples, the number of iterations and the condition numbers of the linear systems emerging in each iteration are presented.

Keywords Richards equation · Reactive transport · Linearization schemes · L-scheme · Modified Picard · Newton method · Splitting solvers

1 Introduction

Many societally relevant applications are involving multiphase flow and multicomponent reactive transport in porous media. Examples in this sense appear in the enhanced oil recovery, geological CO_2 storage, diffusion of medical agents into the human body, or water or soil pollution. In many situations like these, experimental results are difficult and expensive to obtain, therefore numerical simulations become a key technology. Together with laboratory experiments and field data, they provide the key tools in understanding such complex phenomena. The mathematical models for problems as mentioned above are (fully or partially) coupled, nonlinear, possible degenerate partial differential equations. In most cases, deriving explicit solutions is not possible, whereas developing appropriate algorithms for finding numerical solutions is a challenge in itself. Here we investigate robust and efficient methods for solving the nonlinear problems obtained after performing an implicit time discretization. The focus is on iterative, splitting, or monolithic schemes for fully coupled flow and transport.

Of particular interest here is a special case of multiphase, reactive flow in porous media, namely the surfactant transport in soil [2, 19, 23, 25, 27, 33]. Surfactants, which are usually organic compounds, are commonly used for actively combating soil and water pollution [11, 12, 16, 38, 43]. They contain both hydrophobic and hydrophilic groups and are dissolved in the water phase, being transported by diffusion and convection. Typically, the surfactants are employed in soil regions near the surface (vadose zone), where water and air are present in the pores. Consequently, the outcoming mathematical model accounts the transport of at least one species (the surfactant, but often also the contaminant) in a variably saturated porous medium. Whereas the dependence of the species transported from the flow is obvious, one can encounter the reverse dependence as well when surfactants are affecting the interfacial tension between water and air, leading to a dependency of the water flow on the concentration of surfactant. In other words, one has to cope with a fully coupled flow and transport problem, and not only with a one-way coupling, i.e., when only the transport depends on the flow, as mostly considered in reactive transport [35].

Whereas the surfactant transport is described by a reaction-diffusion-convection equation, water flow in variably saturated porous media is modelled by the Richards

✉ Iuliu Sorin Pop
sorin.pop@uhasselt.be; Iuliu.Pop@uib.no

Davide Illiano
Davide.Illiano@uib.no

Florin Adrian Radu
Florin.Radu@uib.no

[1] Department of Mathematics, University of Bergen, Allegaten 41, Bergen, Norway

[2] Faculty of Sciences, University of Hasselt, Agoralaan Building D, BE 3590, Diepenbeek, Belgium

Published online: 14 June 2020

equation [7, 18]. The main assumption in this case is that the air remains in contact with the atmosphere, having a constant pressure (the atmospheric pressure, here assumed zero). This allows reducing the flow model to one equation, the Richards equation. In mathematical terms, this equation is degenerate parabolic, whose solution has typically low regularity [3].

From the above, and adopting the pressure head as the main unknown in the Richards equation, we study here different linearization schemes for the model

$$\frac{\partial \theta(\Psi, c)}{\partial t} - \nabla \cdot (K(\theta(\Psi, c))\nabla(\Psi + z)) = H_1 \tag{1}$$

and

$$\frac{\partial \theta(\Psi, c)c}{\partial t} - \nabla \cdot (D\nabla c - \mathbf{u}_{\mathbf{w}} c) + R(c) = H_2, \tag{2}$$

holding for $\mathbf{x} \in \Omega$ (z being the vertical coordinate of $\mathbf{x}$, pointing against gravity) and $t \in (0, T]$. Here Ω is a bounded, open domain in $\mathbb{R}^d$ ($d = 1, 2$ or 3) having a Lipschitz continuous boundary $\partial\Omega$ and $T > 0$ is the final time. Further, $\theta(\cdot, \cdot)$ denotes the water content, and is a given function depending on the pressure head Ψ and on the surfactant concentration c. Also, $K(\cdot)$ is the hydraulic conductivity, $D > 0$ the diffusion/dispersion coefficient. Finally, $\mathbf{u}_{\mathbf{w}} := -K(\theta, c)\nabla(\Psi + z)$ is the water flux, $R(\cdot)$ the reaction term, expressed as a function of the concentration c, and H_1, H_2 are the external sinks/sources. Initial and boundary conditions, which are specified below, complete the system.

We point out that the water content and the hydraulic conductivity, $\theta(\cdot, \cdot)$ and $K(\cdot)$ are given nonlinear functions. They are medium- and surfactant-dependent and are determined experimentally (see [18]). Specific choices are provided in Section 2.

To solve numerically the system (1)–(2) one needs to discretize in time and space. We refer to [15] for a practical review of numerical methods for the Richards equation. Due to the low regularity of the solution and the need of relatively large time steps, the backward Euler method is the best candidate for the time discretization. Multiple spatial discretization techniques are available, such as the Galerkin finite element method (FEM) [5, 32, 39], the mixed finite element method (MFEM) [4, 36, 44, 47], the multi-point flux approximation (MPFA) [1, 6, 24], and the finite volume method (FVM) [9, 13, 14].

Since the time discretization is not explicit, the outcome is a sequence of nonlinear problems, for which a linearization step has to be performed. Widely used linearization schemes are the quadratic, locally convergent Newton method and the modified Picard method [10]. For both, the convergence is guaranteed if the starting point is close to the solution. Since for evolution equations the initial guess is typically the solution at the previous time, this may induces severe restrictions on the time step size (see [37]). There exist several modifications of the Newton scheme improving this aspect, including like line-search and trust-region methods, or Anderson acceleration techniques, as discussed, e.g., in [15, 21, 26, 28, 45, 46, 49], or exploiting the structure of the nonlinearity appearing in the hyperbolic two-phase flow model, as discussed in [22]. Among alternative approaches we mention the L-scheme (see [30, 34, 40, 48]) and the modified L-scheme [31], both being robust w.r.t. the mesh size, but converging linearly. In particular, the L-scheme converges for any starting point, and the restriction on the time step, if any, is very mild. The modified L-scheme makes explicit use of the choice of the starting point as the solution obtained at the previous time, and has an improved convergence behavior if the changes in the solutions at two successive times are controlled by the time step. Nevertheless, the modified L-scheme involves computation of derivatives while the L-scheme does not. Finally, the robustness of the Newton method is significantly increased if one considers combinations of the Picard and the Newton methods [8], and in particular of the L-scheme and the Newton scheme [30].

We conclude this discussion by mentioning that in this paper we adopt the FEM and the FVM, but the iterative schemes presented here can be applied in combination with any other spatial discretization. The focus is on effectively solving the flow and transport system (1)–(2), and in particular on the adequate treating of the coupling between the two model components (the flow and the reactive transport). The schemes are divided into three main categories: monolithic (Mon), nonlinear splitting (NonLinS) and alternate splitting (AltS). Subsequently, we denote, e.g., by Mon-Newton, the monolithic scheme obtained by applying the Newton method as linearization. The nonlinear splitting schemes (NonLinS) should be understood as solving at each time step first the flow equation until convergence, by using the surfactant concentration from the last iteration, and then with the obtained flow solving the transport equation until convergence. The procedure can be continued iteratively, this being the usual *or classical* splitting method for transport problems. The convergence of NonLinS does not depend on the linearization approach used for each model component (Newton, Picard, or L-scheme), because we assume that the nonlinear subproblems are solved exactly, i.e., until convergence. Finally, the alternate splitting methods (AltS) have a different philosophy. Instead of solving each subproblem until convergence within each iteration, one performs only one step of the chosen linearization. For example, AltS-NE will perform one Newton step for each model component, and iterate. These schemes are illustrated in Figs. 1 and 2.

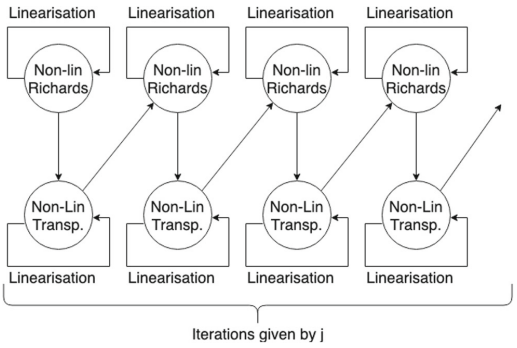


Fig. 1 The nonlinear splitting approach

All the schemes can be analyzed theoretically, and we do this exemplary for Mon-LS, i.e., for the monolithic approach combined with the L-scheme. Based on comparative numerical tests performed for academic and benchmark problems, we see that the alternate methods can save substantial computational time, while maintaining the robustness of the L-scheme.

The remaining of the paper is organized as follows. In Section 2, we establish the mathematical model and the notation used and present the iterative monolithic and splitting schemes. In Section 3, we prove the convergence of the Mon-LS scheme and briefly discuss the convergence of the other schemes. Section 4 presents five different numerical examples. They are inspired by the cases already studied in the literature [25, 30]. Section 5 concludes this work.

2 Problem formulation, discretization, and iterative schemes

We solve the fully coupled system (1)–(2), completed by homogeneous Dirichlet boundary conditions for both Ψ and c and the initial conditions:

$$\Psi = \Psi_0 \text{ and } c = c_0 \text{ at } t = 0.$$

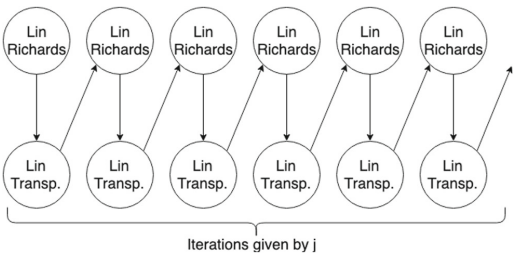


Fig. 2 The alternate splitting approach

We use the van Genuchten-Mualem parameterization [17]

$$\theta(\Psi) = \begin{cases} \theta_r + (\theta_s - \theta_r)\left(\frac{1}{1+(-\alpha\Psi)^n}\right)^{\frac{n-1}{n}}, & \Psi \le 0 \\ \theta_s, & \Psi > 0, \end{cases} \tag{3}$$

$$K(\theta(\Psi)) = \begin{cases} K_s\theta_e(\Psi)^{\frac{1}{2}}\left[1-\left(1-\theta_e(\Psi)^{\frac{n}{n-1}}\right)^{\frac{n-1}{n}}\right], & \Psi \le 0 \\ K_s, & \Psi > 0, \end{cases} \tag{4}$$

where θ_r and θ_s represent the values of the residual and saturated water content, $\theta_e = (\theta - \theta_r)/(\theta_s - \theta_r)$ is the effective water content, K_s is the conductivity, and α and n are model parameters depending on the soil.

Observe that in the expression above for θ, the influence of the surfactant on the water flow is neglected. As reported in [20, 25, 42], the surface tension between water and air does depend on the surfactant concentration c, implying the same for the function θ above. The following parametrization is proposed in [25]

$$\theta(\Psi, c) := \theta(\gamma(c)\Psi), \quad \text{with} \quad \gamma(c) = \frac{1}{1 - b\log(c/a + 1)}. \tag{5}$$

Here $\theta()$ is given in (3) and $\gamma()$ is the surface tensions as depending on the concentration c. The parameters a and b depend on the fluid and the medium. We refer to [41, 42] for details about (5).

This gives the following expressions for θ and K

$$\theta(\Psi, c) = \begin{cases} \theta_r + (\theta_s - \theta_r)\left[1/\left(1+\left(-\alpha(\frac{1}{1-b\log(c/a+1)})\Psi\right)^n\right)\right]^{\frac{n-1}{n}}, & \Psi \le 0 \\ \theta_s, & \Psi > 0, \end{cases} \tag{6}$$

$$K(\theta(\Psi, c)) = \begin{cases} K_s\theta_e(\Psi, c)^{\frac{1}{2}}\left[1-\left(1-\theta_e(\Psi, c)^{\frac{n}{n-1}}\right)^{\frac{n-1}{n}}\right], & \Psi \le 0 \\ K_s, & \Psi > 0. \end{cases} \tag{7}$$

This shows that the flow component also depends on the reactive transport, implying that the model is coupled in both directions.

In the following, we proceed by discretizing (1) and (2) in time and space. We will use common notations in functional analysis. We denote by $L^2(\Omega)$ the space of real valued, squared integrable functions defined on Ω and $H^1(\Omega)$ its subspace, containing the functions having also the first order derivatives in $L^2(\Omega)$. $H_0^1(\Omega)$ is the space of functions belonging to $H^1(\Omega)$ and vanishing on $\partial\Omega$. Further, we denote by $< \cdot, \cdot >$ the $L^2(\Omega)$ scalar product (and by $\|\cdot\|$ the associated norm) or the pairing between H_0^1 and its dual H^{-1}. Finally, by $L^2(0, T; X)$, we mean the Bochner

space of functions taking values in the Banach-space X, the extension to $H^1(0,T;X)$ being straightforward.

With this, we state the weak formulation of the problem related to (1)–(2):

Problem P: Find $\Psi, c \in L^2(0,T;H_0^1(\Omega)) \cap H^1(0,T;H^{-1}(\Omega))$ such that

$$< \partial_t \theta(\Psi,c), v_1 > + < K(\theta(\Psi,c))\nabla(\Psi+z), \nabla v_1 > = < H_1, v_1 > \quad (8)$$

and

$$< \partial_t(\theta(\Psi,c)c), v_2 > + < D\nabla c + \mathbf{u_w}c, \nabla v_2 > = < H_2, v_2 > \quad (9)$$

hold for all $v_1, v_2 \in H_0^1(\Omega)$ and almost every $t \in (0,T]$.

We now combine the backward Euler method with linear Galerkin finite elements for the discretization of Problem P. We let $N \in \mathbb{N}$ be a strictly positive natural number and the time step $\tau := T/N$. Correspondingly, the discrete times are $t_n := n\tau$ ($n \in \{0,1,\ldots,N\}$). Further, we let T_h be a regular decomposition of Ω, $\overline{\Omega} = \bigcup_{T\in T_h} T$ into d-dimensional simplices, with h denoting the mesh diameter. The finite element space $V_h \subset H_0^1(\Omega)$ is defined by

$$V_h := \{v_h \in H_0^1(\Omega)\ s.t.\ v_{h|T} \in \mathbb{P}_1(T),\ \text{for any } T \in T_h\}, \quad (10)$$

where $\mathbb{P}_1(T)$ denotes the space of linear polynomials on T and $v_{h|T}$ the restriction of v_h to T.

For the fully discrete counterpart of Problem P, we let $n \geq 1$ be fixed and assume that $\Psi_h^{n-1}, c_h^{n-1} \in V_h$ are given. The solution pair at time t_n solves

Problem $\mathbf{P}_n$: Find $\Psi_h^n, c_h^n \in V_h$ such that for all $v_h, w_h \in V_h$ there holds

$$< \theta(\Psi_h^n, c_h^n) - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right), v_h > + \tau < K(\theta(\Psi_h^n, c_h^n))(\nabla(\Psi_h^n) + \mathbf{e_z}), \nabla v_h > = \tau < H_1, v_h > \quad (11)$$

and

$$< \theta\left(\Psi_h^n, c_h^n\right)c_h^n - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right)c_h^{n-1}, w_h > + \tau < D\nabla c_h^n + \mathbf{u_w^{n-1}}c_h^n, \nabla w_h > = \tau < H_2, w_h > . \quad (12)$$

$\mathbf{e_z}$ denotes the unit vector in the direction opposite to gravity.

Remark 1 Observe that $\mathbf{u_w^{n-1}}$ appears in the convective term in (12). This choice is made for the ease of presentation. Nevertheless, all calculations carried out in this paper were doubled by ones where $\mathbf{u_w^n}$ has replaced $\mathbf{u_w^{n-1}}$. The differences in the results were marginal.

Observe that Problem P_n is a coupling system of two elliptic, nonlinear equations. In the following, we discuss different iterative schemes for solving this system.

2.1 Iterative linearization schemes

We discuss monolithic and splitting approaches for solving Problem P_n, combined with either the Newton method, the modified Picard [10], or the L-scheme [30, 34]. In the following the index n always refers to the time step, whereas j denotes the iteration index. As a rule, the iterations start with the solution at the previous time, t_{n-1}.

In the monolithic approach, one solves the two equations of the system (11)–(12) at once, combined with a linearization method. Formally, this becomes

Problem $\mathbf{PMon}_{n,j+1}$: Find $\Psi^{n,j+1}$ and $c^{n,j+1}$ such that

$$\begin{cases} F_1^{lin}\left(\Psi^{n,j+1}, c^{n,j+1}\right) = 0, \\ F_2^{lin}\left(\Psi^{n,j+1}, c^{n,j+1}\right) = 0. \end{cases} \quad (13)$$

F_k^{Lin} is a linearization of the expression F_k ($k = 1,2$) appearing in the system (11)–(12). Depending on the used linearization technique, one speaks about a monolithic Newton scheme (Mon-Newton), or monolithic Picard (Mon-Picard) or monolithic L-scheme (Mon-LS). These three schemes will be presented in detail below.

In the iterative splitting approach one solves each equation separately and then iterates between these, using the results previously obtained. We distinguish between two main splitting ways: the nonlinear splitting and the alternate splitting. These are schematized in Figs. 1 and 2 respectively. The former becomes :

Problem $\mathbf{PNonLinS}_{n,j+1}$: Find $\Psi^{n,j+1}$ and $c^{n,j+1}$ such that

$$\begin{cases} F_1\left(\Psi^{n,j+1}, c^{n,j}\right) = 0, \ \text{followed by} \\ F_2\left(\Psi^{n,j+1}, c^{n,j+1}\right) = 0. \end{cases} \quad (14)$$

For the linearization of F_1 and F_2, one can use one of the three linearization techniques mentioned before. In contrast, in the alternate splitting, one performs only one linearization step per iteration (see also Fig. 2). The alternate splitting scheme becomes

Problem $\mathbf{PAltS}_{n,j+1}$: Find $\Psi^{n,j+1}$ and $c^{n,j+1}$ such that

$$\begin{cases} F_1^{lin}\left(\Psi^{n,j+1}, c^{n,j}\right) = 0, \ \text{followed by} \\ F_2^{lin}\left(\Psi^{n,j+1}, c^{n,j+1}\right) = 0. \end{cases} \quad (15)$$

Depending on which linearization is used, one speaks about alternate splitting Newton (AltS-NE), alternate splitting Picard (AltS-Picard), or alternate splitting L-scheme (AltS-LS). The schemes are presented in detail below.

2.1.1 The monolithic Newton method (Mon-Newton)

We recall that the Newton scheme is quadratically, but only locally convergent. The monolithic Newton method applied to (11)–(12) gives

Problem PMon-Newton$_{n,j+1}$: Let $\Psi_h^{n-1}, c^{n-1}, \Psi_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1}, c_h^{n,j+1} \in V_h$ such that for all $v_h, w_h \in V_h$ one has

$$
\begin{aligned}
&< \theta\left(\Psi_h^{n,j}, c_h^{n,j}\right) - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right), v_h > \\
&\quad + < \frac{\partial\theta}{\partial\Psi}\left(\Psi_h^{n,j}, c_h^{n,j}\right)\left(\Psi_h^{n,j+1} - \Psi_h^{n,j}\right), v_h > \\
&\quad + \tau < K\left(\theta\left(\Psi_h^{n,j}, c_h^{n,j}\right)\right)\left(\nabla\Psi_h^{n,j+1} + \mathbf{e_z}\right), \nabla v_h > \\
&\quad + \tau < K'\left(\theta\left(\Psi_h^{n,j}, c_h^{n,j}\right)\right)\frac{\partial\theta}{\partial\Psi}\left(\Psi_h^{n,j}, c_h^{n,j}\right) \\
&\quad \times \left(\nabla\Psi_h^{n,j} + \mathbf{e_z}\right)\left(\Psi_h^{n,j+1} - \Psi_h^{n,j}\right), \nabla v_h > \\
&= \tau < H_1, v_h > \qquad (16)
\end{aligned}
$$

and

$$
\begin{aligned}
&< \theta\left(\Psi_h^{n,j}, c_h^{n,j}\right) c_h^{n,j+1} - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right) c_h^{n-1}, w_h > \\
&\quad + < \frac{\partial\theta}{\partial c}\left(\Psi_h^{n,j}, c_h^{n,j}\right)\left(c_h^{n,j+1} - c_h^{n,j}\right), v_h > \\
&\quad + \tau < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla w_h > \\
&= \tau < H_2, w_h > . \qquad (17)
\end{aligned}
$$

For the ease of presentation, here a simplified monolithic formulation is given, involving only the derivative of θ with respect to Ψ in (16), and only the derivative of θ with respect to c in (17). In the full monolithic approach, both partial derivatives should be involved for all nonlinear functions, e.g.,

$$
\begin{aligned}
&\theta\left(\Psi_h^{n,j+1}, c_h^{n,j+1}\right) \to \theta\left(\Psi_h^{n,j}, c_h^{n,j}\right) \\
&+ \left(\frac{\partial\theta}{\partial\Psi}\right)\left(\Psi_h^{n,j}, c_h^{n,j}\right)\left(\Psi_h^{n,j+1} - \Psi_h^{n,j}\right) \\
&+ \left(\frac{\partial\theta}{\partial c}\right)\left(\Psi_h^{n,j}, c_h^{n,j}\right)\left(c_h^{n,j+1} - c_h^{n,j}\right). \qquad (18)
\end{aligned}
$$

However, we have carried out computations with the full monolithic approach and the results were practically showing no difference.

2.1.2 The monolithic Picard method (Mon-Picard)

The modified Picard method was initially proposed by Celia [10] for the Richards equation. It is similar to the Newton method in dealing with the nonlinearity in the saturation, but not in the permeability. Being a modification of the Newton method, the modified Picard method is only linearly convergent [37]. The monolithic Picard method applied to (11)–(12) becomes

Problem PMon-Picard$_{n,j+1}$: Let $\Psi_h^{n-1}, c_h^{n-1}, \Psi_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1}, c_h^{n,j+1} \in V_h$ such that for all $v_h, w_h \in V_h$, one has

$$
\begin{aligned}
&< \theta\left(\Psi_h^{n,j}, c_h^{n,j}\right) - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right), v_h > \\
&\quad + < \frac{\partial\theta}{\partial\Psi}\left(\Psi_h^{n,j}, c_h^{n,j}\right)\left(\Psi_h^{n,j+1} - \Psi_h^{n,j}\right), v_h > \\
&\quad + \tau < K\left(\theta\left(\Psi_h^{n,j}, c_h^{n,j}\right)\right)\left(\nabla\Psi_h^{n,j+1} + \mathbf{e_z}\right), \nabla v_h > \\
&= \tau < H_1, v_h > \qquad (19)
\end{aligned}
$$

and

$$
\begin{aligned}
&< \theta\left(\Psi_h^{n,j}, c_h^{n,j}\right) c_h^{n,j+1} - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right) c_h^{n-1}, w_h > \\
&\quad + < \frac{\partial\theta}{\partial c}\left(\Psi_h^{n,j}, c_h^{n,j}\right)\left(c_h^{n,j+1} - c_h^{n,j}\right), w_h > \\
&\quad + \tau < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla w_h > \\
&= \tau < H_2, w_h > . \qquad (20)
\end{aligned}
$$

As before, a Picard iteration for the full monolithic approach would involve both partial derivatives of θ.

2.1.3 The monolithic L-scheme (Mon-LS)

The monolithic L-scheme for solving (11)–(12) becomes

Problem PMon-LS$_{n,j+1}$: Let $\Psi_h^{n-1}, \Psi_h^{n,j}, c_h^{n-1}, c_h^{n,j} \in V_h$ be given and with $L_1, L_2 > 0$ large enough (as specified below), find $\Psi_h^{n,j+1}, c_h^{n,j+1} \in V_h$ s.t. for all $v_h, w_h \in V_h$

$$
\begin{aligned}
&< \theta\left(\Psi_h^{n,j}, c_h^{n,j}\right) - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right), v_h > \\
&\quad + L_1 < \Psi_h^{n,j+1} - \Psi_h^{n,j}, v_h > \\
&\quad + \tau < K\left(\theta\left(\Psi_h^{n,j}, c_h^{n,j}\right)\right)\left(\nabla\Psi_h^{n,j+1} + \mathbf{e_z}\right), \nabla v_h > \\
&= \tau < H_1, v_h >, \qquad (21)
\end{aligned}
$$

$$
\begin{aligned}
&< \theta\left(\Psi_h^{n,j}, c_h^{n,j}\right) c_h^{n,j+1} - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right) c_h^{n-1}, w_h > \\
&\quad + L_2 < c_h^{n,j+1} - c_h^{n,j}, w_h > \\
&\quad + \tau < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla w_h > \\
&= \tau < H_2, w_h > . \qquad (22)
\end{aligned}
$$

The parameters L_1 and L_2 should be large enough to ensure the convergence of the scheme (see Section 3). In practice, the values of L_1 and L_2 are connected to the maximal values of $\partial_\Psi\theta$ and $\partial_c\theta$ (recall that θ is assumed increasing in Ψ and in c).

The L-scheme does not involve the computations of derivatives, and the linear systems to be solved within each iteration are better conditioned compared with the ones

given by Newton or Picard methods (see [30]). Moreover, this scheme is (linearly) convergent for any initial guess for the iteration. Finally, the classical full monolithic approach is obtained by involving L_1 and L_2 in both of the equations.

2.1.4 The nonlinear splitting approach (NonLinS)

The nonlinear splitting approach for solving (11)–(12) becomes

Problem PNonLinS$_{n,j+1}$: Let $\Psi_h^{n-1}, c^{n-1}, \Psi_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1} \in V_h$ s.t.

$$\begin{aligned} &< \theta\left(\Psi_h^{n,j+1}, c_h^{n,j}\right) - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right), v_h > \\ &\quad +\tau < K\left(\theta\left(\Psi_h^{n,j}, c_h^{n,j}\right)\right)\left(\nabla\Psi_h^{n,j+1} + \mathbf{e_z}\right), \nabla v_h > \\ &= \tau < H_1, v_h > \end{aligned} \tag{23}$$

holds true for all $v_h \in V_h$. Then, with $\Psi_h^{n,j+1}$ obtained, find $c_h^{n,j+1} \in V_h$ such that for all $w_h \in V_h$, it holds

$$\begin{aligned} &< \theta\left(\Psi_h^{n,j+1}, c_h^{n,j+1}\right) c_h^{n,j+1} - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right) c_h^{n-1}, w_h > \\ &\quad +\tau < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla w_h > \\ &= \tau < H_2, w_h > . \end{aligned} \tag{24}$$

As for the monolithic schemes, one can apply the different linear iterative schemes to obtain fully linear versions of the splitting approach. This is done first to solve (23) and, once a solution to (23) is available, this is employed in the linearization of (24).

2.1.5 The alternate Newton method (AltS-Newton)

In the alternate Newton method applied to (11)–(12), one solves

Problem PAltS-Newton$_{n,j+1}$: Let $\Psi_h^{n-1}, c^{n-1}, \Psi_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1} \in V_h$ s.t.

$$\begin{aligned} &< \theta\left(\Psi_h^{n,j}, c_h^{n,j}\right) - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right), v_h > \\ &\quad + < \frac{\partial\theta}{\partial\Psi}\left(\Psi_h^{n,j}, c_h^{n,j}\right)\left(\Psi_h^{n,j+1} - \Psi_h^{n,j}\right), v_h > \\ &\quad +\tau < K\left(\theta\left(\Psi_h^{n,j}, c_h^{n,j}\right)\right)\left(\nabla\Psi_h^{n,j+1} + \mathbf{e_z}\right), \nabla v_h > \\ &\quad +\tau < K'\left(\theta\left(\Psi_h^{n,j}, c_h^{n,j}\right)\right)\frac{\partial\theta}{\partial\Psi}\left(\Psi_h^{n,j}, c_h^{n,j}\right) \\ &\quad \times\left(\nabla\Psi_h^{n,j} + \mathbf{e_z}\right)\left(\Psi_h^{n,j+1} - \Psi_h^{n,j}\right), \nabla v_h > \\ &= \tau < H_1, v_h > \end{aligned} \tag{25}$$

holds true for all $v_h \in V_h$. Then, with $\Psi_h^{n,j+1}$ obtained above, find $c_h^{n,j+1} \in V_h$ such that for all $w_h \in V_h$, one has

$$\begin{aligned} &< \theta\left(\Psi_h^{n,j+1}, c_h^{n,j}\right) c_h^{n,j+1} - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right) c_h^{n-1}, w_h > \\ &\quad + < \frac{\partial\theta}{\partial c}\left(\Psi_h^{n,j+1}, c_h^{n,j}\right)\left(c_h^{n,j+1} - c_h^{n,j}\right), v_h > \\ &\quad +\tau < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla w_h > \\ &= \tau < H_2, w_h > . \end{aligned} \tag{26}$$

2.1.6 The alternate Picard method (AltS-Picard)

The alternate Picard method applied to (11)–(12) becomes

Problem PAltS-Picard$_{n,j+1}$: Let $\Psi_h^{n-1}, c^{n-1}, \Psi_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1} \in V_h$ s.t.

$$\begin{aligned} &< \theta\left(\Psi_h^{n,j}, c_h^{n,j}\right) - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right), v_h > \\ &\quad + < \frac{\partial\theta}{\partial\Psi}\left(\Psi_h^{n,j}, c_h^{n,j}\right)\left(\Psi_h^{n,j+1} - \Psi_h^{n,j}\right), v_h > \\ &\quad +\tau < K\left(\theta\left(\Psi_h^{n,j}, c_h^{n,j}\right)\right)\left(\nabla\left(\Psi_h^{n,j+1}\right) + \mathbf{e_z}\right), \nabla v_h > \\ &= \tau < H_1, v_h > \end{aligned} \tag{27}$$

hold true for all $v_h \in V_h$. Then, with $\Psi_h^{n,j+1}$ obtained above, find $c_h^{n,j+1} \in V_h$ such that for all $w_h \in V_h$, one has

$$\begin{aligned} &< \theta\left(\Psi_h^{n,j+1}, c_h^{n,j}\right) c_h^{n,j+1} - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right) c_h^{n-1}, w_h > \\ &\quad + < \frac{\partial\theta}{\partial c}\left(\Psi_h^{n,j+1}, c_h^{n,j}\right)\left(c_h^{n,j+1} - c_h^{n,j}\right), w_h > \\ &\quad +\tau < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla w_h > \\ &= \tau < H_2, w_h > . \end{aligned} \tag{28}$$

2.1.7 The alternate L-scheme (AltS-LS)

The alternate L-scheme for solving (11)–(12) becomes

Problem PAltS-LS$_{n,j+1}$: Let $\Psi_h^{n-1}, c^{n-1}, \Psi_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1} \in V_h$ s.t.

$$\begin{aligned} &< \theta\left(\Psi_h^{n,j}, c_h^{n,j}\right) - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right), v_h > \\ &\quad +L_1 < \Psi_h^{n,j+1} - \Psi_h^{n,j}, v_h > \\ &\quad +\tau < K\left(\theta\left(\Psi_h^{n,j}, c_h^{n,j}\right)\right)\left(\nabla\Psi_h^{n,j+1} + \mathbf{e_z}\right), \nabla v_h > \\ &= \tau < H_1, v_h > \end{aligned} \tag{29}$$

hold true for all $v_h \in V_h$. Then, with $\Psi_h^{n,j+1}$ obtained above, find $c_h^{n,j+1} \in V_h$ such that for all $w_h \in V_h$, one has

$$\begin{aligned} &< \theta\left(\Psi_h^{n,j+1}, c_h^{n,j}\right) c_h^{n,j+1} - \theta\left(\Psi_h^{n-1}, c_h^{n-1}\right) c_h^{n-1}, w_h > \\ &\quad +L_2 < c_h^{n,j+1} - c_h^{n,j}, w_h > \\ &\quad +\tau < D\nabla c + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla w_h > = \tau < H_2, w_h > . \end{aligned} \tag{30}$$

Remark 2 (Stopping criterion) For all schemes (monolithic or splitting), the iteration is stopped if for some small numbers $\epsilon_1, \epsilon_2 > 0$ one has

$$\left\| \Psi_h^{n,j+1} - \Psi_h^{n,j} \right\| \leq \epsilon_1, \text{ and } \left\| c_h^{n,j+1} - c_h^{n,j} \right\| \leq \epsilon_2,$$

later in the numerical section we will consider $\epsilon_1 = \epsilon_2$.

3 Convergence analysis

In this section, we analyze the convergence of the monolithic L-scheme introduced through Problem PMon-LS$_{n,j+1}$. We restrict the analysis to this iteration, but mention that the convergence analysis for the other (monolithic and splitting) schemes introduced above can be done in a similar fashion. We start by defining the errors

$$e_\Psi^{j+1} := \Psi_h^{n,j+1} - \Psi_h^{n,j} \text{ and } e_c^{j+1} := c_h^{n,j+1} - c_h^{n,j}, \quad (31)$$

obtained at iteration $j+1$. The scheme is convergent if both errors vanish when $j \to \infty$.

The convergence is obtained under the following assumptions:

(A1) There exist $\alpha_\Psi > 0$ and $\alpha_c \geq 0$ such that for any $\Psi_1, \Psi_2 \in \mathbb{R}$ and $c_1, c_2 \in \mathbb{R}_+$

$$\begin{aligned} &< \theta(\Psi_1, c_1) - \theta(\Psi_2, c_2), \Psi_1 - \Psi_2 > \\ &\quad + < c_1\theta(\Psi_1, c_1) - c_2\theta(\Psi_2, c_2), c_1 - c_2 > \\ &\geq \alpha_\Psi \left\| \theta(\Psi_1, c_1) - \theta(\Psi_2, c_2) \right\|^2 + \alpha_c \left\| \Psi_1 - \Psi_2 \right\|^2. \end{aligned} \quad (32)$$

Furthermore, there exist two constants $\theta_m \geq 0$ and $\theta_M < \infty$ such that $\theta_m \leq \theta(\Psi, c) \leq \theta_M, \ \forall\, \Psi, c \in \mathbb{R}$

(A2) The function $K(\theta(\cdot,\cdot))$ is Lipschitz continuous, with respect to both variables, and there exist two constants K_m and K_M such that $0 \leq K_m \leq K \leq K_M < \infty$.

(A3) There exist also $M_u, M_\Psi, M_c \geq 0$ such that $\left\| \mathbf{u_w^n} \right\|_{L^\infty} \leq M_u$, $\left\| \nabla \Psi^n \right\|_{L^\infty} \leq M_\Psi$ and $\left\| c^n \right\|_{L^\infty} \leq M_c$ for all $n \in \mathbb{N}$.

Remark 3 (A2) is satisfied in most realistic situations. (A3) is a pure technical requirement, being satisfied when data is sufficiently regular, which is assumed to be the case for the present analysis. The inequality (32) in (A1) is a coercivity assumption. It is in particular satisfied if θ only depends on Ψ, and for common relationships $\theta - -\Psi$ encountered in the engineering literature.

Theorem 1 *Let $n \in \{1, 2, \dots N\}$ be given and assume* (A1)–(A3) *be satisfied. If the time step is small enough* (*see* 42 *below*), *the monolithic L-scheme in* (29)–(30) *is linearly convergent for any L_1 and L_2 satisfying* (41).

Proof We follow the ideas in [30, 34] and start by subtracting (11) from (29) to obtain the error equation

$$\begin{aligned} &< \theta_h^{n,j} - \theta_h^n, v_h > + L_1 < \Psi_h^{n,j+1} - \Psi_h^{n,j}, v_h > \\ &\quad + \tau < K_h^{n,j} \nabla e_\Psi^{n,j+1}, \nabla v_h > \\ &\quad + \tau < (K_h^{n,j} - K_h^n) \nabla \Psi_h^{n,j+1}, \nabla v_h > \\ &\quad + \tau < (K_h^{n,j} - K_h^n) \mathbf{e_z}, \nabla v_h > = 0, \end{aligned} \quad (33)$$

where $\theta_h^{n,j} := \theta(\Psi_h^{n,j}, c_h^{n,j})$, $\theta_h^{n-1} := \theta(\Psi_h^n, c_h^n)$, $K_h^{n,j} := K_h(\theta_h^{n,j})$ and $K_h^n := K_h(\theta_h^n)$. Testing now the above equation with $v_h = e_\Psi^{j+1}$, one obtains

$$\begin{aligned} &< \theta_h^{n,j} - \theta_h^n, e_\Psi^{j+1} > + L_1 \\ &\quad + \tau < K^{n,j} \nabla e_\Psi^{n,j+1}, \nabla e_\Psi^{j+1} > \\ &\quad + \tau < (K_h^{n,j} - K_h^n) \nabla \Psi_h^{n,j+1}, \nabla e_\Psi^{j+1} > \\ &\quad + \tau < (K_h^{n,j} - K_h^n) \mathbf{e_z}, \nabla e_\Psi^{j+1} > = 0. \end{aligned} \quad (34)$$

By (A2) and after some algebraic manipulations, we further get

$$\begin{aligned} &< \theta_h^{n,j} - \theta_h^n, e_\Psi^j > + \frac{L_1}{2} \left\| e_\Psi^{j+1} \right\|^2 + \frac{L_1}{2} \left\| e_\Psi^{j+1} - e_\Psi^j \right\|^2 \\ &\quad + \tau K_m \left\| \nabla e_\Psi^{j+1} \right\|^2 \\ &\leq \frac{L_1}{2} \left\| e_\Psi^j \right\|^2 - < \theta_h^{n,j} - \theta_h^n, e_\Psi^{j+1} - e_\Psi^j > \\ &\quad - \tau < (K_h^{n,j} - K_h^n) \nabla \Psi_h^{n,j+1}, \nabla e_\Psi^{j+1} \\ &> - \tau < (K_h^{n,j} - K_h^n) \mathbf{e_z}, \nabla e_\Psi^{j+1} > . \end{aligned} \quad (35)$$

Using now (A1), (A3), the Lipschitz continuity of K, and twice the Young and Cauchy-Schwarz inequalities, for any $\delta_0 > 0$ and $\delta_1 > 0$, from (35), one obtains

$$\begin{aligned} &< \theta_h^{n,j} - \theta_h^n, e_\Psi^j > + \frac{L_1}{2} \left\| e_\Psi^{j+1} \right\|^2 \\ &+ \frac{L_1}{2} \left\| e_\Psi^{j+1} - e_\Psi^j \right\|^2 + \tau K_m \left\| \nabla e_\Psi^{j+1} \right\|^2 \\ &\leq \frac{L_1}{2} \left\| e_\Psi^j \right\|^2 + \frac{\delta_0}{2} \left\| \theta_h^{n,j} - \theta_h^n \right\|^2 + \frac{1}{2\delta_0} \left\| e_\Psi^{j+1} - e_\Psi^j \right\|^2 \\ &\quad + \frac{\tau(M_\Psi^2 + 1) L_k^2}{2\delta_1} \left\| \theta_h^{n,j} - \theta_h^n \right\|^2 + \tau \delta_1 \left\| \nabla e_\Psi^{j+1} \right\|^2. \end{aligned} \quad (36)$$

Similarly, subtracting (12) from (30) and choosing $w_h = e_c^{j+1}$ in the results, one gets

$$\begin{aligned} &< c_h^{n,j+1} \theta_h^{n,j} - c_h^n \theta_h^n, e_c^{j+1} > + L_2 < e_c^{j+1} - e_c^j, e_c^{j+1} > \\ &\quad + \tau < D \nabla e_c^{j+1} + \mathbf{u_w^{n-1}} e_c^{j+1}, \nabla e_c^{j+1} > = 0. \end{aligned} \quad (37)$$

This can be rewritten as

$$
\begin{aligned}
&< c_h^{n,j}\theta_h^{n,j} - c^n\theta_h^n, e_c^j > + < \theta_h^{n,j} e_c^{j+1}, e_c^{j+1} > \\
&+\frac{L_2}{2}\left\|e_c^{j+1}\right\|^2 + \frac{L_2}{2}\left\|e_c^{j+1} - e_c^j\right\|^2 \\
&+\tau D < \nabla e_c^{j+1}, \nabla e_c^{j+1} > \\
&= \frac{L_2}{2}\left\|e_c^j\right\|^2 + < \theta_h^n c_h^n - \theta_h^{n,j} c_h^{n,j}, e_c^{j+1} - e_c^j > -\tau \\
&< \mathbf{u_w^{n-1}} e_c^{j+1}, \nabla e_c^{j+1} > . \qquad (38)
\end{aligned}
$$

Using again (A1), (A3), and the Cauchy-Schwarz and Young inequalities, from (38), it follows that for any $\delta_2, \delta_3, \delta_4 > 0$, one has

$$
\begin{aligned}
&< c_h^{n,j}\theta_h^{n,j} - c_h^n\theta_h^n, e_c^j > + \theta_m \left\|e_c^{j+1}\right\|^2 + \frac{L_2}{2}\left\|e_c^{j+1}\right\|^2 \\
&+\frac{L_2}{2}\left\|e_c^{j+1} - e_c^j\right\|^2 + \tau D \left\|\nabla e_c^{j+1}\right\|^2 \\
&\le \frac{L_2}{2}\left\|e_c^j\right\|^2 + \frac{\delta_2}{2}\left\|\theta_h^n - \theta_h^{n,j}\right\|^2 + \frac{\delta_3}{2}\left\|e_c^j\right\|^2 \\
&+\left(\frac{M_c^2}{2\delta_2} + \frac{\theta_M^2}{2\delta_3}\right)\left\|e_c^{j+1} - e_c^j\right\|^2 + \tau\frac{M_u^2}{2\delta_4}\left\|e_c^{j+1}\right\|^2 \\
&+\tau\frac{\delta_4}{2}\left\|\nabla e_c^{j+1}\right\|^2 . \qquad (39)
\end{aligned}
$$

Adding (36) to (39) and using (A1), one gets

$$
\begin{aligned}
&\alpha_\Psi \left\|\theta_h^n - \theta_h^{n,j}\right\|^2 + \frac{L_1}{2}\left\|e_\Psi^{j+1}\right\|^2 \\
&+\frac{L_1}{2}\left\|e_\Psi^{j+1} - e_\Psi^j\right\|^2 + \tau K_m \left\|\nabla e_\Psi^{j+1}\right\|^2 \\
&+\alpha_c \left\|e_c^j\right\|^2 + \theta_m \left\|e_c^{j+1}\right\|^2 + \frac{L_2}{2}\left\|e_c^{j+1}\right\|^2 \\
&+\frac{L_2}{2}\left\|e_c^{j+1} - e_c^j\right\|^2 + \tau D \left\|\nabla e_c^{j+1}\right\|^2 \\
&\le \frac{L_1}{2}\left\|e_\Psi^j\right\|^2 + \left(\frac{\delta_0}{2} + \frac{\tau(M_\Psi^2+1)L_k^2}{2\delta_1} + \frac{\delta_2}{2}\right)\left\|\theta_h^{n,j} - \theta_h^n\right\|^2 \\
&+\frac{1}{2\delta_0}\left\|e_\Psi^{j+1} - e_\Psi^j\right\|^2 + \tau\delta_1 \left\|\nabla e_\Psi^{j+1}\right\|^2 + \frac{L_2}{2}\left\|e_c^j\right\|^2 \\
&+\frac{\delta_3}{2}\left\|e_c^j\right\|^2 + \left(\frac{M_c^2}{2\delta_2} + \frac{\theta_M^2}{2\delta_3}\right)\left\|e_c^{j+1} - e_c^j\right\|^2 \\
&+\tau\frac{M_u^2}{2\delta_4}\left\|e_c^{j+1}\right\|^2 + \tau\frac{\delta_4}{2}\left\|\nabla e_c^{j+1}\right\|^2 . \qquad (40)
\end{aligned}
$$

Choosing $\delta_0 = \delta_2 = \frac{\alpha_\Psi}{2}$, $\delta_1 = \frac{K_m}{2}$, $\delta_3 = \theta_m$ and $\delta_4 = \frac{D}{2}$ in (40), and assuming that

$$
L_1 \ge \frac{2}{\alpha_\Psi} \quad \text{and} \quad L_2 \ge \frac{2M_c^2}{\alpha_\Psi} + \frac{\theta_M^2}{\theta_m}, \qquad (41)
$$

and the time step τ satisfies the mild conditions

$$
\begin{aligned}
&\alpha_\Psi - 2\tau\frac{\tau(M_\Psi^2+1)L_k^2}{K_m} \ge 0 \quad \text{and} \\
&\theta_m + 2\alpha_c + \frac{\tau D}{C_\Omega} - \frac{2\tau M_u^2}{D} \ge 0, \qquad (42)
\end{aligned}
$$

where C_Ω denotes the Poincare constant; then, we obtain

$$
\begin{aligned}
&\frac{L_1}{2}\left\|e_\Psi^{j+1}\right\|^2 + \tau\frac{K_m}{2}\left\|\nabla e_\Psi^{j+1}\right\|^2 + \left(\frac{L_2}{2} + \theta_m - \tau\frac{M_u^2}{D}\right) \\
&\times \left\|e_c^{j+1}\right\|^2 + \tau\frac{D}{2}\left\|\nabla e_c^{j+1}\right\|^2 \\
&\le \frac{L_1}{2}\left\|e_\Psi^j\right\|^2 + \left(\frac{L_2}{2} + \frac{\theta_m}{2} - \alpha_c\right)\left\|e_c^j\right\|^2 . \qquad (43)
\end{aligned}
$$

Finally, by using the Poincare inequality two times we get from (43)

$$
\begin{aligned}
&\left(L_1 + \tau\frac{K_m}{C_\Omega}\right)\left\|e_\Psi^{j+1}\right\|^2 + \left(L_2 + 2\theta_m + \tau\frac{D}{C_\Omega} - 2\tau\frac{M_u^2}{D}\right) \\
&\times \left\|e_c^{j+1}\right\|^2 \\
&\le L_1\left\|e_\Psi^j\right\|^2 + (L_2 + \theta_m - 2\alpha_c)\left\|e_c^j\right\|^2 . \qquad (44)
\end{aligned}
$$

From (42), (44) implies that the errors are contracting and therefore the monolithic L-scheme (29)–(30) is convergent. □

Remark 4 The convergence rate resulting from (44) does not depend on the spatial mesh size. Also, observe that this convergence is obtained for any initial guess. Based on this, the method is globally convergent, which is in contrast to the Newton and (modified) Picard schemes, converging only locally. It can be observed that, larger the time step and smaller constants L_1 and L_2, result in a faster convergence. For small steps instead the convergence rate can approach 1. On the other hand, if the time step is small enough, one may reach the regime where the Newton scheme becomes convergent (see [37]). Alternatively, one may first perform a number of L-scheme iterations, and use the resulting as an initial guess for the Newton scheme (see [30]), or consider the modified L-scheme in [31]. In either situations, the convergence behavior was much improved.

Remark 5 The convergence of the modified Picard and Newton method applied to the Richards equation has been already proved in [37]. Such results can be extended to the coupled problems considered here.

Table 1 Parameters involved in all examples

L_1	0.1
L_2	0.1
D	1e−3
Van Genuchten parameters	
θ_s	0.42
θ_r	0.026
n	2.9
α	0.551
a	0.44
b	0.0046
K_s	0.12
Accuracy requirement	
$\epsilon_1 = \epsilon_2$	10^{-06}

4 Numerical examples

In this section we consider five test cases for the proposed linearization schemes, inspired by the literature [25, 30]. The schemes have been implemented in the open source software package MRST [29], an open source toolbox based on Matlab, in which multiple solvers and models regarding flows in porous media are incorporated.

Example 1A: Flow and transport in a strictly unsaturated medium

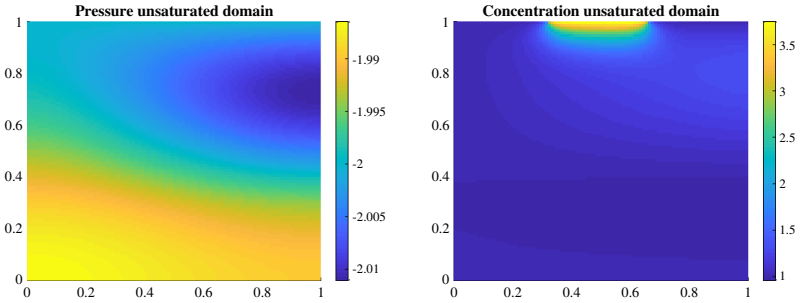
For the first example, a van Genuchten parametrization is considered, with the parameters given in Table 1. The domain Ω is the unit square. To define the initial pressure, Ω is divided into two sub-regions: $\Omega_{up} = (0, 1) \times [1/4, 1)$ and $\Omega_{down} = (0, 1) \times (0, 1/4)$. In Ω_{up}, the source term, in the Richards equation, is $H_1(x, y) = 0.06 \cos(4/3\pi y) \sin(x)$ and $H_2(x, y) = 0$, in the transport. The lower sub-domain Ω_{down} contains no external sources, i.e., $H_1 = H_2 = 0$.

We impose $\Psi = -2$ on $\Gamma_D = [0, 1] \times 1$, $c = 1$ on $\Gamma_D \setminus \Gamma_C$, where $\Gamma_C = [1/3, 2/3] \times 1$, $c = 4$ on Γ_C and no-flow Neumann boundary conditions for both model components on $\Gamma_N = \partial\Omega \setminus \Gamma_D$. At $t = 0$ the initial pressure in the two sub-domains is $\Psi^0_{up} = -2$ and $\Psi^0_{down} = -y - 1/4$, respectively. The initial concentration is $c^0 = 1$.

The simulations are performed on regular meshes, consisting of squares with sides $dx = 1/10,\ 1/20$, and $1/40$. The time steps are $\tau = 1/10,\ 1/20$, and $1/40$. Figure 3 displays the pressure and concentration profiles at the final time $T = 1$. Note that in this example the flow is always partially saturated, implying that the Richards equation does not degenerate. Furthermore, the flow is completely dominated by the source term H_1 while, for the transport, we can notice both diffusion and advection effects.

The total number of iterations and the condition numbers of the linear systems associated to each solving algorithms are presented in Tables 2 and 3. The condition numbers are computed using the L^1 norm and we report here the averaged values over the full simulation. A segment (−) in these tables implies that the method failed to converge for the particular combination of time step size and mesh. Table 2 gives the results obtained for the time step $\tau = 1/10$ and for different mesh sizes, $dx = 1/10, 1/20$ and $1/40$. In this case we can observe that the L-scheme based solvers converges for each mesh. The splitting solvers obtained instead thanks to the Newton and modified Picard linearizations, fail to converge in case of finer meshes. Furthermore, the numbers of iterations for the L-schemes is mesh independent, which agrees with the theory.

Similarly, Table 3 provides the results for a constant $dx = 1/40$, and for $\tau = 1/10,\ 1/20$, and $1/40$. Observe that for the smaller time steps all schemes converge, whereas for $\tau = 1/10$, the Newton and Picard-based variants of the splitting schemes diverge. This is in line with the results reported in Table 2, where a finer spatial mesh has led to the divergence of these schemes. Since the number of iterations is added per each time step, this number is



(a) Pressure profile at T = 1 (b) Concentration profile at T = 1

Fig. 3 Example 1A: pressure and concentration at the final time, $T = 1$. The simulations were performed for $dx = 1/100$ and $\tau = 1/10$. **a** Pressure profile at $T = 1$. **b** Concentration profile at $T = 1$

Table 2 Example 1A: Iterations and condition numbers for fixed $\tau = 1/10$

	Monolithic		NonLinS			AltLinS		
	Newton		Newton			Newton		
				cond. no.				cond. no.
dx	No. of iterations	Condition No.	No. of iterations	Richards	Transport iterations	No. of	Richards	Transport
1/10	40	492.7535	24 - 44	153.3064	2.6810	44	159.4131	2.6760
1/20	64	2.3911e+03	26 - 94	597.8236	5.9056	94	626.5425	5.8943
1/40	189	1.2294e+04	–	–	–	–	–	–
	L Scheme			L Scheme			L Scheme	
				cond. no.				cond. no.
dx	No. of iterations	Condition No.	No. of iterations	Richards	Transport	No. of iterations	Richards	Transport
1/10	124	349.9054	82 - 119	106.9132	1.9051	124	106.9216	1.8867
1/20	125	1.5698e+03	80 - 110	427.3669	3.5700	120	427.3894	3.5920
1/40	125	7.1229e+03	79 - 100	1.7114e+03	8.7182	120	1.7114e+03	8.9720
	Picard			Picard			Picard	
				cond. no.				cond. no.
dx	No. of iterations	Condition No.	No. of iterations	Richards	Transport	No. of iterations	Richards	Transport
1/10	43	667.9851	25 - 44	153.4789	2.6762	44	159.2404	2.6712
1/20	67	3.1574e+03	26 - 94	600.0477	5.8927	94	626.1294	5.8814
1/40	189	1.5969e+04	–	–	–	–	–	–

Table 3 Example 1A: Iterations and condition numbers for fixed $dx = 1/40$

	Monolithic		NonLinS			AltLinS		
	Newton		Newton			Newton		
				cond. no.				cond. no.
τ	No. of iterations	Condition No.	No. of iterations	Richards	Transport	No. of iterations	Richards	Transport
1/10	189	1.2294e+04	–	–	–	–	–	–
1/20	201	6.3754e+03	42 - 319	1.4157e+03	8.8062	320	1.4641e+03	8.8233
1/40	259	3.2828e+03	72 - 320	782.6252	4.9460	320	808.0468	4.9264
	L Scheme			L Scheme			L Scheme	
				cond. no.				cond. no.
τ	No. of iterations	Condition No.	No. of iterations	Richards	Transport	No. of iterations	Richards	Transport
1/10	125	7.1229e+03	79 - 100	1.7114e+03	8.7182	120	1.7114e+03	8.9720
1/20	244	3.7562e+03	162 - 209	932.0422	4.9942	235	932.0541	5.0758
1/40	471	1.9522e+03	331 - 419	480.1427	3.0668	463	480.1378	3.1478
	Picard			Picard			Picard	
				cond. no.				cond. no.
τ	No. of iterations	Condition No.	No. of iterations	Richards	Transport	No. of iterations	Richards	Transport
1/10	189	1.5969e+04	–	–	–	–	–	–
1/20	201	8.4667e+03	44–316	1.4103e+03	8.7983	316	1.4650e+03	8.7777
1/40	258	4.4183e+03	72–318	782.3317	4.9358	320	808.3800	4.9126

increasing as the time step is reduced. This is justified by the fact that smaller τ implies more time steps.

We point out that the alternate splitting schemes are converging much faster than the classical ones, for which we report the iterations required by the flow and the transport equations, separately. Note the differences in the condition numbers, the L-scheme-based algorithms being better conditioned. Observe also that for the splitting schemes, the condition numbers for the Richards equation are much larger than for the transport model component. This is due to the fact that the former is nonlinear and possibly degenerate, whereas the latter has a fairly simple structure. Finally, one can observe that finer meshes results in higher condition numbers while smaller time steps give better conditioned systems.

Example 1B: Flow and transport in a variably saturated porous medium

The situation given above is changed slightly, so that the fully saturated regime is achieved. Specially, we take $\Psi^0_{up} = -2$ and $\Psi^0_{down} = -y + 1/4$. By this, Ψ^0_{down} becomes positive in Ω_{down}, where the medium is fully saturated. Consequently, the Richards equation degenerates to an elliptic one, making the numerical simulation more challenging. The L parameters are $L_1 = L_2 = 0.2$.

Tables 4 and 5 present the iterations and condition numbers for each of the implemented algorithms, and for different mesh diameters and time steps. Note that in this case, only the L-scheme-based algorithms are converging. It is also interesting to observe the difference in the number of iterations between the more commonly used nonlinear splitting approach (NonLinS) and the alternate splitting (AltLinS) approach. The latter appears to be a valid alternative to the common formulation. It produces equally accurate results, requiring fewer iterations.

Finally, we observe as the Newton and Picard-based schemes fail to converge in all situations. This is due to the degeneracy of the Richards equation. The L-scheme-based iterations did converge in all cases. The convergence behavior is as predicted by the theory: the number of iterations increases for smaller time steps, while the mesh size has no influence on the number of iterations.

Example 2A: Well in an unsaturated porous medium

The next example is inspired from [25]. We still consider the unit square-domain, the initial conditions and the parameters as in Example 1A. The medium results again strictly unsaturated. Now Ω_{up} includes a well, and water with a given surfactant concentration is injected. The pressure at the well is set to $\Psi_W = -1/4$ and the

Table 4 Example 1B: Iterations and condition numbers for fixed $\tau = 1/10$

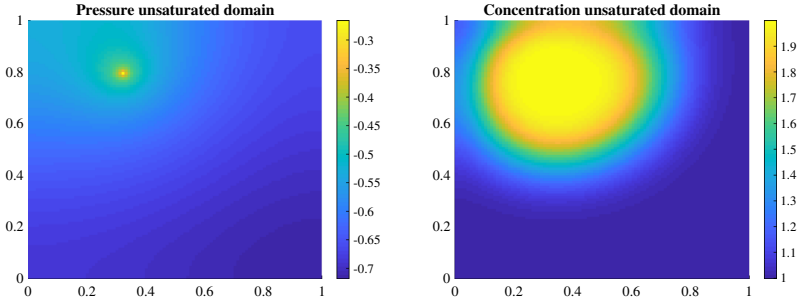
	Monolithic		NonLinS			AltLinS		
	Newton		Newton			Newton		
				cond. no.			cond. no.	
dx	No. of iterations	Condition No.	No. of iterations	Richards	Transport	No. of iterations	Richards	Transport
1/10	–	–	–	–	–	–	–	–
1/20	–	–	–	–	–	–	–	–
1/40	–	–	–	–	–	–	–	–
	L Scheme		L Scheme			L Scheme		
				cond. no.			cond. no.	
dx	No. of iterations	Condition No.	No. of iterations	Richards	Transport	No. of iterations	Richards	Transport
1/10	228	204.8977	180 - 206	58.9991	1.8625	242	58.9938	1.9066
1/20	226	879.2575	175 - 182	233.6193	3.3333	236	233.5856	3.6303
1/40	228	4.0163e+03	175 - 150	932.4226	7.1687	230	932.3206	8.2482
	Picard		Picard			Picard		
				cond. no.			cond. no.	
dx	No. of iterations	Condition No.	No. of iterations	Richards	Transport	No. of iterations	Richards	Transport
1/10	–	–	–	–	–	–	–	–
1/20	–	–	–	–	–	–	–	–
1/40	–	–	–	–	–	–	–	–

Table 5 Example 1B: Iterations and condition numbers for $dx = 1/40$

	Monolithic		NonLinS				AltLinS			
	Newton		Newton				Newton			
					cond. no.				cond. no.	
τ	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	–	–	–	–		–	–	–		–
1/20	–	–	–	–		–	–	–		–
1/40	–	–	–	–		–	–	–		–
	L Scheme			L Scheme				L Scheme		
					cond. no.				cond. no.	
τ	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	228	4.0163e+03	175 - 150	932.4226		7.1687	230	932.3206		8.2482
1/20	457	2.1063e+03	362 - 326	480.1123		4.2676	481	480.1165		4.5628
1/40	877	1.0751e+03	724 - 668	241.9597		2.7304	944	241.9634		2.7427
	Picard			Picard				Picard		
					cond. no.				cond. no.	
τ	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	–	–	–	–		–	–	–		–
1/20	–	–	–	–		–	–	–		–
1/40	–	–	–	–		–	–	–		–

concentration of the surfactant to $c_W = 2$. We impose no-flow boundary conditions for both model components on $\partial\Omega$. The simulations are performed on regular meshes, consisting of squares with sides $dx = 1/10,\ 1/20$, and $1/40$. The time steps are $\tau = 1/25,\ 1/50$, and $1/100$. Furthermore, a reaction term is included in the transport equation, $R(c) := 1e-3*c/(1+c)$. For the iteration $j+1$, this is linearized as $R(c^{n+1,j+1}) \rightarrow 1e-3\frac{c^{n+1,j+1}}{1+c^{n+1,j}}$. The L parameters are $L_1 = L_2 = 0.2$.

Figure 4 shows the pressure and the concentration at the final time step ($T = 1$). As for the first example, the medium being partially saturated, the Richards equation does not degenerate and almost all the schemes converge. The monolithic Newton method requires smaller time steps, as observable in Table 7. We remark, from both Tables 6 and 7, that the alternate splitting approach (AltLinS), once more, requires fewer iterations than the classical splitting algorithm (NonLinS). The linear systems resulting by



(a) Pressure at final time T_1 (b) Concentration at final time T_1

Fig. 4 Example 2A: pressure and concentration at the first time steps and final times. The simulations were performed for $dx = 1/80$ and $\tau = 1/10$. **a** Pressure at final time T_1. **b** Concentration at final time T_1

Table 6 Example 2A: Iterations and condition numbers for fixed $\tau = 1/25$

	Monolithic		NonLinS				AltLinS			
	Newton		Newton				Newton			
				cond. no.					cond. no.	
dx	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	59	1.1597e+03	57 - 32	3.5250e+03		1.6216e+05	–	–		–
1/20	–	–	57 - 35	3.5250e+03		1.6216e+05	–	–		–
1/40	–	–	57 - 33	3.9905e+04		2.4845e+06	–	–		–
	L Scheme			L Scheme				L Scheme		
				cond. no.					cond. no.	
dx	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	368	9.4574e+04	209 - 315	395.8757		7.8808e+03	338	397.4869		7.9035e+03
1/20	364	7.6020e+03	222 - 327	2.8139e+03		9.0509e+04	346	2.8220e+03		9.0971e+04
1/40	368	9.4574e+04	223 - 332	3.0786e+04		1.3442e+06	348	3.0861e+04		1.3515e+06
	Picard			Picard				Picard		
				cond. no.					cond. no.	
τ	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	106	2.1274e+03	61 - 50	426.0623		1.2673e+04	74	416.8809		1.2985e+04
1/20	105	2.1056e+04	70 - 50	3.4855e+03		1.5665e+05	84	3.3929e+03		1.6001e+05
1/40	105	2.7185e+05	84 - 50	3.9872e+04		2.3954e+06	86	3.9462e+04		2.4544e+06

Table 7 Example 2A: Iterations and condition numbers for $dx = 1/20$

	Monolithic		NonLinS				AltLinS			
	Newton		Newton				Newton			
				cond. no.					cond. no.	
τ	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/25	–	–	57 - 35	3.5250e+03		1.6216e+05	–	–		–
1/50	109	1.5493e+03	106 - 56	1.1707e+03		4.3558e+04	–	–		–
1/100	207	795.5026	207 - 105	433.1146		1.2761e+04	–	–		–
	L Scheme			L Scheme				L Scheme		
				cond. no.					cond. no.	
τ	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/25	364	7.6020e+03	222 - 327	2.8139e+03		9.0509e+04	346	2.8220e+03		9.0971e+04
1/50	685	2.3560e+03	436 - 610	994.2747		2.5259e+04	656	996.9713		2.5397e+04
1/100	1284	792.0927	836 - 1133	406.9732		7.8582e+03	1220	408.0727		7.8911e+03
	Picard			Picard				Picard		
				cond. no.					cond. no.	
τ	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/25	105	2.1056e+04	70 - 50	3.4855e+03		1.5665e+05	84	3.3929e+03		1.6001e+05
1/50	206	6.4247e+03	135 - 100	1.1871e+03		4.2339e+04	146	1.1586e+03		4.3126e+04
1/100	406	2.1116e+03	238 - 168	443.2937		1.2691e+04	254	435.4151		1.2755e+04

Table 8 Example 2B: Iterations and condition numbers for $\tau = 1/25$

	Monolithic		NonLinS				AltLinS			
	Newton		Newton				Newton			
					cond. no.				cond. no.	
dx	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	–	–	–	–		–	–	–		–
1/20	–	–	–	–		–	–	–		–
1/40	–	–	–	–		–	–	–		–
	L Scheme			L Scheme				L Scheme		
					cond. no.				cond. no.	
dx	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	681	446.0156	271 - 659	275.7570		8.0058e+03	680	275.5587		8.0092e+03
1/20	676	1.5730e+03	270 - 686	740.6757		8.5552e+04	698	739.6510		8.5630e+04
1/40	681	6.7119e+03	256 - 679	2.7866e+03		1.3095e+06	694	2.7602e+03		1.3121e+06
	Picard			Picard				Picard		
					cond. no.				cond. no.	
dx	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	–	–	–	–		–	–	–		–
1/20	–	–	–	–		–	–	–		–
1/40	–	–	–	–		–	–	–		–

Table 9 Example 2B: Iterations and condition numbers for $dx = 1/20$

	Monolithic		NonLinS				AltLinS			
	Newton		Newton				Newton			
					cond. no.				cond. no.	
τ	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/25	–	–	–	–		–	–	–		–
1/50	–	–	–	–		–	–	–		–
1/100	–	–	–	–		–	–	–		–
	L Scheme			L Scheme				L Scheme		
					cond. no.				cond. no.	
τ	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/25	676	1.5730e+03	270 - 686	740.6757		8.5552e+04	698	739.6510		8.5630e+04
1/50	1253	812.6783	514 - 1285	431.5091		2.4385e+04	1298	430.5647		2.4399e+04
1/100	2350	448.3554	988 - 2388	275.5439		8.0075e+03	2938	275.4834		8.0092e+03
	Picard			Picard				Picard		
					cond. no.				cond. no.	
τ	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/25	–	–	–	–		–	–	–		–
1/50	–	–	–	–		–	–	–		–
1/100	–	–	–	–		–	–	–		–

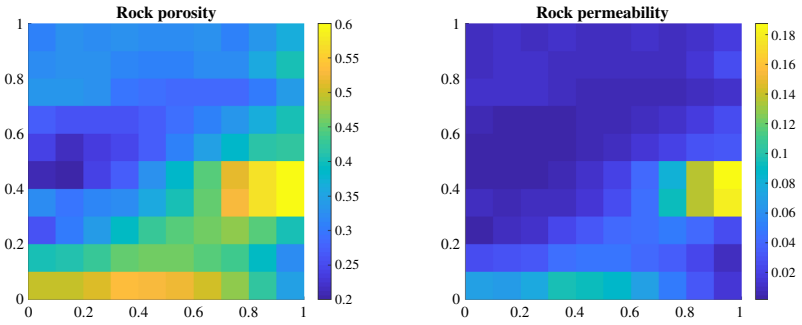


Fig. 5 Example 3: A highly heterogeneous domain. a Porosity of the domain Ω. b Permeability of the domain Ω

(a) Porosity of the domain Ω

(b) Permeability of the domain Ω

applying the L-scheme-based solvers are better conditioned compared to the other solvers. Finally, we can observe as the introduction of a nonlinear reaction term has drastically increased the condition numbers of the system associated to the transport equation.

Example 2B: Well in a variably saturated porous medium

As in Example 1B, now the initial condition for the pressure is changed, leading to a variably saturated porous media. The pressure at the well is fixed equal to $1/4$ and the concentration and boundary conditions are defined as in the Example 2A. The L parameters are now $L_1 = 0.2$ and $L_2 = 0.5$.

As for Example 1B, since the Richards equation degenerates, many of the considered schemes show convergence problems. Tables 8 and 9 present the convergence of the schemes and the condition numbers for the associated linear systems. The results are very similar to those in the previous examples, with the L-scheme-based solvers being the most robust ones and the only converging for all cases. Furthermore, the alternate method is faster than the classical splitting scheme.

Example 3: A heterogeneous porous medium

In more realistic situations, the porous medium is often heterogeneous. In this example we consider again the unit-square domain but with highly heterogeneous properties (porosity and permeability), as presented in Fig. 5. Next to this, the problem is similar to the one in Example 2B, including the same initial conditions and parameters. The well is now located in the lower right part of the domain where we observe larger porosity and permeability. Due to the initial pressure Ψ^0, the domain results to be variably saturated and the problem degenerates. The L parameters are $L_1 = L_2 = 0.7$.

Figure 6 present the pressure and concentration at the final time step. We can observe as, particularly the former, has increased in the regions with higher permeability. Similarly we can observe how the structure of the media has influence the diffusion of the external component.

Tables 10 and 11 present the total numbers of iterations for each algorithm and the condition numbers of the associated linearized systems. For this particular problem, it is interesting to notice that the L-scheme converges, again, for every time step and mesh investigated. In this

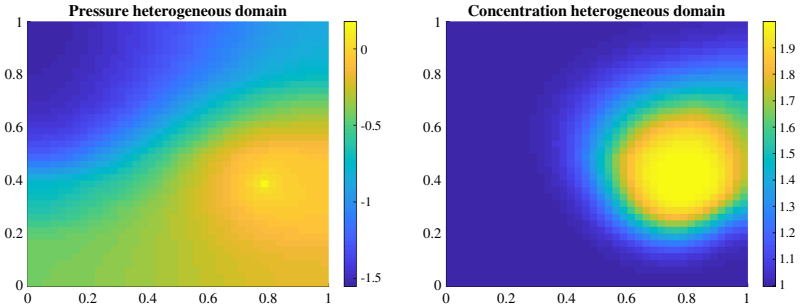


Fig. 6 Example 3: Pressure and concentration at the final time. The simulations are performed for $dx = 1/40$ and $\tau = dx/10$. **a** Pressure at the final time T. b Concentration at the final time T

(a) Pressure at the final time T

(b) Concentration at the final time T

Table 10 Example 3: Iterations and condition numbers for $\tau = 1/10$

	Monolithic		NonLinS				AltLinS			
	Newton		Newton				Newton			
					cond. no.				cond. no.	
dx	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	–	–	–	–		–	–	–		–
1/20	–	–	–	–		–	–	–		–
1/40	–	–	–	–		–	–	–		–
	L Scheme			L Scheme				L Scheme		
					cond. no.				cond. no.	
dx	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	777	326.5548	433 - 208	71.9066		257.0406	938	71.9110		256.4755
1/20	775	3.9276e+03	433 - 212	678.4042		3.0834e+03	940	678.4266		3.0742e+03
1/40	775	5.1305e+04	433 - 215	8.2423e+03		4.5823e+04	940	8.2423e+03		4.5697e+04
	Picard			Picard				Picard		
					cond. no.				cond. no.	
dx	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	–	–	–	–		–	–	–		–
1/20	–	–	–	–		–	–	–		–
1/40	–	–	–	–		–	–	–		–

Table 11 Example 3: iterations and condition numbers for $dx = 1/10$

	Monolithic		NonLinS				AltLinS			
	Newton		Newton				Newton			
					cond. no.				cond. no.	
τ	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	–	–	–	–		–	–	–		–
1/20	–	–	–	–		–	–	–		–
1/40	–	–	–	–		–	–	–		–
	L Scheme			L Scheme				L Scheme		
					cond. no.				cond. no.	
τ	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	777	326.5548	433 - 208	71.9066		257.0406	938	71.9110		256.4755
1/20	1478	119.8774	751 - 362	32.6411		89.7378	1792	32.6459		89.6573
1/40	2861	50.2801	1241 - 611	18.0060		38.7625	3342	18.0085		38.7304
	Picard			Picard				Picard		
					cond. no.				cond. no.	
τ	No. of iterations	Condition No.	No. of iterations	Richards		Transport	No. of iterations	Richards		Transport
1/10	–	–	–	–		–	–	–		–
1/20	–	–	–	–		–	–	–		–
1/40	–	–	–	–		–	–	–		–

case, neither the Newton method nor the modified Picard, converged, even when smaller time steps were investigated. Smaller values were not tested because the resulting number of iterations would have been much larger than the one obtained with the original τ and the L-scheme. These results are coherent with the ones previously investigated. Whenever the Richards equation degenerate, both Newton and modified Picard present convergence problems.

5 Conclusions

In this paper, we studied different algorithms for the numerical solution of a surfactant transport model in variably saturated porous media. The water flow and the transport are fully coupled. Three linearization techniques were considered: the Newton method, the modified Picard and the L-scheme. Based on these, monolithic and splitting schemes were designed, analyzed and tested numerically. We conclude that the only quadratic convergent scheme is the monolithic Newton, that the L-scheme-based solvers are the most robust ones and produce well-conditioned linear systems and that the alternative schemes are often faster than the classical splitting approaches.

Although acknowledging the existence of improved Newton solvers, having a more robust convergence behavior, the present study shows that the L-scheme is a viable alternative. It can be particularly useful in the degenerate cases, or whenever large time steps have to be considered.

Acknowledgments Open Access funding provided by University of Bergen. We thank the members of the *Sintef* research group and in particular to Dr. Olav Moyner for the assistance with the implementation of the numerical examples in *MRST*, the toolbox based on MATLAB developed at *Sintef* itself.

Funding information The research of D. Illiano was funded by VISTA, a collaboration between the Norwegian Academy of Science and Letters and Equinor, project number 6367, project name: adaptive model and solver simulation of enhanced oil recovery. The research of I.S. Pop was supported by the Research Foundation-Flanders (FWO), Belgium, through the Odysseus programme (project G0G1316N) and Equinor through the Akademia grant.

References

1. Aavatsmark, I.: An introduction to multipoint flux approximations for quadrilateral grids. Comput. Geosci. **6**(3-4), 405–432 (2002)
2. Agosti, A., Formaggia, L., Scotti, A.: Analysis of a model for precipitation and dissolution coupled with a Darcy flux. J. Math. Anal. Appl. **431**(2), 752–781 (2015)
3. Alt, W., Luckhaus, H.: Quasilinear elliptic-parabolic differential equations. Math. Z. **183**(3), 311–341 (1983)
4. Arbogast, T., Wheeler, M.F.: A nonlinear mixed finite element method for a degenerate parabolic equation arising in flow in porous media. SIAM J. Numer. Anal. **33**(4), 1669–1687 (1996)
5. Barrett, J.W., Knabner, P.: Finite element approximation of the transport of reactive solutes in porous media. Part 1: error estimates for nonequilibrium adsorption processes. SIAM J. Numer. Anal. **34**(1), 201–227 (1997)
6. Bause, M., Hoffmann, J., Knabner, P.: First-order convergence of multi-point flux approximation on triangular grids and comparison with mixed finite element methods. Numer. Math. **116**(1), 1–29 (2010)
7. Berardi, M., Difonzo, F., Vurro, M., Lopez, L.: The 1D Richards' equation in two layered soils: a Filippov approach to treat discontinuities. Adv. Water Resour. **115**, 264–272 (2018)
8. Bergamaschi, L., Putti, M.: Mixed finite elements and Newton-type linearizations for the solution of Richards' equation. Int. J. Numer. Methods Eng. **45**(8), 1025–1046 (1999)
9. Cances, C., Pop, I.S., Vohralik, M.: An a posteriori error estimate for vertex-centered finite volume discretizations of immiscible incompressible two-phase flow. Math. Comput. **83**, 153–188 (2014)
10. Celia, M., Bouloutas, E., Zarba, R.L.: A General Mass-Conservative Numerical Solution for the Unsaturated Flow Equation. Adv. Water Resour. **26**(7), 1483–1496 (1990)
11. Christofi, N., Ivshina, I.B.: Microbial surfactants and their use in field studies of soil remediation. J. Appl. Microbiol. **93**(6), 915–929 (2002)
12. Dawson, C.: Analysis of an upwind-mixed finite element method for nonlinear contaminant transport equations. SIAM J. Numer. Anal. **35**(5), 1709–1724 (1998)
13. Eymard, R., Gutnic, M., Hilhorst, D.: The finite volume method for Richards equation. Comput. Geosci. **3**(3-4), 256–294 (1999)
14. Eymard, R., Hilhorst, D., Vohral, M.: A combined finite volume-nonconforming/mixed-hybrid finite element scheme for degenerate parabolic problems. Numer. Math. **105**(1), 73–131 (2006)
15. Farthing, M.W., Ogden, F.L.: Numerical solution of Richards' equation: A review of advances and challenges. Soil Sci. Soc. Am. J. **81**, 1257–1269 (2017)
16. Gallo, C., Manzini, G., mixed finite, A.: element/finite volume approach for solving biodegradation transport in groundwater. Internal Journal for Numerical Methods in Fluids **26**(5), 533–556 (1998)
17. van Genuchten, M.: A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Sci. Soc. Am. J. **44**(5), 892–898 (1980)
18. Helmig, R.: Multiphase flow and transport processes in the subsurface: A contribution to the modeling of hydrosystems. Springer-Verlag, Berlin (1997)
19. Henry, E.J., Smith, J.E., Warrick, A.W.: Solubility effects on surfactant-induced unsaturated flow through porous media. J Hydrol **223**(3-4), 164–174 (1999)
20. Husseini, D.: Effects of anions acids on surface tension of water, Undergraduate Research at JMU Scholarly Commons (2015)

21. Jones, J.E., Woodward, C.S.: Newton–Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems. Adv. Water Resour. **24**(7), 763–774 (2001)
22. Jenny, P., Tchelepi, H.A., Lee, S.H.: Unconditionally convergent nonlinear solver for hyperbolic conservation laws with S-shaped flux functions. J. Comput. Phys. **228**(20), 7497–7512 (2009)
23. Karagunduz, A., Young, M.H., Pennell, K.D.: Influence of surfactants on unsaturated water flow and solute transport. Water Resour. Res. **51**(4), 1977–1988 (2015)
24. Klausen, R.A., Radu, F.A., Eigestad, G.T.: Convergence of MPFA on triangulations and for Richards' equation. Int. J. Numer. Methods Fluids **58**(12), 1327–1351 (2008)
25. Knabner, P., Bitterlich, S., Teran, R.I., Prechtel, A., Schneid, E.: Influence of surfactants on spreading of contaminants and soil remediation. Springer, Berlin (2003)
26. Knoll, D.A., Keyes, D.E.: Jacobian-free Newton–Krylov methods: A survey of approaches and applications. J. Comput. Phys. **193**(2), 357–397 (2004)
27. Kumar, K., Pop, I.S., Radu, F.A.: Convergence analysis of mixed numerical schemes for reactive flow in a porous medium. SIAM J. Numer. Anal. **51**(4), 2283–2308 (2013)
28. Lee, S.H., Efendiev, Y.: C1-Continuous relative permeability and hybrid upwind discretization of three phase flow in porous media. Adv. Water Resour. **96**, 209–224 (2016)
29. Lie, K.-A.: An introduction to reservoir simulation using MATLAB: User guide for the Matlab reservoir simulation toolbox (MRST), SINTEF ICT (2016)
30. List, F., Radu, F.A.: A study on iterative methods for solving Richards' equation. Comput. Geosci. **20**(2), 341–353 (2016)
31. Mitra, K., Pop, I.S.: A modified L-Scheme to solve nonlinear diffusion problems. Comput. Math. Appl. **77**, 1722–1738 (2019)
32. Nochetto, R., Verdi, C.: Approximation of degenerate parabolic problems using numerical integration. SIAM J. Numer. Anal. **25**, 784–814 (1988)
33. Prechtel, A., Knabner, P.: Accurate and efficient simulation of coupled water flow and nonlinear reactive transport in the saturated and vadose zone - application to surfactant enhanced and intrinsic bioremediation. Int. J. Water Res. Dev. **47**, 687–694 (2002)
34. Pop, I.S., Radu, F.A., Knabner, P.: Mixed finite elements for the Richards' equation: linearization procedure. J. Comput. Appl. Math. **168**(1), 365–373 (2004)
35. Radu, F.A., Pop, I.S., Attinger, S.: Analysis of an Euler implicit, mixed finite element scheme for reactive solute transport in porous media. Num. Methods Part. Diff. Equ. **26**(2), 320–344 (2010)
36. Radu, F.A., Pop, I.S., Knabner, P.: Order of convergence estimates for an Euler implicit, mixed finite element discretization of Richards' equation. SIAM J. Numer. Anal. **42**(4), 1452–1478 (2004)
37. Radu, F.A., Pop, I.S., Knabner, P.: 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, pp. 1192–1200 (2006)
38. Radu, F.A., Suciu, N., Hoffmann, J., Vogel, A., Kolditz, O., Park, C.H., Attinger, S.: Accuracy of numerical simulations of contaminant transport in heterogeneous aquifers: a comparative study. Adv. Water Resour. **34**(1), 47–61 (2011)
39. Russell, T.F., Wheeler, M.F.: Finite element and finite difference methods for continuous flows in porous media, SIAM, pp. 35–106 (1983)
40. Slodicka, M.: A robust and efficient linearization scheme for doubly non-linear and degenerate parabolic problems arising in flow in porous media. SIAM J. Numer. Anal. **23**(5), 1593–1614 (2002)
41. Smith, J.E., Gillham, R.W.: 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 Resour. Res. **31**(3), 343–354 (1994)
42. Smith, J., Gillham, R.: Effects of solute concentration-dependent surface tension on unsaturated flow: Laboratory sand column experiments. Water Res. Res. **35**(4), 973–982 (1999)
43. Suciu, N.: Diffusion in random velocity fields with applications to contaminant transport in groundwater. Water Res. Res. **69**, 114–133 (2014)
44. Vohralik, M.: A posteriori error estimates for lowest-order mixed finite element discretizations of convection-diffusion-reaction equations. SIAM J. Numer. Anal. **45**(4), 1570–1599 (2007)
45. Walker, H.F., Ni, P.: Anderson acceleration for fixed-point iterations. SIAM J. Numer. Anal. **49**(4), 1715–1735 (2011)
46. Wang, X., Tchelepi, H.A.: Trust-region based solver for nonlinear transport in heterogeneous porous media. J. Comput. Phys. **253**, 114–137 (2013)
47. Woodward, C.S., Dawson, C.N.: Analysis of expanded mixed finite element methods for a nonlinear parabolic equation modeling flow into variably saturated porous media. SIAM J. Numer. Anal. **37**(3), 701–724 (2000)
48. Yong, W.A., Pop, I.S.: A numerical approach to porous medium equations, Preprint 95-50 (SFB 359), IWR University of Heidelberg (1996)
49. Younis, R., Tchelepi, H.A., Aziz, K.: Adaptively localized continuation-newton method–nonlinear solvers that converge all the time. SPE J. **15**(02), 526–544 (2010)

Paper B

An efficient numerical scheme for fully coupled flow and reactive transport in variably saturated porous media including dynamic capillary effects

Davide Illiano, Iuliu Sorin Pop, Florin Adrian Radu

An efficient numerical scheme for fully coupled flow and reactive transport in variably saturated porous media including dynamic capillary effects. Accepted manuscript https://arXiv:1912.06731

An efficient numerical scheme for fully coupled flow and reactive transport in variably saturated porous media including dynamic capillary effects*

Davide Illiano[1], Iuliu Sorin Pop[1,2], and Florin Adrian Radu[1]

[1]University of Bergen
[2]Hasselt University

Abstract

In this paper, we study a model for the transport of an external component, e.g., a surfactant, in variably saturated porous media. We discretize the model in time and space by combining a backward Euler method with the linear Galerkin finite elements. The Newton method and the L-Scheme are employed for the linearization and the performance of these schemes is studied numerically. A special focus is set on the effects of dynamic capillarity on the transport equation.

1 Introduction

In this work, we concentrate on efficiently solving reactive transport models in saturated/unsaturated porous media [8, 10]. Such media are observable in the section of the soil closer to the surface where, in the upper part of the domain, we have a coexistence of both water and air phases while, below the water table, the soil becomes fully saturated.

In particular, our model includes dynamic capillarity effects. The capillary pressure is commonly defined as the difference between the pressures of the two phases, in our case, the air and the water. Note that, in the Richards model, the air pressure is set to be equal to zero.

Typically, the capillary pressure is assumed to be a nonlinear decreasing function depending on the water saturation. However, numerous studies are showing that such formulation is often too simplistic and that dynamic effects, due to the changes in time of the water phase, should also be included [2, 3, 5, 11, 13]. Based on this, we consider here the system:

$$\begin{aligned} \partial_t \theta - \nabla \cdot \left(K(\theta, \Psi)(\nabla \Psi + \mathbf{e_z}) \right) &= \mathbb{S}_1, \\ \Psi + p_{cap}(\theta, c) &= \tau(\theta) \partial_t \theta, \\ \partial_t (\theta c) - \nabla \cdot (D \nabla c - \mathbf{u_w} c) + R(c) &= \mathbb{S}_2. \end{aligned} \tag{1}$$

*Project founded by VISTA, a collaboration between the Norwegian Academy of Science and Letters and Equinor.

The first equation is the Richards equation, whereas the second is an ordinary differential equation used to include the non-equilibrium effects in the capillary pressure/water content relation. Equilibrium models are obtained for $\tau = 0$. Furthermore, the third equation is the reactive transport equation. Here, θ is the water content, Ψ the pressure head, c the concentration of the chemical component, K the conductivity, $\mathbf{e_z}$ the unit vector in the direction opposite to gravity, D the diffusion/dispersion coefficient, $\mathbf{u_w}$ the water flux, $R(c)$ the reaction term and finally $\mathbb{S}_1$ and $\mathbb{S}_2$ are any source terms or external forces involved in the process. Note that $\mathbf{u_w} := -K(\theta, \Psi)(\nabla\Psi + \mathbf{e_z})$ where K is a nonlinear function depending on θ and Ψ. In the van Genuchten model [4] one has:

$$K(\theta, \Psi) = \begin{cases} K_s \theta^{\frac{1}{2}} \left[1 - \left(1 - \theta^{\frac{n}{n-1}} \right)^{\frac{n-1}{n}} \right]^2, & \Psi \leq 0 \\ K_s, & \Psi > 0. \end{cases} \tag{2}$$

K_s is the saturated conductivity and n is a soil dependent parameter.

The system (1) is completed by boundary conditions for Ψ and c, and initial conditions for θ and c.

The rest of the paper is organized as follows: in Section 2 the equations are discretized and linearized. Section 3 includes a numerical example, based on the literature [6], which allows us to compare the different numerical schemes. Finally, Section 4 will conclude this paper with our final remarks.

2 The Numerical Schemes

Applying a Euler implicit time-stepping to (1) gives a sequence of time discrete nonlinear equations. To solve them we apply different linearization schemes: the Newton method, the L-Scheme and a combination of the two [7, 9]. They are here compared thanks to a numerical example inspired by reactive models.

The equations in (1) are fully coupled due to the double dependency of the capillary pressure of both the water content θ and the concentration c. In general, p_{cap} is a function of only θ, e.g., $p_{cap} := 1/\alpha(\theta^{-1/m} - 1)^{1/n}$ as presented in [4]. Anyhow, it has been observed [12] that, if an external component is involved, the surface tension becomes a function of the concentration c and thus, the capillary pressure itself is influenced by this, i.e. $p_{cap} := p_{cap}(\theta, c)$.

In the following, we use the standard notations of functional analysis. The domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2$ or 3, is bounded, open and has a Lipschitz continuous boundary $\partial\Omega$. We denote by $L^2(\Omega)$ the space of real-valued, square-integrable functions defined on Ω and $H^1(\Omega)$ its subspace containing the functions having also the first order weak partial derivatives in $L^2(\Omega)$. $H_0^1(\Omega)$ is the space of functions belonging to $H^1(\Omega)$, having zero values on the boundary $\partial\Omega$. We denote by $< \cdot, \cdot >$ the $L^2(\Omega)$ scalar product and by $\|\cdot\|$ the associated norm. Finally, assume that K is continuous and increasing, $p_{cap} \in C^1\big((0, 1], [0, \infty)\big)$ is decreasing and $\tau \in C^1\big((0, 1], [0, \infty)\big)$.

We now combine the backward Euler method with linear Galerkin finite elements for the discretization of the problem (1). Let $N \in \mathbb{N}$ be a strictly positive natural number, define the time step size $\Delta t = T/N$ and $t_n = n\Delta t$ ($n \in 1, 2, \ldots, N$). Furthermore, $\mathbb{T}_h$ is a regular decomposition of Ω, $\overline{\Omega} = \bigcup_{\mathbb{T} \in \mathbb{T}_h} \mathbb{T}$, into d-dimensional

simplices, with h denoting the maximal mesh diameter. The finite element space $V_h \subset H_0^1(\Omega)$ is defined by

$$V_h := \{v_h \in H_0^1(\Omega)\ s.t.\ v_{h|\mathbb{T}} \in \mathbb{P}_1(\mathbb{T}),\ \mathbb{T} \in \mathbb{T}_h\}, \tag{3}$$

where $\mathbb{P}_1(\mathbb{T})$ denotes the space of the afine polynomials on $\mathbb{T}$.

The fully discrete Galerkin formulation of the system (1) can be written as:

Problem P(n) Let $n \geq 1$ be fixed. Given $\Psi_h^{n-1}, \theta_h^{n-1}, c_h^{n-1} \in V_h$, find $\Psi_h^n, \theta_h^n, c_h^n \in V_h$ such that there holds

$$< \theta_h^n - \theta_h^{n-1}, v_{1,h} > + \Delta t < K(\theta_h^n, \Psi_h^n)(\nabla \Psi_h^n + \mathbf{e_z}), \nabla v_{1,h} > = \Delta t < \mathbb{S}_1, v_{1,h} >, \tag{4}$$

$$\Delta t < \Psi_h^n, v_{2,h} > + \Delta t < p_{cap}(\theta_h^n, c_h^n), v_{2,h} > = < \tau(\theta_h^n)(\theta_h^n - \theta_h^{n-1}), v_{2,h} >, \tag{5}$$

and

$$\begin{aligned} < \theta_h^n c_h^n - \theta_h^{n-1} c_h^{n-1}, v_{3,h} > + \Delta t < D \nabla c_h^n + \mathbf{u_w^{n-1}} c_h^n, \nabla v_{3,h} > \\ + \Delta t < R(c_h^n), v_{3,h} > = \Delta t < \mathbb{S}_2, v_{3,h} >, \end{aligned} \tag{6}$$

for all $v_{1,h}, v_{2,h}, v_{3,h} \in V_h$.

Remark 1 We use $\mathbf{u_w^{n-1}} := -K(\theta_h^{n-1}, \Psi_h^{n-1})(\nabla \Psi_h^{n-1} + \mathbf{e_z})$ for the convective term in the transport equation, for simplicity reasons. Nevertheless, all the simulations presented in this paper have also been performed with $\mathbf{u_w^n} := -K(\theta_h^n, \Psi_h^n)(\nabla \Psi_h^n + \mathbf{e_z})$ instead of $\mathbf{u_w^{n-1}}$ and the results were almost identical.

In the following, we propose different solving strategies for the system of equations presented above. These strategies are built on the ones discussed in [7], extending them to the case of dynamic capillary pressure ($\tau(\theta) \neq 0$). They are either a monolithic solver of the full system, or a splitting approach obtained by solving first the flow component, using a previously computed concentration, then updating the transport equation, using the newly computed pressure and water content. In both cases, one has to iterate. Each iteration requires solving a non-linear problem, for which, either the Newton methods or the L-Scheme [7, 9, 10] are considered. These strategies are then named: monolithic-Newton scheme (MON-Newton), monolithic-L-Scheme (MON-LS), nonlinear splitting-Newton (NonLinS-Newton) and nonlinear splitting-L-Scheme (NonLinS-LS).

The index j denotes the iteration index. As a rule, the iterations start with the solution obtained at the previous time step, for example $\Psi^{n,1} := \Psi^{n-1}$. This is not necessary for the L-Scheme, which is globally convergent, but it appears to be a natural choice.

2.1 The monolithic Newton method (MON-NEWTON)

The Newton method is a well-known linearization scheme, which is quadratic but only locally convergent. Applying the monolithic Newton method to (4)-(6) leads to

Problem MN(n,j+1) Let $\Psi_h^{n-1}, \theta_h^{n-1}, c^{n-1}, \Psi_h^{n,j}, \theta_h^{n,j} c_h^{n,j} \in V_h$ be given, find

$\Psi_h^{n,j+1}, \theta_h^{n,j+1}, c_h^{n,j+1} \in V_h$ such that

$$\begin{aligned} &< \theta_h^{n,j+1} - \theta_h^{n-1}, v_{1,h} > + \Delta t < K(\theta_h^{n,j}, \Psi_h^{n,j})(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_{1,h} > \\ &+ \Delta t < \partial_\theta K(\theta_h^{n,j}, \Psi_h^{n,j})(\nabla(\Psi_h^{n,j}) + \mathbf{e_z})(\theta_h^{n,j+1} - \theta_h^{n,j}), \nabla v_{1,h} > \\ &+ \Delta t < \partial_\Psi K(\theta_h^{n,j}, \Psi_h^{n,j})(\nabla(\Psi_h^{n,j}) + \mathbf{e_z})(\Psi_h^{n,j+1} - \Psi_h^{n,j}), \nabla v_{1,h} > \\ &= \Delta t < \mathbb{S}_1, v_{1,h} >, \end{aligned} \tag{7}$$

$$\begin{aligned} &\Delta t < \Psi_h^{n,j+1}, v_{2,h} > + \Delta t < p_{cap}(\theta_h^{n,j}, c_h^{n,j}), v_{2,h} > \\ &+ \Delta t < \partial_\theta p_{cap}(\theta_h^{n,j}, c_h^{n,j})(\theta_h^{n,j+1} - \theta_h^{n,j}), v_{2,h} > + \Delta t < \partial_c p_{cap}(\theta_h^{n,j}, c_h^{n,j}) \\ &(c_h^{n,j+1} - c_h^{n,j}), v_{2,h} > = < \tau(\theta_h^{n,j})(\theta_h^{n,j+1} - \theta_h^{n-1}), v_{2,h} > \\ &+ < \partial_\theta \tau(\theta_h^{n,j})(\theta_h^{n,j} - \theta_h^{n-1})(\theta_h^{n,j+1} - \theta_h^{n,j}), v_{2,h} >, \end{aligned} \tag{8}$$

and

$$\begin{aligned} &< \theta_h^{n,j} c_h^{n,j+1} - \theta_h^{n-1} c_h^{n-1}, v_{3,h} > + \Delta t < D \nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla v_{3,h} > \\ &+ \Delta t < R(c_h^{n,j}), v_{3,h} > + \Delta t < \partial_c R(c_h^{n,j})(c_h^{n,j+1} - c_h^{n,j}) > \\ &= \Delta t < \mathbb{S}_2, v_{3,h} >, \end{aligned} \tag{9}$$

hold true for all $v_{1,h}, v_{2,h}, v_{3,h} \in V_h$.

2.2 The monolithic L-scheme (MON-LS)

The monolithic L-scheme for solving (4)-(6) reads as

Problem ML(n,j+1) Let $\Psi_h^{n-1}, \theta_h^{n-1}, c^{n-1}, \Psi_h^{n,j}, \theta_h^{n,j} c_h^{n,j} \in V_h$ be given, $L_1^\Psi, L_1^\theta, L_2, L_3 > 0$, big enough.
Find $\Psi_h^{n,j+1}, \theta_h^{n,j+1}, c_h^{n,j+1} \in V_h$ such that

$$\begin{aligned} &< \theta_h^{n,j+1} - \theta_h^{n-1}, v_{1,h} > + \Delta t < K(\theta_h^{n,j}, \Psi_h^{n,j})(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_{1,h} > \\ &+ \Delta t < L_1^\Psi(\Psi_h^{n,j+1} - \Psi_h^{n,j}), \nabla v_{1,h} > + \Delta t < L_1^\theta(\theta_h^{n,j+1} - \theta_h^{n,j}), \nabla v_{1,h} > \\ &= \Delta t < \mathbb{S}_1, v_{1,h} >, \end{aligned} \tag{10}$$

$$\begin{aligned} &\Delta t < \Psi_h^{n,j+1}, v_{2,h} > = -\Delta t < p_{cap}(\theta_h^{n,j}, c_h^{n,j}), v_{2,h} > \\ &+ < \tau(\theta_h^{n,j})(\theta_h^{n,j+1} - \theta_h^{n-1}), v_{2,h} > + < L_2(\theta_h^{n,j+1} - \theta_h^{n,j}), v_{2,h} > \end{aligned} \tag{11}$$

and

$$\begin{aligned} &< \theta_h^{n,j} c_h^{n,j+1} - \theta_h^{n-1} c_h^{n-1}, v_{3,h} > + \Delta t < D \nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla v_{3,h} > \\ &+ \Delta t < R(c_h^{n,j}), v_{3,h} > + < L_3(c_h^{n,j+1} - c_h^{n,j}), v_{3,h} > = \Delta t < \mathbb{S}_3, v_{3,h} >, \end{aligned} \tag{12}$$

hold true for all $v_{1,h}, v_{2,h}, v_{3,h} \in V_h$.

The L-Scheme does not involve the computations of derivatives, the linear systems to be solved within each iteration are better conditioned, compared to the ones given by the Newton method [7, 9], and it is globally (linearly) convergent. The convergence of the scheme has been proved, for the equilibrium model ($\tau(\theta) = 0$) in [7], and can be easily extended to the non-equilibrium formulation given by the system (10)-(12).

2.3 The splitting approach (NonLinS)

The splitting approach for solving (4)-(6) reads as

Problem S(n,j+1) Let $\Psi_h^{n-1}, \theta^{n-1}, c^{n-1}, \Psi_h^{n,j}, \theta_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1}, \theta_h^{n,j+1} \in V_h$ such that

$$< \theta_h^{n,j+1} - \theta_h^{n-1}, v_{1,h} > + \Delta t < K(\theta_h^{n,j+1}, \Psi_h^{n,j+1})(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_{1,h} > = \Delta t < \mathbb{S}_1, v_{1,h} >, \tag{13}$$

$$\begin{aligned} \Delta t < \Psi_h^{n,j+1}, v_{2,h} > + \Delta t < p_{cap}(\theta_h^{n,j+1}, c_h^{n,j}), v_{2,h} > \\ = < \tau(\theta_h^{n,j+1})(\theta_h^{n,j+1} - \theta_h^{n-1}), v_{2,h} >, \end{aligned} \tag{14}$$

hold true for all $v_{1,h}, v_{2,h} \in V_h$.

Then, with $\Psi_h^{n,j+1}$ and $\theta_h^{n,j+1}$ obtained from the equations above, find $c_h^{n,j+1} \in V_h$ such that

$$< \theta_h^{n,j+1} c_h^{n,j+1} - \theta_h^{n-1} c_h^{n-1}, v_{3,h} > + \Delta t < D \nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla v_{3,h} > + \Delta t < R(c_h^{n,j+1}), v_{3,h} > = \Delta t < \mathbb{S}_2, v_{3,h} >, \tag{15}$$

holds true for all $v_{3,h} \in V_h$.

The three equations above can be then linearised using either the Newton method (NonLinS-Newton) or the L-Scheme (NonLinS-LScheme).

2.4 The mixed linearization scheme

It has been already observed, for a different set of equations [9], that combining the Newton method and the L-Scheme can improve the convergence of the scheme. The Newton method is quadratically but only locally convergent and it can produce badly conditioned linearized systems. Moreover, the time step is subject to severe restrictions for guaranteeing the convergence of the scheme, and this has also been observed in numerical examples [1, 7, 9].

Contrarily, the L-Scheme is globally convergent and the linear systems to be solved within each iteration are better conditioned, however, it has only a linear rate of convergence.

The mixed formulation, obtained combining the two schemes, appears to be the best approach and shows practically both global and quadratic convergence. The Newton method commonly fails to converge, if the initial guess is too far from the actual solution. Since this guess is usually the solution at the previous time, this can force restriction on the time step. Instead of reducing the time step one can obtain a better approximation of the initial guess, for the Newton method, by performing few L-Scheme iterations. In the numerical simulation here presented, up to 5 iterations were sufficient to reach a good initial guess for the Newton iteration, which ensured its convergence.

3 Numerical examples

In this section, we use a benchmark problem, from [6], to compare the different linearization schemes and solving algorithms defined above. It describes the recharge of a two-dimensional underground reservoir $\Omega \subset \mathbb{R}^2$, in the interval of time $t \in (0, 3]$. The boundary of the domain and the Dirichlet boundary conditions are defined below.

$$
\begin{aligned}
&\Omega = (0,2) \times (0,3),\\
&\Gamma_{D_1} = \{(x,y) \in \partial\Omega | x \in [0,1] \wedge y = 3\},\\
&\Gamma_{D_2} = \{(x,y) \in \partial\Omega | x = 2 \wedge y \in [0,1]\},\\
&\Gamma_D = \Gamma_{D_1} \cup \Gamma_{D_2},\\
&\Gamma_N = \partial\Omega \setminus \Gamma_D,
\end{aligned}
\qquad
\Psi(x,y,t) = \begin{cases} -2 + 2.2 * t, \text{ on } \Gamma_{D_1}, t \leq 1 \\ 0.2, \text{ on } \Gamma_{D_1}, t > 1 \\ 1 - y, \text{ on } \Gamma_{D_2}, \end{cases}
$$

$$
c(x,y,t) = \begin{cases} 1, \text{ on } \Gamma_{D_1}, t \leq 1 \\ 0, \text{ on } \Gamma_{D_1}, t > 1 \\ 3 - y, \text{ on } \Gamma_{D_2} \cup \Gamma_N. \end{cases}
$$

Furthermore, no flow conditions are imposed on Γ_N. The initial conditions are given by $\Psi(x,y,0) := 1 - y, c(x,y,0) := 3 - y$ and $\theta(x,y,0) := 0.39$. The capillary pressure is defined as $p_{cap}(\theta, c) := (1-\theta)^{2.5} + 0.1 * c$, the conductivity is given by (2) and $\tau(\theta) = 1$. Finally, the parameters implemented are: $K_s = 1$, $L_1^{\Psi}, L_1^{\theta}, L_2 = 0.01, L_3 = 0.1$ and the iterations stop whenever all the error norms, $\left\|\Psi^{n,j+1} - \Psi^{n,j}\right\|, \left\|\theta^{n,j+1} - \theta^{n,j}\right\|$ and $\left\|c^{n,j+1} - c^{n,j}\right\|$, are below 10^{-6}.

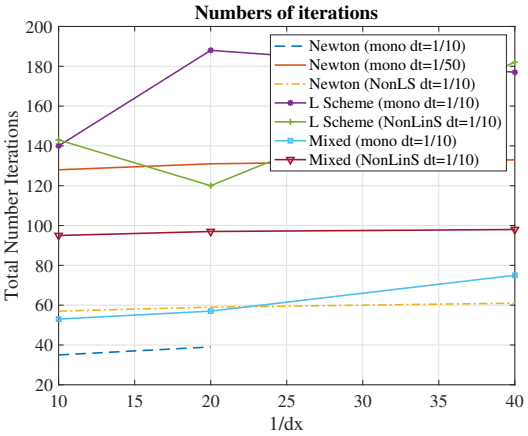


Figure 1: Total numbers of iterations for different solvers

We performed the simulations using regular meshes, consisting of squares, with sides $dx = \{1/10, 1/20, 1/40\}$. We considered two fixed time steps $\Delta t = 1/10$ and $\Delta t = 1/50$.

In Figure 1, we can observe the total numbers of iterations required by the different linearization schemes and solving algorithms. Next to the name of each

scheme we report, between parenthesis, which time step Δt has been used.

We can observe, as the Newton method in the monolithic formulation, converges only for coarse meshes, for $\Delta t = 1/10$. For the smaller time step, $\Delta t = 1/50$, it converges for all of the tested meshes.

The L Scheme converges for both time steps, but, since it is linearly convergent, for $\Delta t = 1/50$ would require more iterations than the Newton method.

The results obtained thanks to the mixed formulation are particularly interesting. We can observe that this scheme, both in the monolithic and splitting formulation, converges for all the tested meshes also in case of a large time step. Moreover, thanks to the Newton iterations, it appears to be faster than the classical L Scheme. It is as robust as the L Scheme and as fast as the Newton method. For more details regarding the mixed scheme, we refer to [9].

4 Conclusions

In this paper, we considered multiphase flow coupled with a one-component reactive transport in variably saturated porous media, including also the dynamic effects in the capillary pressure. The resulting model is nonlinear and for this reason, three different linearization schemes are investigated: the L-Scheme, the Newton method and a combination of the two. We also studied both monolithic solvers and splitting ones.

The tests show that, for this particular set of equations, the best linearization scheme is the one obtained combining the Newton method and the L-Scheme. Such scheme appears to be both quadratically and globally convergent.

Acknowledgments

The research of D. Illiano was funded by VISTA, a collaboration between the Norwegian Academy of Science and Letters and Equinor, project number 6367, project name: adaptive model and solver simulation of enhanced oil recovery. The research of I.S. Pop was supported by the Research Foundation-Flanders (FWO), Belgium through the Odysseus programme (project G0G1316N) and Equinor through the Akademia grant.

We thank the members of the *Sintef* research group and in particular to Dr. Olav Moyner for the assistance with the implementation of the numerical examples in *MRST*, the toolbox based on Matlab developed at *Sintef* itself.

References

[1] Cao, X., Pop, I.S.: Uniqueness of weak solutions for a pseudo-parabolic equation modeling two phase flow in porous media, Applied Mathematics Letters, Volume 46, Pages 25-30, (2015).

[2] Di Carlo, D.: Experimental measurements of saturation overshoot on infiltration, Water Resources Research, Volume 40, Issue 4, (2004).

[3] Fucik, R., Mikyska, J., Sakaki, T., Benes, M., Illangasekare, T.H.: Significance of Dynamic Effect in Capillarity during Drainage Experiments in Layered Porous Media, Vadose Zone Journal 9, Volume 3, (2010).

[4] van Genuchten, M.: A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils, Soil Science Society of America Journal, Volume 44, Issue 5, Pages 892-898, (1980).

[5] Hassanizadeh, S.M., Celia, M.A., Dahle, H.K.: Dynamic Effect in the Capillary Pressure Saturation Relationship and its Impacts on Unsaturated Flow, Vadose Zone Journal, Volume 1, Issue 1, Pages 38-57, (2002).

[6] Haverkamp, R., Vauclin, M., Touma, J., Wierenga, P.J., Vachaud, G.: A comparison of numerical simulation models for one-dimensional infiltration, Soil Science Society of America Journal, Volume 41, Pages 285-294, (1977).

[7] Illiano, D., Pop, I.S., Radu, F.A.: Iterative schemes for surfactant transport in porous media, arXiv preprint arXiv:1906.00224, (2019).

[8] Knabner, P.: Finite element simulation of saturated-unsaturated flow through porous media, LSSC 7, Pages 83-93, (1987).

[9] List, F., Radu, F.A.: A study on iterative methods for solving Richards' equation, Computational Geoscience, Volume 20, Issue 2, Pages 341-353, (2016).

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

[11] Shubao, T., Lei, G., Shun-li, H., Yang, L.: Dynamic effect of capillary pressure in low permeability reservoirs, Petroleum Exploration and Development, Volume 39, Issue 3, Pages 405-411, (2012).

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

[13] Zhuang, L., van Duijn, C.J., Hassanizadeh, S.M.: The effect of dynamic capillarity in modeling saturation overshoot during infiltration, Vadose Zone Journal, Volume 18, Issue 1, Pages 1-14, (2019).

C

Paper C

Efficient Solvers for Nonstandard Models for Flow and Transport in Unsaturated Porous Media

Davide Illiano, Jakub Wiktor Both, Iuliu Sorin Pop, Florin Adrian Radu

Efficient Solvers for Nonstandard Models for Flow and Transport in Unsaturated Porous Media. ...

Efficient Solvers for Nonstandard Models for Flow and Transport in Unsaturated Porous Media

Davide Illiano[a], Jakub Wiktor Both[a], Iuliu Sorin Pop[a,b], Florin Adrian Radu[a]

[a]*Department of Mathematics, University of Bergen, Allegaten 41, Bergen, Norway*
[b]*Faculty of Science, University of Hasselt,Agoralaan Building D, BE 3590 Diepenbeek, Belgium*

Abstract

We study several iterative methods for fully coupled flow and reactive transport in porous media. The resulting mathematical model is a coupled, nonlinear evolution system. The flow model component builds on the Richards equation, modified to incorporate nonstandard effects like dynamic capillarity and hysteresis, and a reactive transport equation for the solute. The two model components are strongly coupled. On one hand, the flow affects the concentration of the solute; on the other hand, the surface tension is a function of the solute, which impacts the capillary pressure and, consequently, the flow. After applying an Euler implicit scheme, we consider a set of iterative linearization schemes to solve the resulting nonlinear equations, including both monolithic and two splitting strategies. The latter include a canonical nonlinear splitting and an alternate linearized splitting, which appears to be overall faster in terms of numbers of iterations, based on our numerical studies. The (time discrete) system being nonlinear, we investigate different linearization methods. We consider the linearly convergent L-scheme, which converges unconditionally, and the Newton method, converging quadratically but subject to restrictions on the initial guess. Whenever hysteresis effects are included, the Newton method fails to converge. The L-scheme converges; nevertheless, it may require many iterations. This aspect is improved by using the Anderson acceleration. A thorough comparison of the different solving strategies is presented in five numerical examples, implemented in MRST, a toolbox based on MATLAB.

1. Introduction

Mathematical models for complex physical phenomena are generally neglecting several processes, in order to guarantee that the result is sufficiently simple and to facilitate the numerical simulations. With a particular focus on porous media applications, in this sense we mention enhanced oil recovery, diffusion of substances in living tissues, and pollution of underground aquifers. With the increase of computational power, and the development of efficient simulation algorithms, mathematical models are improved continuously, and more and more of the neglected effects are included.

Email addresses: Davide.Illiano@uin.no (Davide Illiano), Jakub.Both@uib.no (Jakub Wiktor Both), sorin.pop@uhasselt.be (Iuliu Sorin Pop), Florin.Radu@uib.no (Florin Adrian Radu)

When studying unsaturated flow, the equilibrium capillary pressure plays a fundamental role. It is typically assumed to be a nonlinear, monotone function of the water content. Explicit representations have been obtained thanks to numerous experiments under equilibrium conditions (no flowing phases). Even though this formulation has been the most commonly used in the last decades, it has been observed [Camps-Roach et al.-2010, DiCarlo-2004, Oung et al.-2005, Stauffer-1978], that changes in time of the water content, thus its time derivatives, do influence the profile of the capillary pressure. In terms of modeling, this is achieved by including the so-called dynamic effects [Beliaev et al.-2001, Gray et al.-1998, Mikelic-2010]. Numerous papers investigate the existence of a solution for systems including such effects, among them we cite [Cao et al.-2016, Koch et al.-2012, Milisic -2018]. Furthermore, the problem has been already studied numerically in, e.g., [Abreu et al.-217, Abreu et al.-2020, Cao et al.-2019a].

The hysteresis effect is another phenomenon often neglected. Again, experiments have revealed that the curve obtained when investigating the imbibition process, is different from the one observed during the drainage, [O'Carroll et al.-2005, Hoa et al.-1977, McClure et al.-2018, Morrow et al.-2005]. This is sketched in Fig. 1.

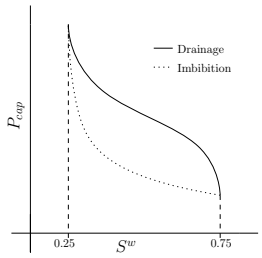


Figure 1: Primary hysteresis loop as presented in [McClure et al.-2018].

In this article, we study unsaturated flow in porous media, modeled by the Richards equation [Berardi et al.-2018, Helmig-1997], however including both dynamic and hysteresis effects. Furthermore, we include a solute component, e.g., a surfactant, in the wetting phase, which can directly influence the fluid properties ([Agosti et al.-2015, Prechtel et al.-2002]). The study of the transportation of an external components, e.g., surfactant, in variably saturated porous media has been already investigated both numerically [Illiano et al.-2020a, Knabner et al.-2003, Radu et al.-2013] and experimentally [Henry et al.-1999, Karagunduz et al.-2015].

Here, we will mainly concentrate on numerical studies, extending the solution techniques in [Illiano et al.-2020a] to include dynamic capillarity and hysteresis We consider the following model for coupled unsaturated flow and reactive transport

$$\begin{aligned} \partial_t \theta(\Psi, c) - \nabla \cdot (K(\theta(\Psi, c))\nabla(\Psi + z)) &= \mathbb{S}_1, \\ \partial_t(\theta(\Psi, c)c) - \nabla \cdot (D\nabla c - \mathbf{u_w}c) + R(c) &= \mathbb{S}_2. \end{aligned} \tag{1}$$

Here, $\theta(\Psi, c)$ is the water content, expressed as a function of both the unknown pressure head Ψ and the concentration of the external component c. K, a function of the water content θ, is the conductivity, z the vertical coordinate of $\vec{x}$, pointing against gravity, D the dispersion/diffusion

coefficient, $\mathbf{u_w} := -K(\theta(\Psi, c))\nabla(\Psi + z)$ the water flux, $R(c)$ the reaction term and $\mathbb{S}_1, \mathbb{S}_2$ the external sink/source terms involved.

Next to a concentration dependence of θ and Ψ, here we include also play-type hysteresis and dynamic capillary effects as introduced in [Beliaev et al.-2001]. More precisely,

$$\Psi \in -p_{cap}(\theta, c) + \tau(\theta)\partial_t\theta + \gamma(\theta)\,\mathrm{sign}(\partial_t\theta), \tag{2}$$

where p_{cap} is the equilibrium capillary pressure, expressed as a function of θ and c, $\tau(\theta)$ the dynamic effects, and $\gamma(\theta)$ the width of the primary hysteresis loop. Later on, for ease of presentation, we consider γ as a positive constant, $\gamma \in \mathbb{R}_{\geq 0}$. Note that (2) is a differential inclusion as the sign graph is multi-valued and defined as follow,

$$sign(\xi) = \begin{cases} 1 & \text{for } \xi > 0, \\ [-1, 1] & \text{for } \xi = 0, \\ -1 & \text{for } \xi < 0. \end{cases} \tag{3}$$

The multi-valued graph allows switching between the imbibition and drainage curves in the play-type hysteresis. For more details on the formulation we refer to [Beliaev et al.-2001].

The primary unknowns of the system are the pressure Ψ, the concentration c and the water content θ. In standard models, also obtained as special case for $\gamma = \tau(\theta) = 0$, θ is a function of pressure and concentration. Therefore, (2) is replaced by an algebraic relationship, which simplifies the model and allows eliminating θ as an unknown. In the extended/nonstandard formulation, θ is an unknown and (2) is required as additional equation of the model. Initial and boundary conditions will complete the system.

To avoid working with a graph, we consider the following regularization,

$$\Phi(\xi) = \begin{cases} \mathrm{sign}(\xi) & \text{if} \quad |\xi| \geq \delta, \\ \frac{\xi}{\delta} & \text{if} \quad |\xi| < \delta, \end{cases} \tag{4}$$

where $\delta \in \mathbb{R}^+$ is a small parameter. Using this in (2) gives the regularized system of equations

$$\begin{aligned} \partial_t\theta - \nabla \cdot (K(\theta)\nabla(\Psi + z)) &= \mathbb{S}_1, \\ \Psi = -p_{cap}(\theta, c) + \tau(\theta)\partial_t\theta + \gamma\Phi(\partial_t\theta), & \\ \partial_t(\theta c) - \nabla \cdot (D\nabla c - \mathbf{u_w} c) + R(c) &= \mathbb{S}_2. \end{aligned} \tag{5}$$

From now on, the system (5) will be further investigated. We will discretize the equations and study different solving algorithms.

Remark 1. *An inverse formulation is proposed in [Beliaev et al.-2001], obtained by solving* (2)*, as its regularized counterpart in* (5)*, in terms of $\partial_t\theta$. This gives*

$$\partial_t\theta = F(\Psi, \theta, c), \tag{6}$$

for a suitable function F. The time derivative in the flow equation can then be substituted by F,

$$F(\Psi, \theta, c) - \nabla \cdot (K(\theta)\nabla(\Psi + z)) = \mathbb{S}_1. \tag{7}$$

This formulation is used for the mathematical analysis of such models, [Cao et al.-2015, Schweizer-2012]. It has been observed, e.g., in [Lunowa et al.-2020], that such formulation can reduce the number of iterations required to solve the system of equations, compared to the formulation in (2)*. However, for the particular test cases investigated here, no remarkable improvements are observed. Thus, for ease of presentation, we will report the results obtained only for the formulation given by (5).*

We point out that the concentration of the external component directly influences the capillary pressure. The presence of such a component results in a non-constant surface tension, which induces a rescaling of the pressures [Husseini-2015, Smith et al.-1994, Smith et al.-1999].

To solve the system (5) numerically, one first needs to discretize in time and space, and then develop solvers for the discretized equations. In this paper, due to the expected low regularity of the solutions [Alt et al.-1983] and the desire of relatively large time steps, we choose to use the backward Euler method for the time discretization. Certain processes investigated in porous media flow can take place on time intervals longer than decades, thus the need for large time steps. Multiple spatial discretization techniques are available, e.g., the Galerkin Finite Element Method (*FEM*) [Barrett et al.-1997, Nocheto et al.-1988, Russell et al.-1983], Discontinuous Galerkin Method (*DGM*) [Arnold et al.-2006, Karpinski et al.-2017, Li et al.-2007, Sun et al.-2005], the Mixed Finite Element Method (*MFEM*) [Arbogast-1996, Cao et al.-2019a, Radu et al.-2010, Radu et al.-2013, Vohralik-2007, Woodward et al.-2000], the Finite Volume Method (*FVM*) [Eymard et al.-1999] and the Multi-Point Flux Approximation (*MPFA*) [Aavatsmark-2001, Arraras et al.-2020, Bause et al.-2010, Klausen et al.-2008]. We will here concentrate on *FEM* and *TPFA* (Two Points Flux Approximation), a particular case of *MPFA*. In particular, we cite [Berardi et al.-2020, Dolejsi et al.2019, Zha et al.2019] for papers on improved numerical schemes applied to the Richards equation.

Since the equations investigated here are characterized by several nonlinear quantities, $K(\theta)$, $p_{cap}(\theta, c)$, $\tau(\partial_t\theta)$, and $R(c)$, and the time discretization is not explicit, one needs to solve a nonlinear system at each time step, requiring a linearization procedure. Examples of possible linearization schemes are: the Newton method [Paniconi et al.-1994], the modified Picard method [Celia et al.-1990] and the L-scheme [List et al.-2016, Pop et al.-2004]. In this paper, we investigate the Newton method and the L-scheme. The former is a commonly used linearization scheme which is quadratically convergent. However, this convergence is only local and one needs to compute the Jacobian matrix, which can be expensive. The L-scheme is instead globally (linearly) convergent, under mild restrictions, and it does not require the computation of any derivative. The L-scheme is in general slower in terms of numbers of iterations than the Newton method. Moreover, the linear systems to be solved within each iteration are better conditioned when compared to the ones given by the Newton method [Illiano et al.-2020a, List et al.-2016]. Furthermore, the rate of convergence of the scheme strongly depends on user-defined parameters. Such aspects are investigated for numerous nonlinear problems, including Richards equation, and two-phase flow in porous media, in [Illiano et al.-2020a, List et al.-2016, Mitra et al.-2019, Pop et al.-2004, Slodicka-2002]. Finally there numerous papers proposing improved formulation of the L-scheme, among them we cite [Albuja et al.-2021, Mitra et al.-2019].

In this work, we test the L-scheme on more complex problems involving hysteresis and dy-

namic effects, and coupled reactive transport and flow. Furthermore, we investigate a post-processing technique, the Anderson Acceleration (AA) [Anderson-1965], which can drastically improve linearly convergent schemes. The acceleration tool requires user-defined parameters. As will be seen below, choosing the suitable parameters for the AA, significantly relaxes the choice of the parameters for the L-scheme linearization.

We observe that the system (5) is fully coupled. This is due to the dependence of the capillary pressure on both θ and c. Therefore, we will investigate multiple solution algorithms, combining different linearization schemes and decoupling techniques. Decoupling/splitting the equations may present multiple advantages such as: an easier implementation, a better conditioned problem to solve, similar convergence properties but faster computations. We divide the schemes into three main categories: monolithic (Mono), nonlinear splitting (NonLinS) and alternate splitting (AltS). Subsequently, we denote, e.g., by Newton-Mono, the monolithic scheme obtained by applying the Newton method as linearization. Such schemes have already been investigated for the standard model in [Illiano et al.-2020a].

The paper is organized as follows. In Section 2, we present the linearization and discretization techniques including monolithic or decoupled solution approaches. Section 3 presents five different numerical examples, which allow to compare the efficiency and robustness of the solving algorithms. Section 4 concludes this work with the final remarks.

2. Problem formulation, discretization and iterative schemes

In the following, we use the standard notations of functional analysis. The domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2$ or 3, is bounded and has a Lipschitz continuous boundary $\partial\Omega$. The final time is $T > 0$, and the time domain is $(0, T]$. $L^2(\Omega)$ denotes the space of real valued, square integrable functions defined on Ω and $H^1(\Omega)$ its subspace containing the functions also having weak first derivatives in $L^2(\Omega)$. $H_0^1(\Omega)$ is the space of functions belonging to $H^1(\Omega)$, having zero trace on the boundary $\partial\Omega$. Furthermore, we denote by $< \cdot, \cdot >$ the standard $L^2(\Omega)$ scalar product and by $\|\cdot\|$ the associated norm.

To numerically solve the system of equations (5), one needs to discretize both in time and space. We combine the backward Euler method with linear Galerkin finite elements. Let $N \in \mathbb{N}$ be a strictly positive natural number. We define the time step size $\Delta t = T/N$ and $t_n = n\Delta t$ ($n \in 1, 2, \ldots, N$). Furthermore, let T_h be a regular decomposition of Ω, $\overline{\Omega} = \bigcup_{T \in T_h} T$, with h denoting the mesh diameter. The finite element spaces $V_h \subset H_0^1(\Omega)$ and $W_h \subset L^2(\Omega)$ are defined by

$$V_h := \left\{ v_h \in H_0^1(\Omega) \; s.t. \; v_{h|T} \in \mathbb{P}_1(T), \; T \in T_h \right\}, \qquad W_h := \left\{ w_h \in L^2(\Omega) \; s.t. \; w_{h|T} \in \mathbb{P}_1(T), \; T \in T_h \right\}, \tag{8}$$

where $\mathbb{P}_1(T)$ denotes the space of the linear polynomials on T. The fully discrete Galerkin formulation of the system (5) can now be written as:

Problem Pn: Let $n \geq 1$ be fixed. Assuming that $\Psi_h^{n-1}, c_h^{n-1} \in V_h$ and $\theta_h^{n-1} \in W_h$ are given, find

$\Psi_h^n, c_h^n \in V_h$ and $\theta_h^n \in W_h$ such that

$$\begin{aligned}
&< \theta_h^n - \theta_h^{n-1}, v_{1,h} > + \Delta t < K(\theta_h^n)(\nabla \Psi_h^n + \mathbf{e_z}), \nabla v_{1,h} > = \Delta t < \mathbb{S}_1, v_{1,h} > \\
\Delta t < \Psi_h^n, w_{1,h} > + \Delta t < p_{cap}(\theta_h^n, c_h^n), w_{1,h} > - < \tau(\theta_h^n)(\theta_h^n - \theta_h^{n-1}), w_{1,h} > &= \Delta t \gamma < \Phi\left(\frac{\theta_h^n - \theta_h^{n-1}}{\Delta t}\right), w_{1,h} > \\
< \theta_h^n(c_h^n - c_h^{n-1}) + c_h^n(\theta_h^n - \theta_h^{n-1}), v_{2,h} > + \Delta t < D\nabla \Psi_h^n + \mathbf{u_w^n} c_h^n, \nabla v_{2,h} > + \Delta t < R(c_h^n), v_{2,h} > &= \Delta t < \mathbb{S}_2, v_{2,h} >
\end{aligned} \tag{9}$$

holds for all $v_{1,h}, v_{2,h} \in V_h$ and for all $w_{1,h} \in W_h$. We denote by $\mathbf{e_z}$ the unit vector in the direction opposite to gravity.

Observe that choosing the space $H_0^1(\Omega)$ implies that homogeneous boundary conditions have been adopted for the pressure and the concentration. However, this choice is made for the ease of presentation, the extension to other boundary conditions being possible without major complications. We also mention that, for $n = 1$, we use the approximation in V_h of the initial water content and concentration, respectively θ_h^0 and c_h^0.

In the following, we investigate different iterative schemes for solving Problem Pn. These schemes are based on the ones discussed in [Illiano et al.-2020a], extending them, not only to the case of dynamic capillary pressure ($\tau(\theta) \neq 0$) [Illiano et al.-2020b], but also to the case of hysteresis. Among the numerous papers investigating numerically the effects of hysteresis and dynamic capillarity pressure, we cite [Peszynska et al.-2008, Zhang et al.-2017]. As mentioned, we compare monolithic (Mono) and splitting (NonLinS and AltS) solvers, combined with two different linearization schemes, the Newton method and the L-scheme. Furthermore, the Anderson acceleration [Anderson-1965] will be taken into account to speed up the linearly convergent L-scheme.

2.1. Solving algorithms

In what follows, when solving (9) iteratively, the index n will always refer to the time step level, whereas j will denote the iteration index. As a rule, the iterations will start with the solution at the last time step, t_{n-1}, for example $\Psi^{n,1} = \Psi^{n-1}$. As mentioned, this choice is not required for L-type schemes but it is a natural one.

In a compact form Problem Pn can be seen as the system

$$\begin{cases} F_1(\Psi_h^n, \theta_h^n) & = 0, \\ F_2(\Psi_h^n, \theta_h^n, c_h^n) & = 0, \\ F_3(\Psi_h^n, \theta_h^n, c_h^n) & = 0, \end{cases} \tag{10}$$

with F_1, F_2 resulting from the flow equations and F_3 from the transport. In the following we will indicate with F^{lin}, the linearized formulation of F obtained by either the Newton method or the L-scheme. Finally, we can proceed to present monolithic and splitting solvers.

In the monolithic approach one solves the three equations of the system (10) at once. Formally, one iteration is:
Find $\Psi_h^{n,j+1}$, $\theta_h^{n,j+1}$ and $c_h^{n,j+1}$ such that

$$\begin{cases} F_1^{lin}(\Psi_h^{n,j+1}, \theta_h^{n,j+1}) & = 0, \\ F_2^{lin}(\Psi_h^{n,j+1}, \theta_h^{n,j+1}, c_h^{n,j+1}) & = 0, \\ F_3^{lin}(\Psi_h^{n,j+1}, \theta_h^{n,j+1}, c_h^{n,j+1}) & = 0, \end{cases} \tag{11}$$

where F_i^{Lin} is the linearization of F_i, $i \in \{1, 2, 3\}$. Depending on which linearization technique is used, we refer to the Newton-monolithic scheme (Newton-Mono) or monolithic-L-scheme (LS-Mono). These two schemes will be presented in details below. Fig. 2 displays the sketched version of the monolithic solver.

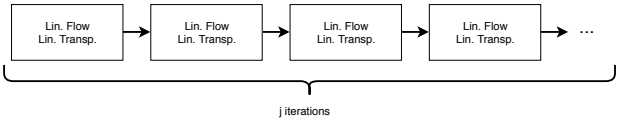


Figure 2: The monolithic approach.

In the iterative splitting approach, the flow and the transport equations are solved subsequently, iterating between them. We will distinguish between two primary splitting schemes: the nonlinear splitting (NonLinS) and the alternate linearized splitting (AltLinS), illustrated in Figure 3 and Figure 4, respectively. Such schemes have already been studied, in the case of a standard flow model in [Illiano et al.-2020a].
In the nonlinear splitting, one iteration step is:
Find first $\Psi_h^{n,j+1}$, $\theta_h^{n,j+1}$ such that

$$\begin{cases} F_1(\Psi_h^{n,j+1}, \theta_h^{n,j+1}) & = 0, \\ F_2(\Psi_h^{n,j+1}, \theta_h^{n,j+1}, c_h^{n,j}) & = 0, \end{cases} \tag{12}$$

and then find $c_h^{n,j+1}$ such that

$$F_3(\Psi_h^{n,j+1}, \theta_h^{n,j+1}, c_h^{n,j+1}) = 0. \tag{13}$$

The two flow equations are solved at once. Each of the nonlinear systems (12) and (13) is solved until some convergence criterion is met. Once the pressure and water content are obtained, $\Psi_h^{n,j+1}$ and $\theta_h^{n,j+1}$, are then used in the transport equation (13) to compute $c_h^{n,j+1}$. The resulting F_1, F_2 and F_3, being nonlinear, are linearized using the Newton method or the L-scheme.

In contrast, the alternate linearized splitting (AltLinS) schemes perform only one linearization step per iteration, see Figure 4. One iteration in the alternate splitting scheme can be written as:
Find $\Psi_h^{n,j+1}, \theta_h^{n,j+1}$ such that

$$\begin{cases} F_1^{lin}(\Psi_h^{n,j+1}, \theta_h^{n,j+1}) & = 0, \\ F_2^{lin}(\Psi_h^{n,j+1}, \theta_h^{n,j+1}, c_h^{n,j}) & = 0, \end{cases} \tag{14}$$

and then $c_h^{n,j+1}$ such that

$$F_3^{lin}(\Psi_h^{n,j+1}, \theta_h^{n,j+1}, c_h^{n,j+1}) = 0. \tag{15}$$

Again, depending on which linearization is used, we refer to alternate splitting Newton (AltS-Newton) or alternate splitting L-scheme (AltS-LS). Both schemes will be presented in detail below.

In the following sections we will illustrate, in the details, the different schemes here investigated.

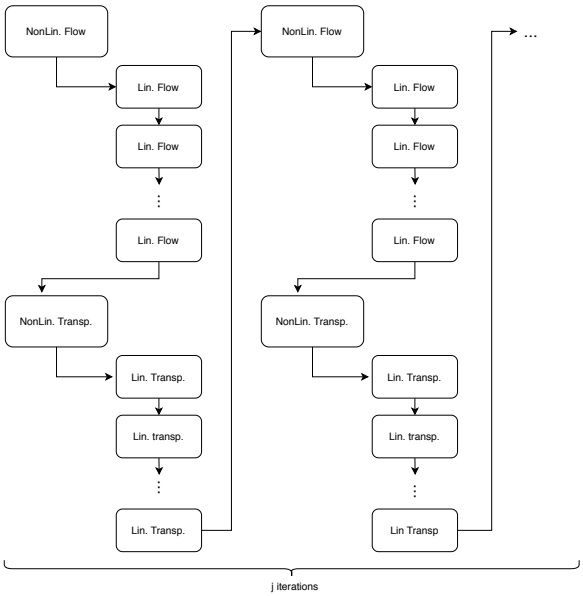


Figure 3: The nonlinear splitting approach.

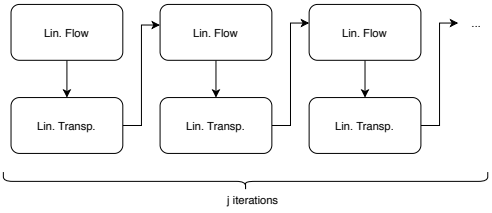


Figure 4: The alternate splitting approach.

2.1.1. The monolithic Newton method (Newton-Mono)

The standard monolithic Newton method applied to (9) reads as:

Problem P-Newton-Mono: Let $j > 1$ be fixed. Let $\Psi_h^{n-1}, \Psi_h^{n,j}, c_h^{n-1}, c_h^{n,j} \in V_h$, and $\theta_h^{n-1}, \theta_h^{n,j} \in W_h$ be given, find $\Psi_h^{n,j+1}, c_h^{n,j+1} \in V_h$, and $\theta_h^{n,j+1} \in W_h$ such that

$$\begin{aligned} &< \theta_h^{n,j+1} - \theta_h^{n-1}, v_{1,h} > + \Delta t < K(\theta_h^{n,j})(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_{1,h} > \\ &+ \Delta t < \partial_\theta K(\theta_h^{n,j})(\nabla(\Psi_h^{n,j}) + \mathbf{e_z})(\theta_h^{n,j+1} - \theta_h^{n,j}), \nabla v_{1,h} >= \Delta t < \mathbb{S}_1, v_{1,h} > \end{aligned} \tag{16}$$

$$\begin{aligned}\Delta t < \Psi_h^{n,j+1}, w_{1,h} >= -\Delta t < p_{cap}(\theta_h^{n,j}, c_h^{n,j}), w_{1,h} > -\Delta t < \partial_\theta p_{cap}(\theta_h^{n,j}, c_h^{n,j})(\theta_h^{n,j+1} - \theta_h^{n,j}), w_{1,h} > \\ - \Delta t < \partial_c p_{cap}(\theta_h^{n,j}, c_h^{n,j})(c_h^{n,j+1} - c_h^{n,j}), w_{1,h} > + < \tau(\theta_h^{n,j})(\theta_h^{n,j+1} - \theta_h^{n-1}), w_{1,h} > \\ + < \partial_\theta \tau(\theta_h^{n,j})(\theta_h^{n,j} - \theta_h^{n-1})(\theta_h^{n,j+1} - \theta_h^{n,j}), w_{1,h} > + \Delta t\gamma < \Phi\left(\frac{\theta_h^{n,j} - \theta_h^{n-1}}{\Delta t}\right), w_{1,h} >\end{aligned} \tag{17}$$

and

$$\begin{aligned}< \theta_h^{n,j}(c_h^{n,j+1} - c_h^{n-1}) + c_h^{n,j}(\theta_h^{n,j+1} - \theta_h^{n-1}), v_{2,h} > + \Delta t < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n,j}} c_h^{n,j+1}, \nabla v_{2,h} > \\ + \Delta t < R(c_h^{n,j}), v_{2,h} > + \Delta t < \partial_c R(c_h^{n,j})(c_h^{n,j+1} - c_h^{n,j}) >= \Delta t < \mathbb{S}_2, v_{2,h} >\end{aligned} \tag{18}$$

hold true for all $v_{1,h}, v_{2,h} \in V_h$, and for all $w_{1,h} \in W_h$. By ∂_θ we denote the partial derivative with respect to the water content θ, and by ∂_c the partial derivative with respect to the concentration c, and $\mathbf{u_w^{n,j}} := -K(\theta_h^{n,j})\nabla(\Psi_h^{n,j} + \mathbf{e_z})$.

2.1.2. The monolithic L-scheme (LS-Mono)

The monolithic L-scheme for solving (9) reads:

Problem P-LS-Mono: Let $j > 1$ be fixed. Let $\Psi_h^{n-1}, \Psi_h^{n,j}, c_h^{n-1}, c_h^{n,j} \in V_h$, and $\theta_h^{n-1}, \theta_h^{n,j} \in W_h$ be given, find $\Psi_h^{n,j+1}, c_h^{n,j+1} \in V_h$, and $\theta_h^{n,j+1} \in W_h$ such that

$$< \theta_h^{n,j+1} - \theta_h^{n-1}, v_{1,h} > + \Delta t < K(\theta_h^{n,j})(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_{1,h} > + L_1 < \Psi_h^{n,j+1} - \Psi_h^{n,j}, v_{1,h} >= \Delta t < \mathbb{S}_1, v_{1,h} > \tag{19}$$

$$\begin{aligned}\Delta t < \Psi_h^{n,j+1}, w_{1,h} > = -\Delta t < p_{cap}(\theta_h^{n,j}, c_h^{n,j}), w_{1,h} > + < \tau(\theta_h^{n,j})(\theta_h^{n,j+1} - \theta_h^{n-1}), w_{1,h} > \\ + \Delta t\gamma < \Phi\left(\frac{\theta_h^{n,j} - \theta_h^{n-1}}{\Delta t}\right), w_{1,h} > + L_2 < (\theta_h^{n,j+1} - \theta_h^{n,j}), w_{1,h} >\end{aligned} \tag{20}$$

and

$$\begin{aligned}< \theta_h^{n,j}(c_h^{n,j+1} - c_h^{n-1}) + c_h^{n,j}(\theta_h^{n,j+1} - \theta_h^{n-1}), v_{2,h} > + \Delta t < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n,j}} c_h^{n,j+1}, \nabla v_{2,h} > \\ + \Delta t < R(c_h^{n,j}), v_{2,h} > + L_3 < c_h^{n,j+1} - c_h^{n,j}, v_{2,h} >= \Delta t < \mathbb{S}_2, v_{2,h} >\end{aligned} \tag{21}$$

hold true for all $v_{1,h}, v_{2,h} \in V_h$, and for all $w_{1,h} \in W_h$. L_1, L_2 and L_3 are three positive, user-defined parameters on which only mild conditions are imposed. We refer to [Illiano et al.-2020a, List et al.-2016, Pop et al.-2004] for the analysis of the numerical schemes which have inspired the ones presented here. Often, one needs to properly tune these parameters to obtain a robust and relatively fast solver.

2.1.3. The nonlinear splitting approach (NonLinS)

The nonlinear splitting approach for solving (9) reads:

Problem P-NonLinS: Let $j > 1$ be fixed. Let $\Psi_h^{n-1}, \Psi_h^{n,j}, c_h^{n-1}, c_h^{n,j} \in V_h$ and $\theta_h^{n-1}, \theta_h^{n,j} \in W_h$ be given, find $\Psi_h^{n,j+1} \in V_h$, and $\theta_h^{n,j+1} \in W_h$ such that

$$< \theta_h^{n,j+1} - \theta_h^{n-1}, v_{1,h} > + \Delta t < K(\theta_h^{n,j+1})(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_{1,h} >= \Delta t < \mathbb{S}_1, v_{1,h} > \tag{22}$$

$$\Delta t < \Psi_h^{n,j+1}, w_{1,h} >= -\Delta t < p_{cap}(\theta_h^{n,j+1}, c_h^{n,j}), w_{1,h} > + < \tau(\theta_h^{n,j+1})(\theta_h^{n,j+1} - \theta_h^{n-1}), w_{1,h} > + \Delta t\gamma < \Phi\left(\frac{\theta_h^{n,j} - \theta_h^{n-1}}{\Delta t}\right), w_{1,h} > \tag{23}$$

holds true for all $v_{1,h} \in V_h$ and for all $w_{1,h} \in W_h$.

Then let $\Psi_h^{n-1}, \Psi_h^{n,j}, c_h^{n-1}, c_h^{n,j} \in V_h$ and $\theta_h^{n-1}, \theta_h^{n,j} \in W_h$ be given, $\Psi_h^{n,j+1} \in V_h$ and $\theta_h^{n,j+1} \in W_h$ are obtained from the equations above, find $c_h^{n,j+1} \in V_h$ such that

$$< \theta_h^{n,j+1}(c_h^{n,j+1} - c_h^{n-1}) + c_h^{n,j}(\theta_h^{n,j+1} - \theta_h^{n-1}), v_{2,h} > + \Delta t < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n,j+1}} c_h^{n,j+1}, \nabla v_{2,h} > + \Delta t < R(c_h^{n,j+1}), v_{2,h} >= \Delta t < \mathbb{S}_2, v_{2,h} > \tag{24}$$

holds true for all $v_{2,h} \in V_h$. The water flux is given by $\mathbf{u_w^{n,j+1}} := -K(\theta_h^{n,j+1})\nabla(\Psi_h^{n,j+1} + \mathbf{e_z})$.

Observe that (22)–(23) and (24) are nonlinear. To approximate their respective solutions, one can employ, e.g., the Newton method (NonLinS-Newton) or the L-scheme (NonLinS-LS).

2.1.4. The alternate splitting Newton method (Newton-AltLinS)

Applied to (9), the alternate splitting Newton method reads:

Problem P-Newton-AltLinS: Let $j > 1$ be fixed. Let $\Psi_h^{n-1}, \Psi_h^{n,j}, c_h^{n-1}, c_h^{n,j} \in V_h$ and $\theta_h^{n-1}, \theta_h^{n,j} \in W_h$ be given, find $\Psi_h^{n,j+1} \in V_h$, and $\theta_h^{n,j+1} \in W_h$ such that

$$\begin{aligned} &< \theta_h^{n,j+1} - \theta_h^{n-1}, v_{1,h} > + \Delta t < K(\theta_h^{n,j})(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_{1,h} > \\ &+ \Delta t < \partial_\theta K(\theta_h^{n,j})(\nabla(\Psi_h^{n,j}) + \mathbf{e_z})(\theta_h^{n,j+1} - \theta_h^{n,j}), \nabla v_{1,h} >= \Delta t < \mathbb{S}_1, v_{1,h} > \end{aligned} \tag{25}$$

$$\begin{aligned} \Delta t < \Psi_h^{n,j+1}, w_{1,h} >= &-\Delta t < p_{cap}(\theta_h^{n,j}, c_h^{n,j}), w_{1,h} > -\Delta t < \partial_\theta p_{cap}(\theta_h^{n,j}, c_h^{n,j})(\theta_h^{n,j+1} - \theta_h^{n,j}), w_{1,h} > \\ &+ < \tau(\theta_h^{n,j})(\theta_h^{n,j+1} - \theta_h^{n-1}), w_{1,h} > + < \partial_\theta \tau(\theta_h^{n,j})(\theta_h^{n,j} - \theta_h^{n-1})(\theta_h^{n,j+1} - \theta_h^{n,j}), w_{1,h} > \\ &+ \Delta t\gamma < \Phi\left(\frac{\theta_h^{n,j} - \theta_h^{n-1}}{\Delta t}\right), w_{1,h} > \end{aligned} \tag{26}$$

hold true for all $v_{1,h} \in V_h$ and $w_{1,h} \in W_h$.

Then, with given $\Psi_h^{n-1}, \Psi_h^{n,j}, c_h^{n-1}, c_h^{n,j} \in V_h$ and $\theta_h^{n-1}, \theta_h^{n,j} \in W_h$, $\Psi_h^{n,j+1} \in V_h$ and $\theta_h^{n,j+1} \in W_h$ are obtained from the equations above, find $c_h^{n,j+1} \in V_h$ such that

$$\begin{aligned} &< \theta_h^{n,j+1}(c_h^{n,j+1} - c_h^{n-1}) + c_h^{n,j}(\theta_h^{n,j+1} - \theta_h^{n-1}), v_{2,h} > + \Delta t < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n,j+1}} c_h^{n,j+1}, \nabla v_{2,h} > \\ &+ \Delta t < R(c_h^{n,j}), v_{2,h} > + \Delta t < \partial_c R(c_h^{n,j})(c_h^{n,j+1} - c_h^{n,j}), v_{2,h} >= \Delta t < \mathbb{S}_2, v_{2,h} > \end{aligned} \tag{27}$$

hold true for all $v_{2,h} \in V_h$.

2.1.5. The alternate splitting L-scheme (LS-AltLinS)

The alternate splitting L-scheme for solving (9) is:

Problem P-LS-AltLinS: Let $j > 1$ be fixed. Let $\Psi_h^{n-1}, \Psi_h^{n,j}, c_h^{n-1}, c_h^{n,j} \in V_h$ and $\theta_h^{n-1}, \theta_h^{n,j} \in W_h$ be given, find $\Psi_h^{n,j+1} \in V_h$, and $\theta_h^{n,j+1} \in W_h$ such that

$$< \theta_h^{n,j+1} - \theta_h^{n-1}, v_{1,h} > + \Delta t < K(\theta_h^{n,j})(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_{1,h} > + L_1 < \Psi_h^{n,j+1} - \Psi_h^{n,j}, v_{1,h} >= \Delta t < \mathbb{S}_1, v_{1,h} > \tag{28}$$

$$\Delta t < \Psi_h^{n,j+1}, w_{1,h} >= -\Delta t < p_{cap}(\theta_h^{n,j}, c_h^{n,j}), w_{1,h} > + < \tau(\theta_h^{n,j})(\theta_h^{n,j+1} - \theta_h^{n-1}), w_{1,h} > \\ + L_2 < \theta_h^{n,j+1} - \theta_h^{n,j}, w_{1,h} > + \Delta t\gamma < \Phi\left(\frac{\theta_h^{n,j} - \theta_h^{n-1}}{\Delta t}\right), w_{1,h} > \tag{29}$$

holds true for all $v_{1,h} \in V_h$ and $w_{1,h} \in W_h$.

Then, with given $\Psi_h^{n-1}, \Psi_h^{n,j}, c_h^{n-1}, c_h^{n,j} \in V_h$ and $\theta_h^{n-1}, \theta_h^{n,j} \in W_h$, and $\Psi_h^{n,j+1} \in V_h$ and $\theta_h^{n,j+1} \in W_h$ from the equations above. We find $c_h^{n,j+1} \in V_h$ such that

$$< \theta_h^{n,j+1}(c_h^{n,j+1} - c_h^{n-1}) + c_h^{n,j}(\theta_h^{n,j+1} - \theta_h^{n-1}), v_{2,h} > + \Delta t < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n,j+1}} c_h^{n,j+1}, \nabla v_{2,h} > \\ + \Delta t < R(c_h^{n,j}), v_{2,h} > + L_3 < c_h^{n,j+1} - c_h^{n,j}, v_{2,h} >= \Delta t < \mathbb{S}_2, v_{2,h} > \tag{30}$$

hold true for all $v_{2,h} \in V_h$.

Remark 2. *There exist multiple improved formulations of both the Newton method and L-scheme. We refer, among others, to the trust region techniques [Wang et al.-2013], and the modified L-scheme in [Mitra et al.-2019].*

Remark 3. *(Stopping criterion) For all schemes (monolithic or splitting), the iterations are stopped when,*

$$\left\|\Psi_h^{n,j+1} - \Psi_h^{n,j}\right\|_\infty \leq \epsilon_1, \qquad \left\|\theta_h^{n,j+1} - \theta_h^{n,j}\right\|_\infty \leq \epsilon_2 \qquad \textit{and} \qquad \left\|c_h^{n,j+1} - c_h^{n,j}\right\|_\infty \leq \epsilon_3,$$

where by $\|\cdot\|_\infty$ we mean the $L^\infty(\Omega)$ norm. Later on, for ease of presentation, we consider $\epsilon_1 = \epsilon_2 = \epsilon_3 = \epsilon$. The parameter ϵ will be defined in the numerical section.

2.2. Anderson acceleration

Although the L-scheme is robust and converges under mild restrictions, the convergence rate depends strongly on the linearization parameters. We refer to [List et al.-2016, Pop et al.-2004, Slodicka-2002] for the analysis in case of standard Richards equation, and to [Karpinski et al.-2017] for the nonstandard model. Tuning the parameters to obtain optimal results in terms of numbers of iterations and thus of computational times, can be tedious and time-consuming. The Anderson Acceleration (AA) is a powerful post-processing tool which can drastically reduce the numbers of iterations required by linearly convergent schemes, such as the L-scheme here investigated. In addiction, it reduces the need for finding close to optimal linearization parameters.

D. G. Anderson introduced the acceleration tool in 1965 [Anderson-1965], and since then it has been investigated in multiple works, to name a few [Both et al.-2019, Evans et al.-2020, Walker et al.-2011]. We recall here the definition of AA, presented in [Walker et al.-2011], formulated for a general fixed point problem, of the form: given $g : \mathbb{R}^n \to \mathbb{R}^n$, solve the system $x = g(x)$.

Opposed to utilize only the last iteration x_k, in the AA the new approximation is a linear combination of previously computed ones, see Algorithm 2. In the following, we denote by AA(m) the Anderson acceleration where $m + 1$ previously computed iterates are taken into account. With this, AA(0) is the non-accelerated formulation. As revealed in the test cases below, this technique can drastically reduce the number of iterations required by the L-scheme.

Algorithm 1 Classical Fixed-Point iteration

1: Given x_0
2: **for** $k = 0, 1,...$ until convergence **do**
3: $\quad x_{k+1} = g(x_k)$
4: **end for**

Algorithm 2 Anderson Acceleration AA(m)

1: Given x_0
2: **for** $k = 1, 2...$ until convergence **do**
3: $\quad$ Set $m_k = min\{m, k-1\}$
4: $\quad$ Define the matrix $F_k = (f_{k-m_k-1}, \cdots, f_{k-1})$, where $f_i = g(x_i) - x_i$
5: $\quad$ Find $\alpha \in \mathbb{R}^{m_k+1}$ that solves

$$\min_{\alpha=(\alpha_0,\cdots,\alpha_{m_k})^T} \|F_k\alpha\| \text{ s.t. } \sum_{i=0}^{m_k} \alpha_i = 1.$$

6: $\quad$ Define $x_k := \sum_{i=0}^{m_k} \alpha_i g(x_{k-m_k+i-1})$
7: **end for**

The original formulation presented in [Anderson-1965] allows a for more general step,

$$x_k := \beta_k \sum_{i=0}^{m_k} \alpha_i g(x_{k-m_k+i-1}) + (1-\beta_k) \sum_{i=0}^{m_k} \alpha_i x_{k-m_k+i-1},$$

for a user-defined tuning parameter $\beta_k \in (0, 1]$. We considered the simplified formulation, obtained with $\beta_k = 1$, because no improvements have been observed in the numerical results when using the extended one.

We remark that large values for the depth m can result in an instability of the solution algorithm. When implementing the Anderson acceleration, one has to tune this parameter properly. A small m could produce only a small reduction in the numbers of iterations; too large m could result in a non-converging algorithm [Fang et al.-2008].

Remark 4. *The definition of the nonlinear splitting solvers allows for different ways to apply the AA. We study three different loops: the coupling one and the linearizing ones, one for each set of equations. We apply the Anderson acceleration to each of them. Two different parameters, m and m_{lin}, are defined. The former is used for the AA on the coupling loop, the latter for the implementation on the linearization ones. The same m_{lin} will be used for the loop regarding the flow equations and for the one regarding the transport.*

3. Numerical examples

In the following, we consider four numerical examples with increasing complexity, based on a manufactured solution, and an example in which the boundary conditions drive the flow but no

manufactured solution is given. The first four will differ in the different values for γ, δ and $\tau(\theta)$ taken into account. We have implemented the models and solving schemes in MRST, a toolbox based on Matlab for the simulations of flow in porous media [Lie-2016]. We use the two point flux approximation, one of the most common spatial discretization techniques. We remark that the linearization schemes and solving algorithms do not depend on the particular choice of the spatial discretization, so one may apply these solvers to other methods as well, without any difficulty.

The domain is the unit square Ω and the final time taken into consideration is $T = 3$. The simulations are performed on regular meshes, consisting of squares with sides $dx = 1/10,\ 1/20$, and $1/40$. The time steps are $\Delta t = T/25,\ T/50$ and $T/100$. The L parameters, used in the L-scheme formulations, are $L_1 = L_2 = L_3 = 0.1$, if not specified otherwise. We took into consideration different values, but the aforementioned choice seems to produce a robust algorithm which required fewest iterations to achieve the convergence. For the ease of the presentation, we set the three L parameters equal to each other; one could define different values for each parameter, investigating even further the linearization of each equation. We avoided this due to the application of the AA. We will observe that the schemes can be drastically accelerated, even though the L parameters are not optimal.

The condition numbers, for the stiffness matrices resulting from the different solving algorithms are computed using the L^1 norm, and we here report the averaged values over the full simulation. A minus sign $(-)$, in the tables reporting on iterations and condition numbers, implies that the method failed to converge for the particular combination of the time step and mesh size. The tolerance ϵ used in the stopping criterion presented in Remark 3 is $\epsilon = 1e-6$. We always report the total numbers of iterations required by the full simulation, not the average number required by each time step. For the splitting solver, we present, separately, the condition numbers of both flow and transport equations. Furthermore, for the nonlinear splitting, the iterations are divided in two, the ones required by the flow equations and the ones for the transport. Finally, the condition numbers reported are obtained by averaging over the full simulations.

We apply the Anderson acceleration to each solving algorithm, always reporting the depths m and m_{lin} used. Once more, m_{lin} is the Anderson parameter used for the acceleration of the linearization loops regarding the flow and transport equations in the nonlinear splitting solvers.

Inspired by [Lunowa et al.-2020], the first four examples are constructed in such a way that the following is an exact solution:

$$\theta_m(x,y,t) = \begin{cases} 1 - \frac{1}{2}\cos((t_1(x,y) - t)^2) & \text{if} \quad t < t_1(x,y), \\ \frac{1}{2} & \text{if} \quad t_1(x,y) \le t \le t_2(x,y), \\ 1 - \frac{1}{2}\cos((t - t_2(x,y))^2) & \text{if} \quad t > t_2(x,y), \end{cases} \tag{31}$$

$$\Psi_m(x,y,t) = \begin{cases} -p_{cap}(\theta_m) + \tau(\theta_m)\partial_t\theta_m - \gamma & \text{if} \quad \partial_t\theta_m < -\delta, \\ -p_{cap}(\theta_m) + \tau(\theta_m)\partial_t\theta_m + \frac{\gamma}{\delta}\partial_t\theta_m & \text{if} \quad -\delta \le \partial_t\theta_m \le \delta, \\ -p_{cap}(\theta_m) + \tau(\theta_m)\partial_t\theta_m + \gamma & \text{if} \quad \partial_t\theta_m > \delta, \end{cases} \tag{32}$$

$$c_m(x,y,t) = x(x-1)y(y-1)t, \tag{33}$$

where $t_1(x,y) = xy$, $t_2(x,y) = xy + 2$. Once the manufactured water content is defined, one obtains the pressure by simply using the second equation in (5). The capillary pressure is expressed

as $p_{cap}(\theta, c) = 1 - \theta^2 - 0.1\ c^3$ and the conductivity as $K(\theta) = 1 + \theta^2$. Even though such a formulation may appear non-realistic, we are mainly interested in the nonlinearities. Furthermore, a nonlinear reaction term, $R(c) = c/(c+1)$, is taken into account in the transport equation and the diffusion/dispersion coefficient D is set equal to 1.

Given the analytical expressions above, we can easily define the initial conditions, the Dirichlet boundary conditions on the unit square and compute the source terms $\mathbb{S}_1$ and $\mathbb{S}_2$ such that Ψ_m, θ_m and c_m are solutions of the system. In particular, the initial concentration and water content are

$$\begin{aligned} c(x, y, 0) &= 0 && \text{on} \quad \Omega, \\ \theta(x, y, 0) &= 1 - \frac{1}{2}\cos\left(t_1(x, y)^2\right) && \text{on} \quad \Omega. \end{aligned}$$

We impose a zero concentration $c(x, y, t) = 0$ on the boundary of the domain. The remaining boundary conditions, concerning the pressure, are time dependent. One needs to compute t_1 and t_2 on every side of the unit square. Once the time intervals given by t_1 and t_2 are obtained, the pressure can easily be imposed. For example, on the left side $x = 0$, thus $t_1 = 0$ and $t_2 = 2$. The water content θ becomes

$$\theta(0, y, t) = \theta_{left}(t) = \begin{cases} \frac{1}{2} & \text{if} \quad 0 < t < 2, \\ 1 - \frac{1}{2}\cos((t-2)^2) & \text{if} \quad t \geq 2, \end{cases}$$

and the resulting pressure boundary condition is

$$\Psi_{left}(0, y, t) = \begin{cases} -p_{cap}(\theta_{left}) + \tau(\theta_{left})\partial_t\theta_{left} - \gamma & \text{if} \quad \partial_t\theta_{left} < -\delta, \\ -p_{cap}(\theta_{left}) + \tau(\theta_{left})\partial_t\theta_{left} + \frac{\gamma}{\delta}\partial_t\theta_{left} & \text{if} \quad -\delta \leq \partial_t\theta_{left} \leq \delta, \\ -p_{cap}(\theta_{left}) + \tau(\theta_{left})\partial_t\theta_{left} + \gamma & \text{if} \quad \partial_t\theta_{left} > \delta. \end{cases}$$

Analogously, one can compute the pressure boundary conditions on the remaining sides.

3.1. Example 1, $\gamma = 0$, $\tau(\theta) = 1$

In the first example we impose $\gamma = 0$, thus, the hysteresis effects are neglected but we include a dynamic effect by considering a constant $\tau(\theta) = 1$. We compare the different algorithms presented in Section 2.1, reporting in the Tables 1 and 2 the total numbers of iterations required by each algorithm, and the condition numbers of the systems associated with each scheme. In the former, we investigate a fixed time step size, $\Delta t = T/25$, in the latter a fixed mesh size, $dx = 1/10$. As expected, a finer mesh results in worse conditioned systems, while smaller time steps give better conditioned ones. Moreover, the total number of iterations is increasing as we reduce the time step; smaller Δt implies more time steps and thus more iterations.

The schemes based on the L-scheme appear to be better conditioned than those based on the Newton method. The result is coherent with the theory [Illiano et al.-2020a, List et al.-2016, Mitra et al.-2019, Pop et al.-2004, Slodicka-2002]. One could even improve the condition numbers by using larger L parameters. However, larger values would have also increased the total numbers of iterations.

The alternate splitting schemes are converging much faster than the nonlinear ones. It is also interesting to observe that the numbers of iterations, required by the alternate splitting schemes, are comparable with the ones associated with the monolithic solvers. In [Illiano et al.-2020a], we observed similar results when solving the models without hysteresis and dynamic effects.

We notice also some reduction in the number of iterations required by the L-schemes thanks to the Anderson acceleration. The results obtained for the non-accelerated L-schemes ($m = 0$) are already optimal in terms of numbers of iterations; thus, the improvement can only be minimal. We report the total number of iterations for the full simulation, but mention that on average, for each time step, the L-scheme requires only five or six iterations. This is already a remarkable result, achieved thanks to the optimal L parameters. Furthermore, the Newton solvers have resulted in being slower when combined with the AA. This is coherent with the theory where it has been observed that quadratically convergent schemes, cannot be improved and the resulting accelerated solvers appear slower [Evans et al.-2020].

	Monolithic		NonLinS			AltLinS		
	Newton		Newton			Newton		
dx	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
1/10	65	4.91e+02	56 - 50	1.82e+02	1.60e+02	66	1.83e+02	1.60e+02
1/20	69	2.22e+03	57 - 50	7.61e+02	6.33e+02	66	7.60e+02	6.3359e+02
1/40	70	1.11e+04	58 - 50	3.29e+03	2.52e+03	66	3.42e+03	2.5165e+03
	Newton	(AA m = 1)	Newton (AA $m = m_{lin} = 1$)			Newton (AA m = 1)		
dx	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
1/10	93	4.76e+02	65 - 50	1.82e+02	1.60e+02	74	1.82e+02	1.60e+02
1/20	98	2.10e+03	66 - 50	7.54e+02	6.33e+02	74	7.59e+02	6.34e+02
1/40	100	9.92e+03	69 - 50	3.28e+03	2.52e+03	76	3.36e+03	2.52e+03
	L-scheme		L-scheme			L-scheme		
dx	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
1/10	134	4.15e+02	117 - 116	1.43e+02	1.36e+02	140	1.62e+02	1.3717e+02
1/20	140	1.82e+03	119 - 115	6.95e+02	5.41e+02	144	7.37e+02	5.41e+02
1/40	146	8.52e+03	128 - 116	3.12e+03	2.15e+03	150	3.18e+03	2.15 e03
	L-scheme	(AA m = 1)	L-scheme (AA $m = m_{lin} = 1$)			L-scheme (AA m= 1)		
dx	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
1/10	127	4.14e+02	107 - 100	1.9690e+02	1.3681e+02	136	1.44e+02	1.33e+02
1/20	129	1.85e+03	111 - 100	7.9408e+02	5.4122e+02	142	7.13e+02	5.31e+02
1/40	130	8.80e+03	118 - 100	3.3661e+03	2.1522e+03	146	3.01e+03	2.02e+03

Table 1: Example 1: Total number of iterations and condition numbers for fixed $\Delta t = T/25$, and different dx. Here, $L_1 = L_2 = L_3 = 0.1$ and $m = m_{lin} = 1$.

	Monolithic		NonLinS			AltLinS		
	Newton		Newton			Newton		
Δt	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
T/25	65	4.91e+02	56 - 50	1.82e+02	1.60e+02	66	1.83e+02	1.60e+02
T/50	103	2.75e+02	98 - 50	1.23e+02	8.86e+01	99	1.24e+02	8.89e+01
T/100	186	1.97e+02	172 - 50	8.69e+01	4.73e+01	172	8.74e+01	4.74e+01
	Newton	(AA m = 1)	Newton (AA $m = m_{lin} = 1$)			Newton (AA m = 1)		
Δt	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
T/25	93	4.76e+02	65 - 50	1.82e+02	1.60e+02	74	1.82e+02	1.60e+02
T/50	137	2.75e+02	114 - 50	1.23e+02	8.86e+01	114	1.23e+02	8.89e+01
T/100	244	1.96e+02	201 - 100	8.59e+01	4.73e+01	200	8.69e+01	4.74e+01
	L-scheme		L-scheme			L-scheme		
Δt	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
T/25	134	4.16e+02	117 - 116	1.43e+02	1.36e+02	140	1.62e+02	1.3717e+02
T/50	219	2.39e+02	182 - 200	1.12e+02	7.51e+01	218	1.2460e+02	7.5203e+01
T/100	425	1.64e+02	346 - 400	8.35e+01	3.98e+01	438	8.2955e+01	3.9923e+01
	L-scheme	(AA m = 1)	L-scheme (AA $m = m_{lin} = 1$)			L-scheme (AA m = 1)		
Δt	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
T/25	127	4.14e+02	107 - 100	1.97e+02	1.37e+02	136	1.44e+02	1.33e+02
T/50	217	2.36e+02	192 - 160	1.25e+02	7.51e+01	238	1.14e+02	7.50e+01
T/100	387	1.66e+02	347 - 300	8.33e+01	3.98e+01	432	8.21e+01	3.91e+01

Table 2: Example 1: Total number of iterations and condition numbers for fixed $dx = 1/10$, and different Δt. Here $L_1 = L_2 = L_3 = 0.1$, $m = m_{lin} = 1$.

In Table 3 we present the numerical errors and the estimated order of convergence of the spatial discretization based on the successively refined meshes investigated. Given the manufactured solution Ψ_m, we compute the numerical error $e_\Psi = \|\Psi_m - \Psi_n\|$, where Ψ_n is the numerical pressure computed. Similarly, we can define e_θ and e_c. Furthermore, $e_{\Psi,1}$ is the numerical error obtained for the mesh size $dx = 1/10$ and $\Delta t = T/25$, $e_{\Psi,2}$ for $dx = 1/20$ and $\Delta t = T/50$, and finally $e_{\Psi,3}$ for $dx = 1/40$ and $\Delta t = T/100$. $EOC = \log\left(\frac{e_{\Psi,i}}{e_{\Psi,i+1}}\right)/\log(2)$ is the estimated order of convergence. These results are independent from the solving algorithm taken into account, only the discretization approach plays a role. In this case we use a TPFA which is known to have a order of convergence equal to 1, as also reported here in the table.

	e_1	EOC	e_2	EOC	e_3
Ψ	4.61e-02	0.98	2.33e-02	1.00	1.16e-02
θ	1.31e-02	0.97	6.71e-03	0.99	3.38e-03
c	6.24e-03	1.53	2.15e-03	1.32	8.60e-04

Table 3: Example 1: Numerical error and estimated order of convergence (EOC) of the discretization method.

In Table 4 we tested different values of m and L. We can observe that, for large L, the Monolithic L-Scheme, here investigated, requires more iterations than for smaller parameters. If many iterations are required to achieve the convergence at each time step, one can take in consideration larger m values. For $L = 0.1$, the optimal choice, in terms of numbers of iterations, is $m = 1$; larger values result in slightly slower schemes. For the largest L tested, $L = 2.3$, the optimal choice is $m = 2$. Such L value corresponds to the theoretical L, $\max_\theta\{\frac{\partial p_{cap}}{\partial \theta}\} \approx 2.3$. To ensure the monotone convergence of the scheme it has been proved, that the parameter chosen must be larger than the Lipschitz constant of the nonlinearity, in this case the capillary pressure as a function of θ (see

[List et al.-2016, Pop et al.-2004, Slodicka-2002]). We set L_1, L_2 and L_3 equal to the theoretical L computed for the capillary pressure.

We can conclude that it is possible to obtain significant improvements by investigating the AA and thus finding the appropriate depth m. In this work, we have focused more on individuating the optimal L parameters, as refining the AA can be done more easily. The depths used are, in fact, small natural numbers.

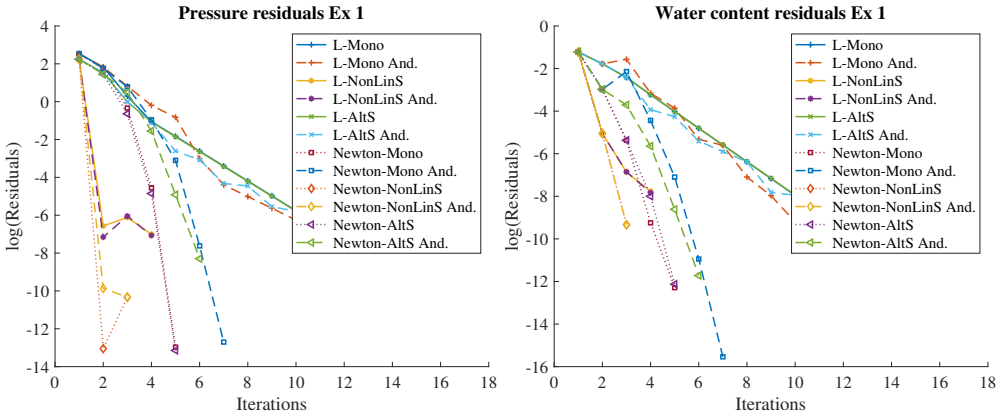
L	$m = 0$	$m = 1$	$m = 2$	$m = 3$	$m = 5$
.1	146	130	146	179	215
.5	247	161	152	172	200
1	411	202	168	165	195
2	711	290	199	206	217
2.3	810	317	207	215	227

Table 4: Example 1: Comparison of number of iterations for different m and L parameters for L-Mono.
Here, $dx = 1/40$, $\Delta t = T/25$ and $L_1 = L_2 = L_3 = L$.

Finally, we investigate the order of convergence of the linearization schemes. In Figure 5, we plot the residuals of pressure, water content and concentration, obtained at the final time step for the finest mesh size, $dx = 1/40$ and $\Delta t = T/25$. We can deduce the rates of convergence of the different linearization schemes. The L-schemes appear to be, as expected, linearly convergent in term of numbers of iterations. The AA improves the results only slightly, as already observed in Tables 1 and 2. This is justified by the fact that the L parameters (L_1, L_2, L_3) chosen here, appear to be optimal. The Newton methods are instead quadratically convergent. More precise results are observable in Table 5. Here we present the exact order of convergence of the different schemes. Given the residual of each unknown $(res_\Psi, res_\theta, res_c)$, at each iteration j, we compute the order of convergence as follow: $ORD_j = (\log(res_{j+1}/res_j))/(\log(res_j/res_{j-1}))$. For a fixed time step, we can average the orders obtained over the number of iterations required to achieve the convergence. We report below the values obtained by investigating the final time step; similar results have been observed for previous time steps.

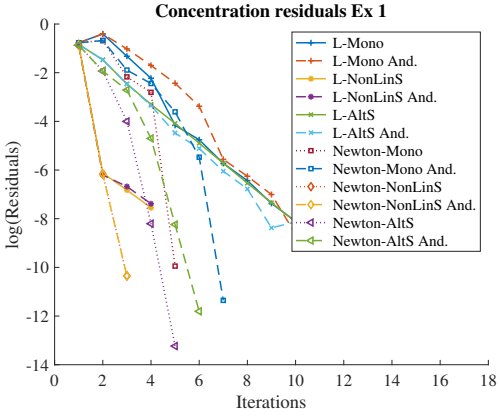
	LS-Mono	LS-Mono And.	LS-NonLinS	LS-NonLinS And.	LS-AltLinS	LS-AltLinS And.
Ψ	1.00	1.26	1.13	1.46	1.13	2.07
c	0.96	1.55	1.01	1.77	1.01	1.30
θ	1.01	1.40	0.89	1.01	0.94	1.32
	New.-Mono	New.-Mono And.	New.-NonLinS	New.-NonLinS And.	New.-AltLinS	New.-AltLinS And.
Ψ	2.03	1.61	1.94	1.86	2.04	2.03
c	2.41	2.01	1.61	1.41	1.85	1.46
θ	1.97	0.57	1.81	0.97	1.61	0.95

Table 5: Example 1: Estimated order of convergence for the different linearization schemes.



(a) Pressure residuals.

(b) Water content residuals.



(c) Concentration residuals.

Figure 5: Example 1: Residuals of each unknown at the final time step, for the different schemes.
Here, $L_1 = L_2 = L_3 = 0.1$, $m = m_{lin} = 1$, $dx = 1/40$, $\Delta t = T/25$.

Remark 5. *Since the Anderson acceleration with small depth is a cheap post-processing step, reducing the number of iterations directly reduces the CPU time almost proportionally.*

3.2. *Example 2, $\gamma = 0$, $\tau(\theta) = 1 + \theta^2$*

In the second example, the setup of the first is extended by adopting a nonlinear τ, precisely $\tau(\theta) = 1+\theta^2$. In Tables 6 and 7, we present the condition numbers and the required iteration counts associated with each solving algorithm.

The introduction of a nonlinear $\tau(\theta)$ increases the numbers of iterations required by each solver. The L-scheme is linearly convergent while the Newton method is quadratically convergent. Furthermore, the conclusions from Example 1 concerning the AA remain the same. In particular, we observe only small reductions in the numbers of iterations required by the L-schemes. Once more, this is justified by the optimal choice of the L parameters. We can observe that, for each time step, only a few iterations are required; thus, no further acceleration is expected.

	Monolithic		NonLinS			AltLinS		
	Newton		Newton			Newton		
dx	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
1/10	67	481.17	59 - 50	161.39	161.00	64	152.62	161.76
1/20	69	2.20e+03	57 - 50	656.26	636.33	66	644.98	644.99
1/40	70	1.10e+04	59 - 50	2.85e+03	2.53e+03	68	2.93e+03	2.53e+03
	Newton	(Anderson m = 1)	Newton (AA $m = m_{lin} = 1$)			Newton (Anderson m = 1)		
dx	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
1/10	94	467.04	69 - 50	160.36	161.00	76	152.45	161.94
1/20	98	2.07e+03	68 - 50	636.14	636.33	76	642.25	638.40
1/40	100	9.85e+03	71 - 50	2.80e+03	2.53e+03	80	2.87e+03	2.53e+03
	L-scheme		L-scheme			L-scheme		
dx	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
1/10	136	403.49	112 - 117	158.33	137.81	130	158.05	138.50
1/20	139	1.77e+03	120 - 116	644.85	544.34	136	651.07	545.85
1/40	144	8.34e+03	125 - 116	2.73e+03	2.16e+03	142	2.81e+03	2.16e+03
	L-scheme	(Anderson m = 1)	L-scheme (AA $m = m_{lin} = 1$)			L-scheme (AA m = 1)		
dx	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
1/10	129	401.61	106 - 100	158.87	137.74	132	156.89	138.21
1/20	131	1.78e+03	113 - 100	644.63	543.86	134	647.35	545.18
1/40	137	8.44e+03	117 - 100	2.76e+03	2.16e+03	142	2.78e+03	2.16e+03

Table 6: Example 2: Total number of iterations and condition numbers for fixed $\Delta t = T/25$, and different dx. Here, $L_1 = L_2 = L_3 = 0.1$ and $m = m_{lin} = 1$.

	Monolithic		NonLinS			AltLinS		
	Newton		Newton			Newton		
Δt	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
T/25	67	481.17	59 - 50	161.39	161.00	64	152.62	161.76
T/50	107	256.90	102 - 50	95.40	89.26	102	95.08	89.80
T/100	193	159.33	184 - 100	73.42	47.70	184	72.37	47.91
	Newton	(AA m = 1)	Newton (AA $m = m_{lin} = 1$)			Newton (AA m = 1)		
Δt	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
T/25	94	467.04	69 - 50	160.36	161.00	76	152.45	161.94
T/50	142	254.28	117 - 50	94.64	89.26	118	95.71	89.88
T/100	259	159.96	207 - 100	73.05	47.70	208	72.78	47.95
	L-scheme		L-scheme			L-scheme		
Δt	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
T/25	136	403.49	112 - 117	157.86	137.81	130	158.05	138.50
T/50	220	211.60	185 - 201	98.63	75.62	220	97.80	75.86
T/100	433	125.49	356 - 400	65.18	40.16	446	65.24	40.28
	L-scheme	(AA m = 1)	L-scheme (AA $m = m_{lin} = 1$)			L-scheme (AA m = 1)		
Δt	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
T/25	129	401.61	106 - 100	158.87	137.74	132	156.89	138.21
T/50	214	212.13	192 - 161	97.72	75.69	240	97.63	75.66
T/100	388	126.15	355 - 302	64.73	40.16	440	64.45	40.17

Table 7: Example 2: Total number of iterations and condition numbers for fixed $dx = 1/10$, and different Δt. Here $L_1 = L_2 = L_3 = 0.1$, $m = m_{lin} = 1$.

As for the results presented in Tables 6 and 7, the numerical errors and EOC reported in Table 8 are similar to the ones from the first example.

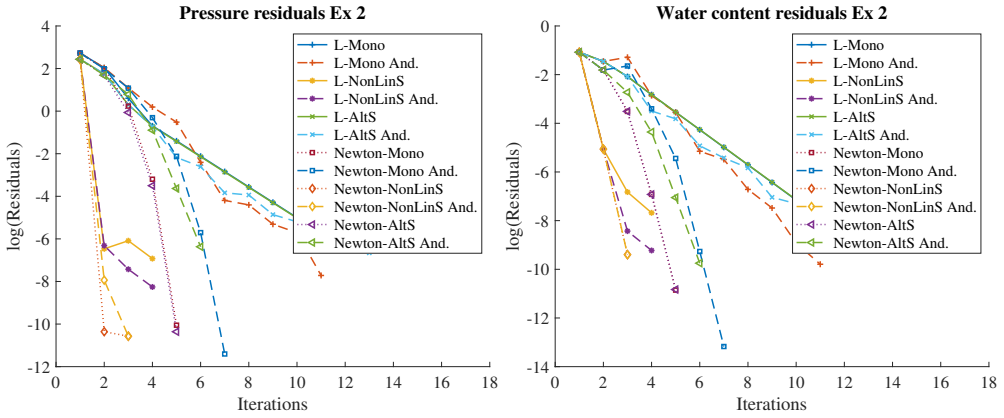
	e_1	EOC	e_2	EOC	e_3
Ψ	0.0759	0.9688	0.0388	0.9913	0.0195
θ	0.0140	0.9556	0.0072	0.9830	0.0036
c	0.0084	1.5198	0.0029	1.3037	0.0012

Table 8: Example 2: Numerical error and estimated order of convergence (EOC) of the discretization method.

In Figure 6, we report the residuals of the pressure, water content and concentration at the final time step. The L-schemes are linearly convergent, and applying the AA does not result in significant improvements. The convergence rates and number of iterations remain the same. Also for the Newton solvers, since they are quadratically convergent, the AA cannot improve this aspect. Table 9 presents the precise order of convergence of the different linearization schemes.

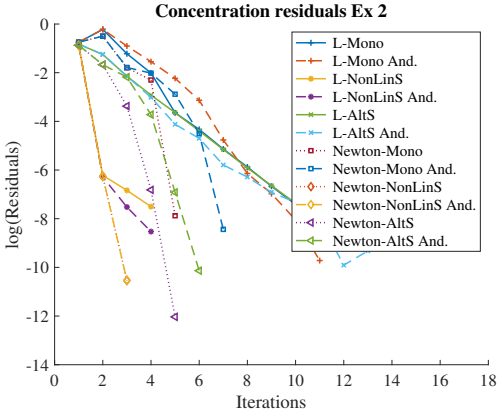
	LS-Mono	LS-Mono And.	LS-NonLinS	LS-NonLinS And.	LS-AltLinS	LS-AltLinS And.
Ψ	1.07	1.40	1.11	1.36	1.14	1.24
c	0.99	1.23	0.98	1.45	0.96	1.10
θ	1.03	1.15	0.97	1.25	0.93	0.98
	New.-Mono	New.-Mono And.	New.-NonLinS	New.-NonLinS And.	New.-AltLinS	New.-AltLinS And.
Ψ	1.61	1.58	1.97	1.69	2.15	2.14
c	2.68	1.47	1.98	1.54	1.98	1.83
θ	1.99	1.61	1.89	1.70	2.17	1.95

Table 9: Example 2: Estimated order of convergence for the different linearization schemes.



(a) Pressure residuals.

(b) Water content residuals.



(c) Concentration residuals.

Figure 6: Example 2: Residuals of each unknown at the final time step, for the different schemes. Here, $L_1 = L_2 = L_3 = 0.1$, $m = m_{lin} = 1$, $dx = 1/40$, $\Delta t = T/25$.

3.3. Example 3, $\gamma = 1$, $\delta = 5e-3$, $\tau(\theta) = 0$

With the same manufactured solutions, we now consider the case without dynamic effects ($\tau(\theta) = 0$), but include hysteresis by choosing $\gamma = 1$ and $\delta = 5e-3$.

From the results in Tables 10 and 11, we notice that the Newton method, in all its formulations, fails to converge. In Table 11, smaller time steps are taken, but no improvements are observable. A further reduction of the time step could have resulted in converging Newton solvers but the total numbers of iterations for the full simulation would have been larger than the ones required by the L-schemes on fewer but larger time steps. In contrast, the L-schemes are more robust and, even though requiring a higher number of iterations than previously, they converge. We take $L_1 = L_2 = L_3 = L = 1$, which appear to be the optimal choice in terms of numbers of iterations.

The AA improves the convergence of the L-schemes. This is the first example of this study in which the results obtained thanks to the AA are improved substantially. This is due to the presence of the hysteresis, requiring a large L for the overall convergence, and thus the total numbers of iterations is larger. On average, the monolithic L-scheme solver requires circa 18 iterations per time step. For $m = 1$, the AA reduces the iterations by circa 50%. Different m values have been tested but none of the ones investigated lead to the convergence of the Newton schemes. On all tests, Newton has failed to converge, whereas the L-schemes converged and the AA yields further improvement.

	Monolithic		NonLinS			AltLinS		
	Newton		Newton			Newton		
dx	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
1/10	-	-	-	-	-	-	-	-
1/20	-	-	-	-	-	-	-	-
1/40	-	-	-	-	-	-	-	-
	L-scheme		L-scheme			L-scheme		
dx	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
1/10	448	409.16	210 - 441	484.83	69.22	450	361.76	69.34
1/20	456	1.65e+03	266 - 439	1.98e+03	259.35	452	1.46e+03	260.02
1/40	468	6.62e+03	276 - 438	7.74e+03	996.36	460	5.88e+03	999.13
	L-scheme	(AA m = 2)	L-scheme (AA m = 2, m_{lin} = 5)			L-scheme (AA m = 1)		
dx	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
1/10	226	468.28	179 - 150	497.56	70.10	328	450.28	71.81
1/20	278	1.97e+03	187 - 141	2.03e+03	261.40	408	2.09e+03	269.72
1/40	303	8.24e+03	-	-	-	378	7.76e+03	967.29

Table 10: Example 3: Total number of iterations and condition numbers for fixed $\Delta t = T/25$, and different dx. Here, $L_1 = L_2 = L_3 = 1$, different m and m_{lin} are taken into account.

	Monolithic		NonLinS			AltLinS		
	Newton		Newton			Newton		
dx	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
T/25	-	-	-	-	-	-	-	-
T/50	-	-	-	-	-	-	-	-
T/100	-	-	-	-	-	-	-	-
	L-scheme		L-scheme			L-scheme		
				cond. #			cond. #	
Δt	# iterations	condition #	# iterations	Flow	Transport	# iterations	Flow	Transport
T/25	448	409.16	210 - 441	484.83	69.22	450	361.76	69.34
T/50	836	363.15	513 - 846	422.16	35.96	838	332.02	36.03
T/100	1787	395.55	1261 - 1597	428.24	18.91	1764	363.40	19.07
	L-scheme	(AA m = 1)	L-scheme (AA m = 2, m_{lin} = 5)			L-scheme (AA m = 1)		
dx	# iterations	condition #	# iterations	cond. # Flow	cond. # Transport	# iterations	cond. # Flow	cond. # Transport
T/25	226	468.28	179 - 150	497.56	70.10	328	450.28	71.81
T/50	533	410.93	346 - 316	504.00	36.85	664	442.54	37.43
T/100	1217	467.07	861 - 944 (m_{lin} = 1)	491.24	19.15	1842	473.32	19.75

Table 11: Example 3: Total number of iterations and condition numbers for fixed $dx = 1/10$, and different Δt. Here, $L_1 = L_2 = L_3 = 1$, different m and m_{lin} are taken into account

Once more, we report the numerical errors and the estimated orders of convergence associated with the discretization technique here implemented (TPFA). In Table 12, we present the values obtained for the monolithic L-scheme. The EOC depends only on the discretization technique, not the linearization scheme or solving algorithm.

In Figure 7, we report the residuals of pressure, water content and concentration, at the final time step. The differences between the accelerated and non-accelerated schemes seem to be

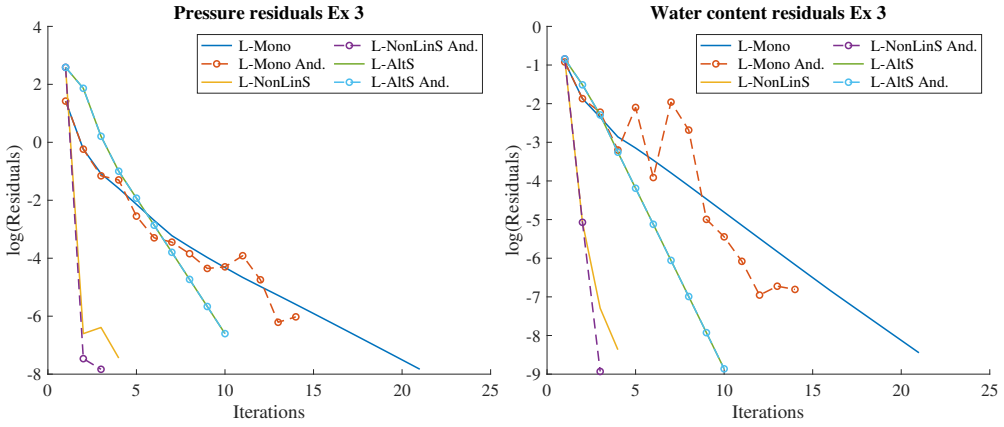
	e_1	EOC	e_2	EOC	e_3
Ψ	0.0308	1.0345	0.0150	0.9281	0.0085
θ	0.0350	1.2985	0.0142	1.0997	0.0066
c	0.0060	1.3202	0.0024	1.3000	0.0010

Table 12: Example 3: Numerical error and estimated order of convergence (EOC) of the discretization method.

minimal at the final time step but we observe in Tables 10 and 11 that the total improvements are actually substantial. The precise orders of convergence for the different solving algorithms are reported in Table 13. The non-accelerated L-schemes have an order of convergence equal to one, while the accelerated ones have slightly larger values. No result was reported for the Newton schemes due to the lack of convergence.

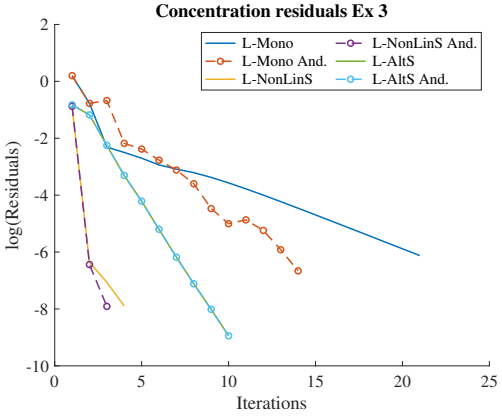
	LS-Mono	LS-Mono And.	LS-NonLinS	LS-NonLinS And.	LS-AltLinS	LS-AltLinS And.
Ψ	1.14	1.33	1.00	1.19	1.00	1.81
c	1.00	1.32	1.00	1.51	0.99	1.52
θ	1.00	1.29	1.00	1.39	1.00	1.16

Table 13: Example 3: Estimated order of convergence for the different linearization schemes.



(a) Pressure residuals.

(b) Water content residuals.



(c) Concentration residuals.

Figure 7: Example 3: residuals of each unknown at the final time step, for the different schemes.
Here, $L_1 = L_2 = L_3 = 1$, $dx = 1/40$, $\Delta t = T/25$ and $m \neq m_{lin}$.

3.4. Example 4, $\gamma = 1$, $\delta = 5e - 3$, $\tau(\theta) = 1 + \theta^2$

Finally, we study a problem which includes both hysteresis and dynamic capillary effects. We choose $\delta = 5e - 3$, $\gamma = 1$ and $\tau(\theta) = 1 + \theta^2$. As for the previous examples, we report the total numbers of iterations required by each algorithm, the condition numbers associated with the linearized equations, the EOC of the discretization technique and the residual for each unknown.

In Tables 14 and 15, we present the total number of iterations required by each algorithm, and the condition numbers associated with each system. As in the previous example, the Newton method fails to converge, while the L-scheme based solvers present no difficulties. The L parameters are all set equal to 0.1. This leads to a faster convergence, when compared to the

previous example, where larger values have been required for robustness. This is explained by the fact that, since the dynamic effects are introduced ($\tau > 0$), the solution is more regular [Cao et al.-2015, Mikelic-2010].

We have tested different values of m on the Newton methods, but none ensured the convergence of the schemes. As in the previous test cases, we have investigated smaller time steps, but the Newton solvers has still failed to converge.

Regarding the results obtained thanks to the AA, we can notice some improvements which are smaller than the ones observed for the previous test cases. Once more, this is due to the optimal choice of the L parameters, ensuring that the L-scheme converges, on average, in 5 iterations per time step. Therefore further improvements are not expected. Note that, compared to the first example (Table 4), larger L values are used leading to larger numbers of iterations. This explains why the AA with proper parameters m have improved the convergence behaviour of the L-scheme there.

	Monolithic		NonLinS			AltLinS		
	Newton		Newton			Newton		
				cond. #			cond. #	
dx	# iterations	condition #	# iterations	Flow	Transport	# iterations	Flow	Transport
1/10	-	-	-	-	-	-	-	-
1/20	-	-	-	-	-	-	-	-
1/40	-	-	-	-	-	-	-	-
	L-scheme		L-scheme			L-scheme		
				cond. #			cond. #	
dx	# iterations	condition #	# iterations	Flow	Transport	# iterations	Flow	Transport
1/10	152	290.13	128 - 122	208.09	162.46	251	206.24	174.77
1/20	160	768.54	137 - 121	558.01	621.77	259	486.23	598.01
1/40	165	3.04e+03	141 - 120	2.10e+03	2.35e+03	328	2.14e+03	2.4415e+03
	L-scheme	(AA m = 1)	L-scheme (AA m = 2, m_{lin} = 3)			L-scheme (AA m = 2)		
				cond. #			cond. #	
dx	# iterations	condition #	# iterations	Flow	Transport	# iterations	Flow	Transport
1/10	139	288.40	112 - 85	212.29	165.24	152	198.15	166.47
1/20	144	752.41	117 - 89	550.05	630.43	162	522.01	636.00
1/40	149	3.04e+03	127 - 88	2.19e+03	2.39e+03	166	2.06e+03	2.40e+03

Table 14: Example 4: Total number of iterations and condition numbers for fixed $\Delta t = T/25$, and different dx. Here, $L_1 = L_2 = L_3 = 0.1$ and $m \neq m_{lin}$.

	Monolithic		NonLinS			AltLinS		
	Newton		Newton			Newton		
				cond. #			cond. #	
Δt	# iterations	condition #	# iterations	Flow	Transport	# iterations	Flow	Transport
T/25	-	-	-	-	-	-	-	-
T/50	-	-	-	-	-	-	-	-
T/100	-	-	-	-	-	-	-	-
	L-scheme		L-scheme			L-scheme		
				cond. #			cond. #	
Δt	# iterations	condition #	# iterations	Flow	Transport	# iterations	Flow	Transport
T/25	152	290.13	128 - 122	208.09	162.46	251	206.24	174.77
T/50	248	310.93	201 - 225	264.09	89.64	424	263.91	97.09
T/100	508	415.80	403- 405	403.31	47.53	768	403.81	52.72
	L-scheme	(AA m = 1)	L-scheme (AA m = 2, m_{lin} = 3)			L-scheme (AA m = 2)		
				cond. #			cond. #	
Δt	# iterations	condition #	# iterations	Flow	Transport	# iterations	Flow	Transport
T/25	139	288.40	112 - 85	212.29	165.24	152	198.15	166.47
T/50	233	312.95	195 - 167	267.27	90.55	250	260.41	91.42
T/100	448	416.05	358 - 308	404.34	47.65	506	403.30	48.38

Table 15: Example 4: Total number of iterations and condition numbers for fixed $dx = 1/10$, and different Δt. Here $L_1 = L_2 = L_3 = 0.1$ and $m \neq m_{lin}$

The numerical errors and the estimated orders of convergence of the discretization technique (TPFA), presented in Table 16, are consistent with the ones from the previous test cases.

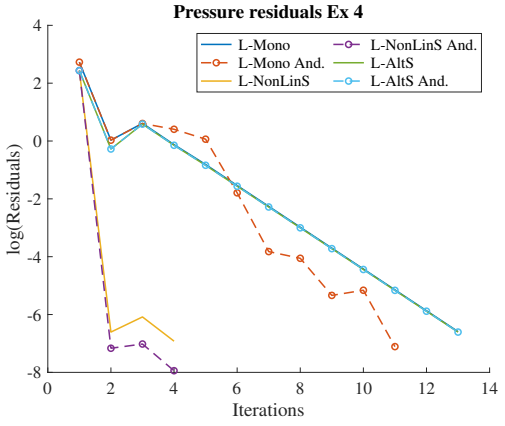
	e_1	EOC	e_2	EOC	e_3
Ψ	0.0759	0.9558	0.0391	0.8837	0.0212
θ	0.0138	0.9115	0.0073	0.9463	0.0038
c	0.0101	1.2531	0.0042	1.2655	0.0018

Table 16: Example 4: Numerical error and estimated order of convergence (EOC) of the discretization method.

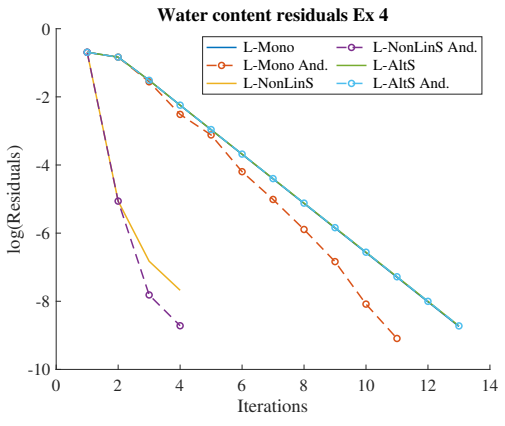
Finally, regarding the order of convergence of the different solving algorithms, Figure 8 presents the residuals for each unknown, and Table 17 the precise orders computed averaging over iterations at the final time step.

	LS-Mono	LS-Mono And.	LS-NonLinS	LS-NonLinS And.	LS-AltLinS	LS-AltLinS And.
Ψ	1.00	1.29	0.99	1.27	0.99	1.30
c	1.29	1.29	1.02	1.15	1.07	1.11
θ	1.00	1.16	1.00	1.35	1.00	1.20

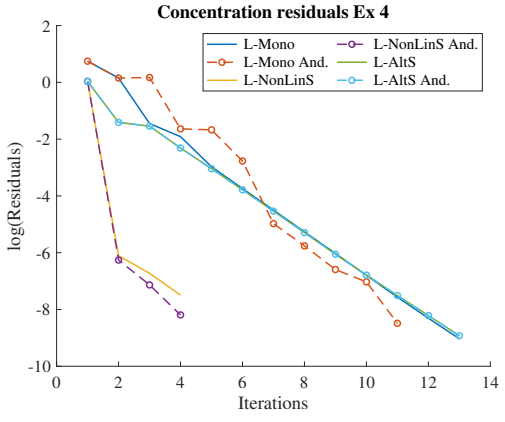
Table 17: Example 4: Estimated order of convergence for the different linearization schemes.



(a) Pressure residuals.



(b) Water content residuals.



(c) Concentration residuals.

Figure 8: Example 4: Residuals of each unknown at the final time step, for the different schemes.
Here, $L_1 = L_2 = L_3 = 0.1$, $dx = 1/40$, $\Delta t = T/25$ and different $m \neq m_{lin}$.

3.5. Physical example

As final numerical study, we investigate a test case that involves realistic parameters, but without having a manufactured solution. The flow will be driven by the boundary conditions. The domain Ω is the unit square and the final time is $T = 4$. The capillary pressure and conductivity expressions are given by the van Genuchten formulation [van Genuchten-1980], $K(\theta) = \theta_e^l\left(1-(1-\theta_e^{1/M})^M\right)^2$ and $p_{cap}(\theta, c) = (1 - b\ln(c/a + 1))^{-1}(-\theta^{-1/M})^{1-M}$,
where $\theta_e = (\theta - \theta_r)/(\theta_s - \theta_r)$ is the effective water content, $\theta_s = 0.9$, $\theta_r = 0.005$, $M = 2$, $l = 0.31$, $a = 0.04$ and $b = 0.47$. Furthermore, we take $\tau(\theta) = 1 + \theta^2$, and the hysteresis effects are included

by setting $\gamma = 1$ and $\delta = 5e - 3$, as in Example 4.

Dirichlet boundary conditions are imposed at the left side of the unit square

$$\Psi|_{x=0} = 1 + \begin{cases} 0.5t & \text{if } t < 1, \\ 0.5 & \text{if } 1 \le t < 2, \\ 0.5(3-t) & \text{if } 2 \le t < 3, \\ -0.4 & \text{if } 3 \le t \le 4, \end{cases}$$

$$c|_{x=0} = 2,$$

whereas, at the remaining sides, we consider homogeneous Neumann boundary conditions. The discontinuity in time $t = 3$ makes solving the problem numerically even more complex. The initial conditions are

$$\theta^0 = x, \qquad \text{and} \qquad c^0 = 1.$$

All L parameters are set to 0.5. We have tested different values, but, $L = 0.5$ seems to give the best results in terms of numbers of iterations. Furthermore, the results may be improved even further by choosing different values for each parameter, L_1, L_2 and L_3, but for ease of presentation this has been omitted here.

In Tables 18 and 19, we report the total numbers of iterations and condition numbers associated to each algorithm. We observe that, due to the higher nonlinearities of the conductivity K and capillary pressure p_{cap} involved, the results are different compared to the ones presented for the previous examples.

Again, the Newton solvers have failed to converge and the systems associated with the linearized equations are badly conditioned. Considering smaller time steps did not resolve this.

The L-schemes on the other hand converge, but require high numbers of iterations. We observe a significant improvement thanks to the AA. The performance of the monolithic solver is for example drastically improved, the iterations required are reduced by circa 50%. In case of finer meshes, one needs to use a larger L, precisely $L = 1$. The AA can introduce some instabilities, and thus a larger L may be required. Clearly, this leads to an increase in the number of iterations. Such results are still better than the one obtained for smaller L without acceleration. Similar observations can also be made for the splitting solvers. Even though larger L parameters may be required, the accelerated schemes perform better then the non-accelerated ones. The nonlinear splitting seems to be less stable than the alternate linear one. For a coarse mesh, we could use a large m, $m = 5$, resulting in an extreme reduction in the numbers of iterations. For finer meshes, m had to be set equal to 1, otherwise the schemes did not converge, and a larger L parameter was required. The alternate linearized splitting seems to be more stable, as the L-scheme linearization acts as stabilization term for both the nonlinearities and the decoupling. Unfortunately, for the first time, the results are slower than for the nonlinear splitting. The main differences are observable in the second table, Table 19. The mesh size is fixed, $dx = 1/10$, and the nonlinear solvers converge even for $m = 5$. The large AA parameter m ensures a strong reduction in the numbers of iterations. The alternate linearized splitting converges only for $m = 1$, anyhow the improvements are remarkable.

	Monolithic		NonLinS			AltLinS		
	Newton		Newton			Newton		
					cond. #			cond. #
dx	# iterations	condition #	# iterations	Flow	Transport	# iterations	Flow	Transport
1/10	-	-	-	-	-	-	-	-
1/20	-	-	-	-	-	-	-	-
1/40	-	-	-	-	-	-	-	-
	L-scheme		L-scheme			L-scheme		
					cond. #			cond. #
dx	# iterations	condition #	# iterations	Flow	Transport	# iterations	Flow	Transport
1/10	899	702.16	1522 - 235	790.37	155.18	1490	623.78	155.10
1/20	930	3.58e+03	1515 - 240	3.32e+03	615.72	1428	2.54e+03	596.60
1/40	941	1.52e+04	1680 - 243	1.44e+04	2.46e+03	1548	1.24e+04	2.45e+03
	L-scheme	(AA m = 1)	L-scheme (AA m = 5, m_{lin} = 1)			L-scheme (AA m = 1)		
					cond. #			cond. #
dx	# iterations	condition #	# iterations	Flow	Transport	# iterations	Flow	Transport
1/10	480	1.37e+03	210 - 96	811.29	155.16	795	774.00	155.18
1/20	541 (L = 1)	5.27e+03	758 - 170 (m = 1, L = 1)	2.76e+03	395.74	532	2.72e+03	468.04
1/40	603 (L = 1)	2.29e+04	1214 - 261 (m = 1, L = 2)	8.84e+03	911.31	1798 (L_2 = 2)	1.89e+04	2.46e+03

Table 18: Example 5: Total number of iterations and condition numbers for fixed $\Delta t = T/25$, and different dx. Here, $L_1 = L_2 = L_3 = 0.5$ and $m \neq m_{lin}$.

	Monolithic		NonLinS			AltLinS		
	Newton		Newton			Newton		
					cond. #			cond. #
dx	# iterations	condition #	# iterations	Flow	Transport	# iterations	Flow	Transport
T/25	-	-	-	-	-	-	-	-
T/50	-	-	-	-	-	-	-	-
T/100	-	-	-	-	-	-	-	-
	L-scheme		L-scheme			L-scheme		
					cond. #			cond. #
Δt	# iterations	condition #	# iterations	Flow	Transport	# iterations	Flow	Transport
T/25	899	702.16	1522 - 235	790.37	155.18	1490	623.78	155.10
T/50	2892	908.39	5218 - 450	747.79	84.75	4942	533.58	84.83
T/100	9809	707.12	18261 - 849	692.77	46.72	17406	485.71	46.84
	L-scheme	(AA m =1)	L-scheme (AA m = 5, m_{lin} = 1)			L-scheme (AA m = 1)		
					cond. #			cond. #
Δt	# iterations	condition #	# iterations	Flow	Transport	# iterations	Flow	Transport
T/25	480	1.37e+03	210 - 96	811.29	155.16	795	774.00	155.18
T/50	897	1.65e+03	517 - 214	618.31	53.28	1784	775.76	84.77
T/100	2106	1.04e+03	1354 - 424	576.54	28.62	4928	1.05e+03	46.73

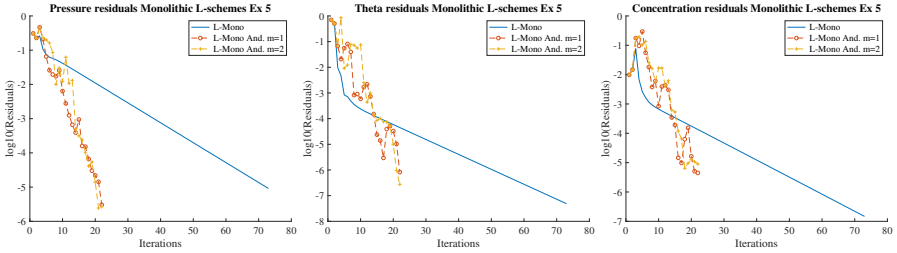
Table 19: Example 5: Total number of iterations and condition numbers for fixed $dx = 1/20$, and different Δt. Here, $L_1 = L_2 = L_3 = 0.5$ and $m \neq m_{lin}$.

It is also interesting to notice that, for decreasing time steps, the number of L iterations per time step is increasing. This is coherent with the theory. With the AA, this is mitigated; the average number of iterations remains more stable.

In Table 20, we report the different m and m_{lin} values investigated for the AA. We observe that for the L-scheme Mono solver, the optimal choice, in terms of numbers of iterations is $m = 1$. For the nonlinear splitting solver, only one value of m_{lin} has been taken in consideration for the coupling loop, precisely $m_{lin} = 1$. This is justified by the fact that the majority of the iterations have taken place in the inside loops, the nonlinearities of the equations are playing a larger role than the coupling aspect.

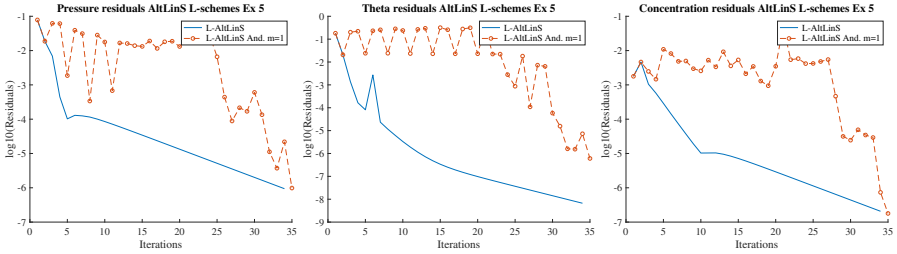
	# it. m=0	# it. m=1	# it. m=2	# it. m=5
Mono L-scheme	899	480	775	–
NonLinS L-scheme	1522 - 325 (m_{lin}=0)	322 - 118 (m_{lin}=1)	232 - 96 (m_{lin}=1)	210 - 96 (m_{lin}=1)
AltLinS L-scheme	1490	795	–	–
Newton-Mono	–	–	–	–

Table 20: Example 5: Numbers of iterations associated to different m and m_{lin} values, $dx = 1/10$ $dt = T/25$.



(a) Pressure residuals. (b) Water content residuals. (c) Concentration residuals.

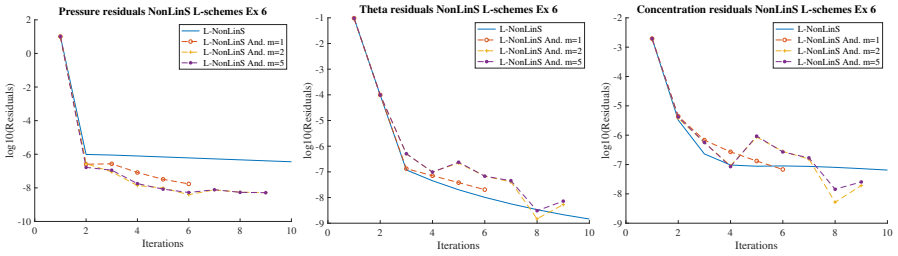
Figure 9: Example 5: Residuals of each unknown at the final time step, monolithic L-scheme. Here, $L_1 = L_2 = L_3 = 0.1$, different m are tested, $dx = 1/10$, and $\Delta t = T/25$.



(a) Pressure residuals. (b) Water content residuals. (c) Concentration residuals.

Figure 10: Example 5: Residuals of each unknown at the final time step, AltLinS L-scheme. Here, $L_1 = L_2 = L_3 = 0.1$, different m are tested, $m_{lin} = 1$, $dx = 1/40$, and $\Delta t = T/25$.

Finally, in the Figures 9, 10 and 11 we report the residuals at the final time step, for each unknown and each algorithm, and for different values of m. The results are coherent with the ones presented in the Tables 18 and 19. The monolithic solver shows a clear improvement thanks to the AA, for both $m = 1$ and $m = 2$. For the alternate linearized splitting solver, the AA does not



(a) Pressure residuals. (b) Water content residuals. (c) Concentration residuals.

Figure 11: Example 5: Residuals of each unknown at the final time step, NonLinS L-scheme. Here, $L_1 = L_2 = L_3 = 0.1$, different m are tested , $dx = 1/40$, and $\Delta t = T/25$.

seem to produce any improvement, as the rates of convergence seem to be worse. The result does not directly contradict the ones presented in the Tables 18 and 19; it simply states that, at the final time step, the AA does not produce any improvement. On the full simulation, we could observe a clear reduction in the numbers of iterations. For the nonlinear splitting, we can observe some improvements but once more they are not as evident as for the monolithic solvers. We already observed that both splitting solvers became unstable once the AA was applied, either requiring a larger L or a smaller m.

Table 21 presents the precise rates of convergence of the different linearization schemes. Once more we can observe as the AA improves the rates of convergence of the solvers based on the L-scheme.

	LS-Mono	LS-Mono And.	LS-NonLinS	LS-NonLinS And.	LS-AltLinS	LS-AltLinS And.
Ψ	0.94	1.30	0.98	1.87	0.98	1.21
c	1.01	1.78	0.95	1.42	0.93	1.80
θ	0.95	1.34	0.99	1.35	0.95	1.02

Table 21: Example 5: Order of convergence of the linearization schemes.

4. Conclusions

We consider models for flow and reactive transport in a porous medium. Next, to account for the influence of the solute concentration on the flow parameters, we incorporate effects like dynamic capillary pressure and hysteresis. The problem results being fully coupled.

For solving the time discrete equations (9), obtained after applying the Euler implicit scheme, we investigate different approaches: a monolithic solution algorithm and two splitting ones. Furthermore, for solving the nonlinear problem, two linearizations are studied: the Newton method and the L-scheme. The latter appears to be more stable than the former, which is more commonly implemented.

Finally, we have studied the effects of the Anderson acceleration. We observed that its implementation is particularly simple and can result in significant improvements. There were cases in which the differences between the accelerated and non-accelerated schemes were minimal, but due to its simplicity and the possibility of the great reduction in the numbers of iterations, we think it should always be tested. Particularly, one can either invest time in finding the optimal L parameters or the best depth m for which the AA results in the fastest scheme. Often, finding the most suitable m is simpler, and it can results in impressive improvements.

Acknowledgments. *The research of D. Illiano was funded by VISTA, a collaboration between the NorwegianAcademy of Science and Letters and Equinor, project number 6367, project name: adaptive model and solver simulation of enhanced oil recovery. The research of J.W. Both was supported by the Research Council of Norway Project 250223, as well as the FracFlow project funded by Equinor through Akademiaavtalen. The research of I.S. Pop was supported by the Research Foundation-Flanders (FWO), Belgium through the Odysseus programme (project G0G1316N) and Equinor through the Akademia grant.*

References

[Aavatsmark-2001] I. Aavatsmark, *An introduction to multipoint flux approximations for quadrilateral grids, Computational Geosciences Volume 6, Issue 3-4, Pages 405-432*, 2002.

[Abreu et al.-217] E. Abreu, J. Vieira, *Computing numerical solutions of the pseudo-parabolic Buckley Leverett equation with dynamic capillary pressure, Mathematics and Computers in Simulation, Volume 137, Pages 29-48*, 2017.

[Abreu et al.-2020] E. Abreu, P. Ferraz, J. Vieira, *Numerical resolution of a pseudo-parabolic Buckley-Leverett model with gravity and dynamic capillary pressure in heterogeneous porous media, Journal of computational Physics, Volume 411*, 2020.

[Agosti et al.-2015] A. Agosti, L. Formaggia, A. Scotti, *Analysis of a model for precipitation and dissolution coupled with a Darcy flux, Journal of Mathematical Analysis and Applications, Volume 431, Issue 2, Pages 752-781*, 2015.

[Albuja et al.-2021] G. Albuja, A. I. Avila, *A family of new globally convergent linearization schemes for solving Richards' equation, Applied Numerical Mathematics, Volume 159, Pages 281-296*, 2021.

[Alt et al.-1983] W. Alt, H. Luckhaus, *Quasilinear elliptic-parabolic differential equations, Mathematische Zeitschrift, Volume 183, Issue 3, Pages 311-341*, 1983.

[Anderson-1965] D. G. Anderson, *Iterative Procedures for Nonlinear Integral Equations, Journal of the ACM, Volume 12, Issue 4, Pages 547-560*, 1965.

[Arbogast-1996] T. Arbogast, M. F. Wheeler, *A nonlinear mixed finite element method for a degenerate parabolic equation arising in flow in porous media, SIAM Journal on Numerical Analysis, Volume 33, Issue 4, Pages 1669-1687*, 1996.

[Arnold et al.-2006] D. N. Arnold, F. Brezzi, B. Cockburn, L. D. Marini, *Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems, SIAM Journal on Numerical Analysis, Volume 39, Issue 5, Pages 1749-1779*, 2006.

[Arraras et al.-2020] A. Arraras, F. J. Gaspar, L. Portero, C. Rodrigo, *Multigrid solvers for multipoint flux approximations of the Darcy problem on rough quadrilateral grids, Computational Geosciences, https://doi.org/10.1007/s10596-020-09979-w,* 2020.

[Barrett et al.-1997] J. W. Barrett, P. Knabner, *Finite element approximation of the transport of reactive solutes in porous media. Part 1: error estimates for nonequilibrium adsorption processes, SIAM Journal on Numerical Analysis, Volume 34, Issue 1, Pages 201-227*, 1997.

[Bause et al.-2010] M. Bause, J. Hoffmann, P. Knabner, *First-order convergence of multi-point flux approximation on triangular grids and comparison with mixed finite element methods, Numerische Mathematik, Volume 116, Issue 1, Pages 1-29*, 2010.

[Beliaev et al.-2001] A.Y. Beliaev, S.M. Hassanizadeh, *A Theoretical Model of Hysteresis and Dynamic Effects in the Capillary Relation for Two-phase Flow in Porous Media. Transport in Porous Media, Volume 43, Issue 3, Pages 487-510*, 2001.

[Berardi et al.-2018] M. Berardi, F. Difonzo, M. Vurro, L. Lopez, *The 1D Richards' equation in two layered soils: a Filippov approach to treat discontinuities, Advances in Water Resources, Volume 115, Pages 264-272*, 2018.

[Berardi et al.-2020] M. Berardi, F. Difonzo, L. Lopez, *A mixed MoL-TMoL for the numerical solution of the 2D Richards' equation in layered soils, Computers & Mathematics with Applications Volume 79, Issue 7, Pages 1990-2001*, 2020.

[Both et al.-2019] J. W. Both, K. Kumar, J. M. Nordbotten, F. A. Radu, *Anderson accelerated fixed-stress splitting schemes for consolidation of unsaturated porous media, Computers & Mathematics with Applications, Volume 77, Issue 6, Pages 1479-1502*, 2019.

[Bottero et al.-2011] S. Bottero, S. M. Hassanizadeh, P. J. Kleingeld, T. J. Heimovaara, *Nonequilibrium capillarity effects in two-phase flow through porous media at different scales, Water Resources Research, Volume 47, Issue 10*, 2011.

[Camps-Roach et al.-2010] G. Camps-Roach, D. M. O'Carroll, T. A. Newton, T. Sakaki, T. H. Illangasekare, *Experimental investigation of dynamic effects in capillary pressure: Grain size dependency and upscaling, Water Resources Research, Volume 46, Issue 8,* 2010.

[Cao et al.-2015] X. Cao, I. S. Pop, *Two-phase porous media flows with dynamic capillary effects and hysteresis: Uniqueness of weak solutions, Computers & Mathematics with Applications, Volume 69, Issue 7, Pages 688-695*, 2015.

[Cao et al.-2016] X. Cao, I. S. Pop, *Degenerate two-phase porous media flow model with dynamic capillarity, Journal of Differential Equations Volume 260, Issue 3, Pages 2418-2456,* 2016.

[Cao et al.-2019a] X. Cao, K. Mitra, *Error estimates for a mixed finite element discretization of a two-phase porous media flow model with dynamic capillarity, Journal of Computational and Applied Mathematics, Volume 353, Pages 164 - 178*, 2019.

[Cao et al.-2019b] X. Cao, S.F. Nemadjieu, I.S. Pop, *Convergence of an MPFA finite volume scheme for two phase porous media flow with dynamic capillarity, IMA Journal of Numerical Analysis, Volume 39, Issue 1, Pages 512–544,* 2019.

[O'Carroll et al.-2005] D. M. O'Carroll, T. J. Phelan, L. M. Abriola, *Exploring dynamic effects in capillary pressure in multistep outflow experiments, Water Resources Research, Volume 41, Issue 11*, 2005.

[Celia et al.-1990] M. Celia, E. Bouloutas, R. L. Zarba, *A General Mass-Conservative Numerical Solution for the Unsaturated Flow Equation, Advances in Water Resources, Volume 26, Issue 7, Pages 1483-1496*, 1990.

[DiCarlo-2004] D. A. DiCarlo, *Experimental measurements of saturation overshoot on infiltration, Water Resources Research, Volume 40, Issue 4*, 2004.

[Dolejsi et al.2019] V. Dolejsi, M. Kuraz, P. Solin, *Adaptive higher-order space-time discontinuous Galerkin method for the computer simulation of variably-saturated porous media flows, Applied Mathematical Modelling, Volume 72, Pages 276-305*, 2019.

[Eymard et al.-1999] R. Eymard, M. Gutnic, D. Hilhorst, *The finite volume method for Richards equation, Computational Geosciences, Volume 3, Issue 3-4, Pages 256-294*, 1999.

[Evans et al.-2020] C. Evans, S. Pollock, L. G. Rebholz, M. Xiao, *A proof that Anderson acceleration improves the convergence rate in linearly converging fixed point methods (but not in those converging quadratically), SIAM Journal on Numerical Analysis, Volume 58, Issue 1, Pages 788-810*, 2020.

[Fang et al.-2008] H. Fang, Y. Saad, *Two classes of multisecant methods for nonlinear acceleration, Numerical Linear Algebra with Application, Volume 16, Issue 3, Pages 197 - 221*, 2008.

[Gray et al.-1998] W. G. Gray, S. M. Hassanizadeh, *Macroscale continuum mechanics for multiphase porous-*

media flow including phases, interfaces, common lines and common points, Volume 21, Issue 4, Pages 261 - 281, 1998.

[van Genuchten-1980] M. van Genuchten, *A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils, Soil Science Society of America Journal, Volume 44, Issue 5, Pages 892-898*, 1980.

[Helmig-1997] R. Helmig, *two-phase flow and transport processes in the subsurface: a contribution to the modeling of hydrosystems, Springer-Verlag*, 1997.

[Henry et al.-1999] E. J. Henry, J. E. Smith, A. W. Warrick, *Solubility effects on surfactant-induced unsaturated flow through porous media, Journal of Hydrology, Volume 223, Issues 3-4, Pages 164-174*, 1999.

[Hoa et al.-1977] N. T. Hoa, R. Gaudu, C. Thirriot, *Influence of the hysteresis effect on transient flows in saturated-unsaturated porous media, Water Resources Research, Volume 13, Issue 6, Pages 992-996*, 1977.

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

[Illiano et al.-2020a] D. Illiano, I. S. Pop, F. A. Radu, *Iterative schemes for surfactant transport in porous media, Computational Geosciences, https://doi.org/10.1007/s10596-020-09949-2*, 2020.

[Illiano et al.-2020b] D. Illiano, I. S. Pop, F. A. Radu, *An efficient numerical scheme for fully coupled flow and reactive transport in variably saturated porous media including dynamic capillary effects, accepted author manuscript, Numerical Mathematics and Advanced Applications ENUMATH 2019, Lecture Notes in Computational Science and Engineering 139, https://doi.org10.1007/978-3-030-55874-1_55*, 2020.

[Karagunduz et al.-2015] A. Karagunduz, M. H. Young, K. D. Pennell, *Influence of surfactants on unsaturated water flow and solute transport, Water Resources Research, Volume 51, Issue 4, Pages 1977-1988*, 2015.

[Karpinski et al.-2017] S. Karpinski, I. S. Pop, F. A. Radu, *Analysis of a linearization scheme for an interior penalty discontinuous Galerkin method for two-phase flow in porous media with dynamic capillarity effects, International Journal for Numerical Methods in Engineering, Volume 112, Issue 6, Pages 553-577*, 2017.

[Klausen et al.-2008] R. A. Klausen, F. A. Radu, G. T. Eigestad, *Convergence of MPFA on triangulations and for Richards' equation, International Journal for Numerical Methods in Fluids, Volume 58, Issue 12, Pages 1327-1351*, 2008.

[Koch et al.-2012] J. Koch, A. Ratz, B. Schweizer, *Two-phase flow equations with a dynamic capillary pressure, European Journal of Applied Mathematics, Volume 24, Issue 1*, 2012.

[Knabner et al.-2003] P. Knabner, S. Bitterlich, R. I. Teran, A. Prechtel, E. Schneid, *Influence of Surfactants on Spreading of Contaminants and Soil Remediation, Jager W., Krebs HJ. (eds) Mathematics — Key Technology for the Future. Springer, https://doi.org/10.1007/978-3-642-55753-8-12*, 2003.

[Li et al.-2007] H. Li, M. W. Farthing, C. N. Dawson, C. T. Miller, *Local discontinuous Galerkin approximations to Richards' equation, Advances in Water Resources, Volume 30, Issue 3, Pages 555-575*, 2007.

[Lie-2016] K.-A. Lie, *An Introduction to Reservoir Simulation Using MATLAB: User guide for the Matlab Reservoir Simulation Toolbox (MRST), SINTEF ICT*, 2016.

[List et al.-2016] F. List, F. A. Radu, *A study on iterative methods for solving Richards' equation, Computational Geoscience, Volume 20, Issue 2, Pages 341-353*, 2016.

[Lunowa et al.-2020] S. B. Lunowa, I. S. Pop, B. Koren, *Linearized domain decomposition methods for two-phase porous media flow models involving dynamic capillarity and hysteresis, Computer Methods in Applied Mechanics and Engineering, Volume 372, https://doi.org/10.1016/j.cma.2020.113364*, 2020.

[Mikelic-2010] A. Mikelic, *A global existence result for the equations describing unsaturated flow in porous media with dynamic capillary pressure, Journal of Differential Equations, Volume 248, Issue*

6, Pages 1561-1577, 2010.

[Milisic -2018] J.-P. Milisic, *The unsaturated flow in porous media with dynamic capillary pressure, Journal of Differential Equations, Volume 264, Issue 9, Pages 5629-5658*, 2018.

[Mitra et al.-2019] K. Mitra, I. S. Pop, *A modified L-scheme to solve nonlinear diffusion problems, Computers & Mathematics with Applications, Volume 77, Issue 6, Pages 1722-1738*, 2019.

[McClure et al.-2018] J. McClure, R. T. Armstrong, M. Berrill, S. Schluter, S. Berg, W. G. Gray, C. T. Miller, *Geometric state function for two-fluid flow in porous media, Physical Review Fluids, Volume 3, Issue 8*, 2018.

[Morrow et al.-2005] N. R. Morrow, C. C. Harris, *Exploring dynamic effects in capillary pressure in multistep outflow experiments, Water Resources Research, Volume 41, Issue 11*, 2005

[Nocheto et al.-1988] R. Nochetto, C. Verdi, *Approximation of degenerate parabolic problems using numerical integration, SIAM Journal on Numerical Analysis, Volume 25, Issue 4, Pages 784-814*, 1988.

[Oung et al.-2005] O, Oung, S. M. Hassanizadeh, A. Bezuijen, *Two-phase flow experiments in a geocentrifuge and the significance of dynamic capillary pressure effect, Journal of Porous Media, Volume 8, Issue 3, Pages 247-257*, 2005.

[Paniconi et al.-1994] C. Paniconi, M. Putti, *A comparison of Picard and Newton iteration in the numerical solution of multidimensional variably saturated flow problems, Water Resources Research, Volume 30, Issue 12, Pages 3357-3374*, 1994.

[Peszynska et al.-2008] M. Peszynska, S. Y. Yi, *Numerical methods for unsaturated flow with dynamic capillary pressure in heterogeneous porous media, International journal of Numerical Analysis and Modeling, Volume 5, Pages 126-149*, 2008.

[Prechtel et al.-2002] A. Prechtel, P. Knabner, *Accurate and efficient simulation of coupled water flow and nonlinear reactive transport in the saturated and vadose zone - application to surfactant enhanced and intrinsic bioremediation, International Journal of Water Resources Development, Volume 47, Pages 687-694*, 2002.

[Pop et al.-2004] I. S. Pop, F. A. Radu, P. Knabner, *Mixed finite elements for the Richards' equation: linearization procedure, Journal of Computational and Applied Mathematics, Volume 168, Issue 1, Pages 365-373*, 2004.

[Radu et al.-2010] F. A. Radu, I. S. Pop, S. Attinger, *Analysis of an Euler implicit, mixed finite element scheme for reactive solute transport in porous media, Numerical Methods for Partial Differential Equations, Volume 26, Issue 2, Pages 320-344*, 2010.

[Radu et al.-2013] F. A. Radu, A. Muntean, I. S. Pop, N. Suciu, O. Kolditz, *A mixed finite element discretization scheme for a concrete carbonation model with concentration-dependent porosity, Journal of Computational and Applied Mathematics Volume 246, Pages 74-85,* 2013.

[Russell et al.-1983] T. F. Russell, M. F. Wheeler, *Finite element and finite difference methods for continuous flows in porous media, SIAM, Pages 35-106*, 1983.

[Schweizer-2012] B. Schweizer, *The Richards equation with hysteresis and degenerate capillary pressure, Journal of Differential Equations, Volume 252, Issue 10, Pages 5594-5612*, 2012.

[Slodicka-2002] M. Slodicka, *A robust and efficient linearization scheme for doubly nonlinear and degenerate parabolic problems arising in flow in porous media, SIAM Journal on Numerical Analysis, Volume 23, Issue 5, Pages 1593-1614*, 2002.

[Smith et al.-1994] J. E. Smith, R. W. 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, Volume 31, Issue 3, Pages 343-354*, 1994.

[Smith et al.-1999] J. Smith, R. Gillham, *Effects of solute concentration-dependent surface tension on unsaturated flow: Laboratory sand column experiments, Water Resource Research, Volume 35, Issue 4, Pages 973-982*, 1999.

[Stauffer-1978] F. Stauffer, *Time dependence of the relations between capillary pressure, water content and conductivity during drainage of porous media, IAHR symposium on scale effects in porous media, Volume 29*, 1978.

[Sun et al.-2005] S. Sun, M. F. Wheeler, *Discontinuous Galerkin methods for coupled flow and reactive trans-*

port problems, Applied Numerical Mathematics Volume 52, Issues 2–3, Pages 273-298, 2005.

[Vohralik-2007] M. Vohralik, *A posteriori error estimates for lowest-order mixed finite element discretizations of convection-diffusion-reaction equations, SIAM Journal on Numerical Analysis, Volume 45, Issue 4, Pages 1570-1599*, 2007.

[Walker et al.-2011] H. F. Walker, P. Ni, *Anderson Acceleration for Fixed-Point Iterations, SIAM Journal on Numerical Analysis, Volume 49, Issue 4, Pages 1715 - 1735*, 2011.

[Wang et al.-2013] X. Wang, H. A. Tchelepi, *Trust-region based solver for nonlinear transport in heterogeneous porous media, Journal of Computational Physics, Volume 253, Pages 114-137*, 2013.

[Woodward et al.-2000] C. S. Woodward, C. N. Dawson, *Analysis of expanded mixed finite element methods for a nonlinear parabolic equation modeling flow into variably saturated porous media, SIAM Journal on Numerical Analysis, Volume 37, Issue 3, Pages 701-724*, 2000.

[Zha et al.2019] Y. Zha, J. Yang, J. Zeng, C.-H. M. Tso, W. Zeng, L. Shi, *Review of numerical solution of Richardson-Richards equation for variably saturated flow in soils, WIREs Water, Volume 6, Issue 5*, 2019.

[Zhang et al.-2017] H. Zhang, P. A. Zegeling, *A Numerical Study of Two-Phase Flow Models with Dynamic Capillary Pressure and Hysteresis, Transport in Porous Media, Volume 116, Issue 2, Pages 825-846*, 2017.

Paper D

Global random walk solvers for fully coupled flow and transport in saturated/unsaturated porous media

Nicolae Suciu, Davide Illiano, Alexander Prechtel, Adrian Florin Radu

Global random walk solvers for fully coupled flow and transport in saturated/unsaturated porous media. ...
https://arXiv:2011.12889

Global random walk solvers for fully coupled flow and transport in saturated/unsaturated porous media

Nicolae Suciu[a,b,*], Davide Illiano[c], Alexander Prechtel[a], Florin A. Radu[c]

[a] *Mathematics Department, Friedrich-Alexander University of Erlangen-Nürnberg, Cauerstraße. 11, 91058 Erlangen, Germany*
[b] *Tiberiu Popoviciu Institute of Numerical Analysis, Romanian Academy, Fântanele 57, 400320 Cluj-Napoca, Romania*
[c] *Department of Mathematics, University of Bergen, Allégaten 41, 5007 Bergen, Norway*

Abstract

In this article, we present new random walk methods to solve flow and transport problems in saturated/unsaturated porous media, including coupled flow and transport processes in soils, heterogeneous systems modeled through random hydraulic conductivity and recharge fields, processes at the field and regional scales. The numerical schemes are based on global random walk algorithms (GRW) which approximate the solution by moving large numbers of computational particles on regular lattices according to specific random walk rules. To cope with the nonlinearity and the degeneracy of the Richards equation and of the coupled system, we implemented the GRW algorithms by employing linearization techniques similar to the *L*-scheme developed in finite element/volume approaches. The resulting GRW *L*-schemes converge with the number of iterations and provide numerical solutions that are first-order accurate in time and second-order in space. A remarkable property of the flow and transport GRW solutions is that they are practically free of numerical diffusion. The GRW solvers are validated by comparisons with mixed finite element and finite volume solvers in one- and two-dimensional benchmark problems. They include Richards' equation fully coupled with the advection-diffusion-reaction equation and capture the transition from unsaturated to saturated flow regimes.

Keywords: Richards equation, Coupled flow and transport, Linearization, Iterative schemes, Global random walk
MSC: 76S05, 65N12, 86A05, 65C35, 76R50

1. Introduction

The accuracy and the robustness of the numerical schemes is the primary requirement for reliable and meaningful results of the current efforts to improve the understanding of the complexity and interdependence of the flow and transport processes in subsurface hydrology through numerical investigations. Numerical solvers for partial differential equations modeling individual or coupled processes are often used as basic elements in the formulation of the more complex problems of practical interest, such as parameter identification [14], hydraulic tomography [6], Monte Carlo approaches for systems with randomly distributed parameters [22], or upscaling for mutiphase flows in heterogeneous subsurface formations [12], among others.

A central issue in subsurface hydrology is the need of robust and computationally efficient numerical models for partially saturated soil-groundwater systems. The transition between unsaturated and saturated zones is particularly challenging. In unsaturated flows the water content θ and the hydraulic conductivity K depend nonlinearly on the pressure head ψ through material laws based on experiments, as far as $\psi < 0$. The evolution of ψ is governed by the parabolic Richards' equation which degenerates to a (generally) linear elliptic equation (i.e. the equation for steady-state flow in aquifers) if $\psi \geq 0$ [3]. Since the regions where degeneracy takes place depend on the evolution of the pressure ψ in time and space, they are not known *a priori*. To cope with the nonlinearity and degeneracy of the Richards' equation, different linearization methods are needed, such as the Newton scheme [28, 12, 16], which is second-order convergent but converges only locally (requires a starting point close enough to the solution) or the more robust but only first-order convergent Picard, modified Picard, or *L* schemes [30, 23, 21, 26].

Explicit and implicit schemes proposed for nonlinear flows in unsaturated regime provide solutions with comparable accuracy but are generally ambiguous to compare in terms of computing time. Since they do not need to solve systems of linear algebraic equations at every time step, explicit schemes are in principle faster [20] but

*Corresponding author.
Email address: suciu@math.fau.de (Nicolae Suciu)

their speed may be seriously affected by the need to use very small time steps [13, 1]. The time step in explicit schemes is constraint by stability conditions [32] and has to be significantly reduced to ensure small local Péclet number (Pé), defined with respect to the space step. Large (global) Pé characterizes advection-dominated transport problems [4, 18]. In such cases, reducing the local Pé is a remedy to avoid the numerical diffusion and the oscillatory behavior of the solution [24]. The criterion of small local Pé is also recommended for numerical schemes solving the pressure equation in saturated flows [11] and, since Richards' equation has the structure of the advection-diffusion equation, the recommendation holds for the unsaturated flows as well.

Well known approaches to avoid the numerical diffusion are the particle tracking in continuous space and the discrete random walk on lattices [33]. The accuracy of these schemes is determined by the number of computational particles undergoing random jumps in continuous space or on discrete lattices. In random walk schemes, the increase of the computation time with the number of particles is simply avoided by randomly distributing the particles along the spatial directions with a global procedure, according to appropriate jump probabilities. In this way, one obtains a global random walk (GRW) which performs the spreading of all the particles from a given site with computational costs that are practically the same as for generating the jump of a single random walker in sequential procedures [36]. In particular cases (e.g., when using biased jump probabilities to account for variable coefficients or for advective displacements) the GRW algorithms are equivalent to explicit finite difference schemes with time step size constrained by stability requirements. In unbiased GRW schemes for transport problems with variable coefficients, which still satisfy stability conditions, no restrictions on the time step are needed to reduce the local Pé number, which renders the approach particularly efficient in large scale simulations of transport in groundwater (see [33] for details and examples).

The elliptic and parabolic equations governing the pressure head for flows in unsaturated/saturated porous media are essentially diffusion equations with second order operator in Stratonovich form. They can be recast as Fokker-Planck equations, with drift augmented by the row derivative of the coefficient tensor, and further solved by random walk approaches [33]. An alternative approach starts with a staggered finite difference scheme, further used to derive biased random walk rules governing the movement on a regular lattice of a system of computational particles. The particle density at lattice sites provides a numerical approximation of the pressure head solution. This approach has been already illustrated for flows in saturated porous media with heterogeneous hydraulic conductivity [1, 34].

In this article, we present new GRW schemes for nonlinear and non-steady flows in soils which model the transition from unsaturated to saturated regime in a way consistent with the continuity of the constitutive relationships $\theta(\psi)$ and $K(\psi)$. Following [21, 26], the nonlinearity of the Richards equation is solved with an iterative procedure similar to the L-scheme used in finite element/volume approaches. Numerical tests demonstrate the convergence of the L-scheme for unsaturated/saturated flows. For fully saturated flow regime with constant water content θ and time independent boundary conditions the GRW L-scheme is equivalent to a transient finite difference scheme.

Coupled flow and reactive transport problems for partially saturated soils rise new stability and consistency issues and demand augmented computational resources. Our GRW approach in this case consists of coupling the flow solver described above with existing GRW transport solvers [33] adapted for nonlinear problems, which are implemented as L-schemes as well. The flow and transport solvers are coupled via an alternating splitting procedure [15] which successively iterates the corresponding L-schemes until the convergence of the pressure head and concentration solutions is reached, within the same tolerance, at every time step. Code verification tests using analytical manufactured solutions are employed to verify the convergence of the iterations and the accuracy of the splitting scheme.

The GRW scheme for one-dimensional solutions of the Richards equation, which captures the transition from unsaturated to saturated flow regimes is validated by comparisons with solutions provided by Richy software, based on the mixed finite element method (MFEM), with backward Euler discretization in time and Newton linearization, developed at the Mathematics Department of the Friedrich-Alexander University of Erlangen-Nürnberg [27, 28]. The two-dimensional GRW solutions are compared on benchmark problems with two-point flux approximation (TPFA) finite volume solvers using backward Euler discretization in time and L-scheme linearization [15]. The TPFA codes are implemented in MRST, the MATLAB Reservoir Simulation Toolbox [19].

The paper is organized as follows. Section 2 presents the GRW algorithm and the linearization approach for one-dimensional flow problems. The one-dimensional solver is further validated through comparisons with MFEM solutions in Section 2.2. Two dimensional GRW algorithms for fully coupled and decoupled flow and transport problems are introduced in Section 3. Code verification tests and comparisons with TPFA solutions for benchmark problems are presented in Section 4. Some examples of flow and transport solutions for groundwater problems at the field and regional scale are presented in Section 5. The main conclusions of this work are finally presented in Section 6. GRW codes implemented in Matlab for model problems considered in this article are stored in the Git repository https://github.com/PMFlow/RichardsEquation.

2. One-dimensional GRW solutions

2.1. One-dimensional GRW algorithm for unsaturated/saturated flow in soils

We consider the water flow in unsaturated/saturated porous media described by the one-dimensional Richards equation [13, 27, 16] in the space-time domain $[0, L_z] \times [0, T]$,

$$\frac{\partial \theta(\psi)}{\partial t} - \frac{\partial}{\partial z}\left[K(\theta(\psi))\frac{\partial}{\partial z}(\psi + z)\right] = 0, \tag{1}$$

where $\psi(z,t)$ is the pressure head expressed in length units, θ is the volumetric water content, K stands for the hydraulic conductivity of the medium, and z is the height oriented positively upward. According to (1), the water flux given by Darcy's law is $q = -K(\theta(\psi))\frac{\partial}{\partial z}(\psi + z)$.

To design a GRW algorithm, we start with the staggered finite difference scheme with backward discretization in time which approximates the solution of Eq. (1) at positions $z = i\Delta z$, $i = 1, \ldots, I$, $I = L_z/\Delta z$, and time points $t = k\Delta t$, $k = 1, \ldots, T/\Delta t$, according to

$$\theta(\psi_{i,k}) - \theta(\psi_{i,k-1}) = \frac{\Delta t}{\Delta z^2}\left\{\left[K(\psi_{i+1/2,k})(\psi_{i+1,k} - \psi_{i,k}) - K(\psi_{i-1/2,k})(\psi_{i,k} - \psi_{i-1,k})\right] + \left(K(\psi_{i+1/2,k}) - K(\psi_{i-1/2,k})\right)\Delta z\right\}. \tag{2}$$

To cope with the double nonlinearity due to the dependencies $K(\theta)$ and $\theta(\psi)$ we propose an explicit scheme similar to the linearization approach known as "L-scheme", originally developed for implicit methods [e.g. 23, 21, 26]. The approach consists of the addition of a stabilization term $L(\psi_{i,k}^{s+1} - \psi_{i,k}^{s})$, $L = const$, in the left-hand side of (2) and of performing successive iterations $s = 1, 2, \ldots$ of the modified scheme until the discrete L^2 norm of the solution $\psi_k^s = (\psi_{i,k}^s, \ldots, \psi_{I,k}^s)$ verifies

$$\|\psi_k^s - \psi_k^{s-1}\| \le \varepsilon_a + \varepsilon_r \|\psi_k^s\| \tag{3}$$

for some given tolerances ε_a and ε_r. The adapted L-scheme reads

$$\psi_{i,k}^{s+1} = \left[1 - (r_{i+1/2,k}^s + r_{i-1/2,k}^s)\right]\psi_{i,k}^s + r_{i+1/2,k}^s \psi_{i+1,k}^s + r_{i-1/2,k}^s \psi_{i-1,k}^s + \left(r_{i+1/2,k}^s - r_{i-1/2,k}^s\right)\Delta z - \left(\theta(\psi_{i,k}^s) - \theta(\psi_{i,k-1})\right)/L, \tag{4}$$

where

$$r_{i\pm1/2,k}^s = K(\psi_{i\pm1/2,k}^s)\Delta t/(L\Delta z^2). \tag{5}$$

For fixed time step k, the iterations start with the solution after the last iteration at the previous time $k-1$, $\psi_{i,k}^1 = \psi_{i,k-1}$, $i = 1, \ldots, I$. Note that, unlike implicit L-schemes (e.g., [30, 23, 21]), the explicit scheme (4) uses forward increments of ψ. In this way, the solution $\psi_{i,k}^{s+1}$ is obtained from values of ψ and r evaluated at the previous iteration, without solving systems of algebraic equations.

The solution $\psi_{i,k}^s$ is further represented by the distribution of $\mathcal{N}$ computational particles at the sites of the one-dimensional lattice, $\psi_{i,k}^s \approx n_{i,k}^s a/\mathcal{N}$, with a being a constant equal to a unit length, and the L-scheme (4) becomes

$$n_{i,k}^{s+1} = \left[1 - \left(r_{i+1/2,k}^s + r_{i-1/2,k}^s\right)\right] n_{i,k}^s + r_{i+1/2,k}^s n_{i+1,k}^s + r_{i-1/2,k}^s n_{i-1,k}^s + \lfloor \mathcal{N} f^s \Delta t \rfloor, \tag{6}$$

where the source term is defined as $f^s = \left(r_{i+1/2,k}^s - r_{i-1/2,k}^s\right)\Delta z - \left[\theta(n_{i,k}^s) - \theta(n_{i,k-1}\right]/L$ and $\lfloor \cdot \rfloor$ denotes the floor function.

The physical dimension of the parameter L of the scheme is that of an inverse length unit to ensure that $r_{i\pm1/2,k}^s$ defined by (5) are non-dimensional parameters, as needed in random walk approaches [36, 33]. By imposing the constraint $r_{i\pm1/2,k}^s \le 1/2$, the parameters $r_{i\pm1/2,k}^s$ can be thought of as biased jump probabilities. Hence, the contributions to $n_{i,k}^{s+1}$ from neighboring sites $i \pm 1$ summed up in (6) can be obtained with the GRW algorithm which moves particles from sites j to neighboring sites $i = j \mp 1$ according to the rule

$$n_{j,k}^s = \delta n_{j,j,k}^s + \delta n_{j-1,j,k}^s + \delta n_{j+1,j,k}^s. \tag{7}$$

For consistency with (6), the quantities δn^s in (7) have to satisfy in the mean [33, Sect. 3.3.4.1],

$$\overline{\delta n_{j,j,k}^s} = \left[1 - \left(r_{j-1/2,k}^s + r_{j+1/2,k}^s\right)\right]\overline{n_{j,k}^s}, \quad \overline{\delta n_{j\mp1/2,j,k}^s} = r_{j\mp1/2,k}^s \overline{n_{j,k}^s}. \tag{8}$$

The quantities δn^s are binomial random variables approximated by using the unaveraged relations (8) for the mean, summing up the reminders of multiplication by r and of the floor function $\lfloor \mathcal{N} f^s \Delta t \rfloor$, and allocating one particle to the lattice site where the sum reaches the unity.

Remark 1. *Consider the saturated regime, $\theta = const$, with space-variable hydraulic conductivity K and a given source term f. With the parameter L set to $L = 1/a$, after disregarding the time index k the scheme (6) solves the following equation for the hydraulic head $h = \psi + z$,*

$$\frac{1}{a}\frac{\partial h}{\partial s} - \frac{\partial}{\partial z}\left[K\frac{\partial h}{\partial z}\right] = f. \tag{9}$$

For boundary conditions independent of s, the solution of Eq. (9) approaches a steady-state regime corresponding to the saturated flow (see also [1, 34]). The modified GRW scheme (6) is equivalent to a convergent finite difference scheme first order accurate in time and second order in space [35, Remark 1].

2.2. Validation of the one-dimensional GRW flow algorithm

The one-dimensional algorithm for flow in unsaturated/saturated soils is validated in the following by comparisons with MFEM solutions obtained with the Richy software [27, 28]. For this purpose, we solve one-dimensional model-problems for the vertical infiltration of the water through both homogeneous and non-homogeneous soil columns [31], previously used in [28] to assess the accuracy and the convergence of the MFEM solutions.

We consider the domain $z \in [0, 2]$ and the boundary conditions specified by a constant pressure $\psi(0, t) = \psi_0$ at the bottom of the soil column and a constant water flux q_0 at the top. Together, these constant conditions determine the initial pressure distribution $\psi(z, 0)$ as solution of the steady-state flow problem. For $t > 0$, the pressure ψ_0 is kept constant, at the bottom, and the water flux at the top of the column is increased linearly from q_0 to q_1 until $t \le t_1$ and is kept constant for $t > t_1$.

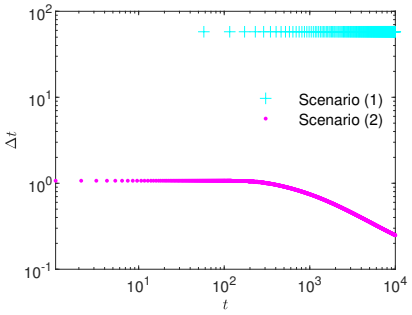


Figure 1: Time steps for Scenario (1) and Scenario (2).

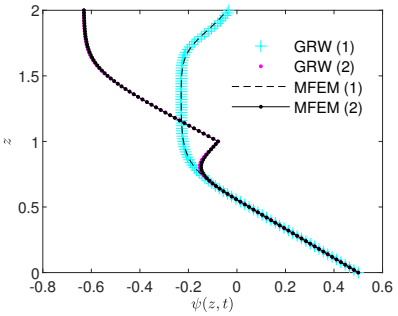


Figure 2: Pressure head solutions at $t = 10^4$ seconds computed by GRW and MFEM codes.

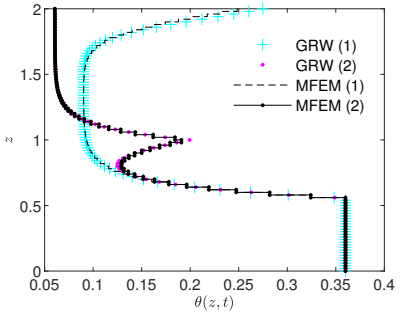


Figure 3: Water content solutions at $t = 10^4$ seconds computed by GRW and MFEM codes.

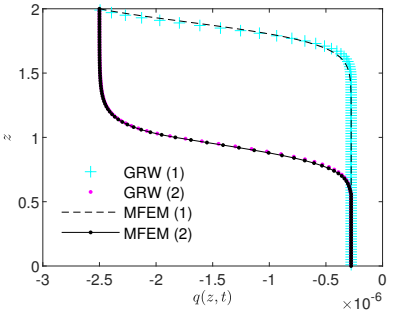


Figure 4: Water flux solutions at $t = 10^4$ seconds computed by GRW and MFEM codes.

For the unsaturated regions ($\psi < 0$) we consider the constitutive relationships given by the simple exponential model [10]

$$\theta(\psi) = \theta_{res} + (\theta_{sat} - \theta_{res})e^{\alpha\psi}, \tag{10}$$

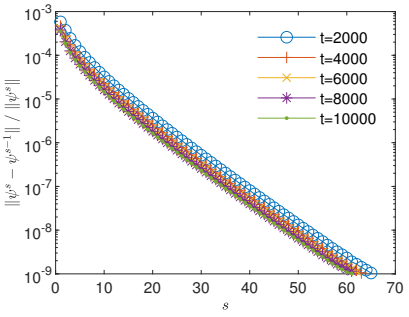


Figure 5: Convergence of the L-scheme implementation of the GRW flow solver in Scenario (1).

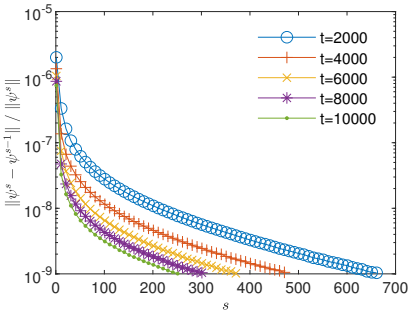


Figure 6: Convergence of the L-scheme implementation of the GRW flow solver in Scenario (2).

$$K(\theta(\psi)) = K_{sat}\frac{\theta(\psi) - \theta_{res}}{\theta_{sat} - \theta_{res}}, \tag{11}$$

where $\theta = \theta_{sat}$ and $K = K_{sat}$ denote the constant water content respectively the constant hydraulic conductivity in the saturated regions ($\psi \geq 0$) and θ_{res} is the residual water content. The more complex and physically sounded van Genuchten-Mualem parameterization model will be used for two-dimensional problems in the following sections.

The flow problem for Eq. (1) with the parameterization (10-11) is solved in two Scenarios: (1) homogeneous soil, with $K_{sat} = 2.77 \cdot 10^{-6}$, $\theta_{res} = 0.06$, $\theta_{sat} = 0.36$, $\alpha = 10$, $q_0 = 2.77 \cdot 10^{-7}$, $q_1 = 2.50 \cdot 10^{-6}$, which are representative for a sandy soil, and (2) non-homogeneous soil, with the same parameters as in Scenario (1), except the saturated hydraulic conductivity, which takes two constant values, $K_{sat} = 2.77 \cdot 10^{-6}$ for $z < 1$ and $500K_{sat}$ for $z \geq 1$ (modeling, for instance, a column filled with sand and gravel). To capture the transition from unsaturated to saturated regime, the pressure at the bottom boundary is fixed at $\psi_0 = 0.5$. For the parameters of the one-dimensional flow problems solved in this section we consider meters as length units and seconds as time units. The simulations are conducted up to $T = 10^4$ (about 2.78 hours) and the intermediate time is taken as $t_1 = T/10^2$.

We consider a uniform GRW lattice with $\Delta z = 10^{-2}$, equal to the length of the linear elements in the MFEM solver. The GRW computations are initialized by multiplying the initial condition by $\mathcal{N} = 10^{24}$ particles. Since, as shown by (11), the hydraulic conductivity varies in time, the length of the time step determined by (5) for the maximum of K at every time iteration and by specifying a maximum $r_{max} = 0.8$ of the parameter $r_{i\pm1/2,k}$ may vary in time (see Fig. 1). The parameter of the regularization term in the L-scheme is set to $L = 1$ for the computation of the initial condition (solution of the stationary problem, i.e. for $\partial\theta/\partial t = 0$ in (1)) and to $L = 2$ for the solution of the non-stationary problem. In both cases, the convergence criterion (3) is verified by choosing $\varepsilon_a = 0$ and a relative tolerance $\varepsilon_r = 10^{-9}$.

The comparison with the MFEM solutions presented in Figs. 2-4 shows a quite good accuracy of the GRW solutions for pressure, water content, and water flux. The relative errors, computed with the aid of the L^2 norms by $\varepsilon_\psi = \|\psi^{GRW} - \psi^{MFEM}\|/\|\psi^{MFEM}\|$, and similarly for θ and q, are presented in Table 1.

Table 1: Error norms of the GRW solutions.

	ε_ψ	ε_θ	ε_q
Scenario (1)	1.81e-02	2.20e-02	3.50e-02
Scenario (2)	5.20e-03	2.35e-02	2.07e-02

The L-scheme converges with speeds depending on the problem. To solve the problem for the initial condition, one needs $3.5 \cdot 10^4$ iterations in Scenario (1) and $6.5 \cdot 10^6$ iterations in Scenario (2). Instead, to solve the non-stationary problem for a final time $T = 10^4$, one needs about 70 iterations in Scenario (1) and about 700 iterations in Scenario (2) (see Fig. 5 and Fig. 6). The convergence of the iterative GRW L-scheme can be further investigated through assessments of the computational order of convergence of the sequence of successive correction norms $\|\psi_k^s - \psi_k^{s-1}\|$ [8, 9]. Estimations provided in [35, Appendix A] indicate a linear convergence for Scenario (1) but only a power law convergence $\sim s^{-1}$, which is slower than the linear convergence [9], for Scenario (2).

Supplementary tests done in Scenario (1) indicate the existence of a lower bound of the constant L which ensures the convergence [35, Sect. 3]. It is found that increasing L above the value which ensures the convergence

of the GRW L-scheme with a desired accuracy only results in increasing number of iterations and more computing time. The parameter L has to be established experimentally by checking the convergence and, as highlighted by the examples presented in Section 4 below, it depends on the complexity of the problem to be solved.

3. Two-dimensional GRW solutions

3.1. Two-dimensional GRW algorithm for flow in soils and aquifers

In two spatial dimensions the pressure head $\psi(x, z, t)$ satisfies the equation

$$\frac{\partial}{\partial t}\theta(\psi) - \nabla \cdot [K(\theta(\psi)\nabla(\psi + z)] = 0. \tag{12}$$

The two-dimensional GRW algorithm on regular staggered grids ($\Delta x = \Delta z$) which approximates the solution of (12) by computational particles, $\psi \approx na/\mathcal{N}$, is constructed similarly to (6-8). The solution at iteration $s + 1$ is obtained by gathering particles from neighboring sites according to

$$\begin{aligned} n^{s+1}_{i,j,k} &= \left[1 - \left(r^s_{i+1/2,j,k} + r^s_{i-1/2,j,k} + r^s_{i,j+1/2,k} + r^s_{i,j-1/2,k}\right)\right] n^s_{i,j,k} \\ &\quad + r^s_{i+1/2,j,k} n_{i+1,j,k} + r^s_{i-1/2,j,k} n_{i-1,j,k} \\ &\quad + r^s_{i,j+1/2,k} n_{i,j+1,k} + r^s_{i,j-1/2,k} n_{i,j-1,k} + \lfloor \mathcal{N} f^s \Delta t \rfloor, \end{aligned} \tag{13}$$

where the source term is defined as $f^s = \left(r^s_{i,j+1/2,k} - r^s_{i,j-1/2,k}\right)\Delta z - \left[\theta(n^s_{i,j,k}) - \theta(n_{i,j,k-1})\right]/L$. The two-dimensional GRW rule which at time k moves particles from sites (l, m) to neighboring sites $(l \mp 1, m \mp 1)$ reads as follows,

$$n^s_{l,m,k} = \delta n^s_{l,m|l,m,k} + \delta n^s_{l-1,m|l,m,k} + \delta n^s_{l+1,m|l,m,k} + \delta n^s_{l,m-1|l,m,k} + \delta n^s_{l,m+1|l,m,k}. \tag{14}$$

For consistency with (13), the numbers of particles δn^s verify in the mean

$$\begin{aligned} \overline{\delta n^s_{l,m|l,m,k}} &= \left[1 - \left(r^s_{l-1/2,m,k} + r^s_{l+1/2,m,k} + r^s_{l,m-1/2,k} + r^s_{l,m+1/2,k}\right)\right] \overline{n^s_{l,m,k}} \\ \overline{\delta n^s_{l\mp1,m|l,m,k}} &= r^s_{l\mp1/2,m,k} \overline{n^s_{l,m,k}} \\ \overline{\delta n^s_{l,m\mp1|l,m,k}} &= r^s_{l,m\mp1/2,k} \overline{n^s_{l,m,k}}. \end{aligned} \tag{15}$$

The parameters $r^s_{l\mp1/2,m,k}$ and $r^s_{l,m\mp1/2,k}$, defined by relations similar to (5), are dimensionless positive real numbers. They represent biased jump probabilities on the four allowed spatial directions of the GRW lattice and are constraint by the first relation (15) such that their sum be less or equal to one. A sufficient condition would be that each of them verifies $r \leq 1/4$.

The binomial random variables variables δn are approximated in the same way as in the one-dimensional case. By giving up the particle indivisibility, one obtains deterministic GRW algorithms which represent the solution n by real numbers and use the unaveraged relations (15) for the computation of the δn terms. In the following we use this deterministic implementation of the GRW algorithm to compute flow solutions for unsaturated/saturated porous media.

Remark 2. *After disregarding the index k and letting $L = 1/a$, $\theta = const$, the algorithm (13-15) becomes a transient scheme to solve the equation governing flows in saturated porous media [1, 34] (see also Remark 1).*

3.2. GRW algorithms for two-dimensional fully coupled flow and surfactant transport

Let the pressure $\psi(x, z, t)$ and the concentration $c(x, z, t)$ solve the equations of the following model of fully coupled flow and surfactant transport in unsaturated/saturated porous media [17, 15],

$$\frac{\partial}{\partial t}\theta(\psi, c) - \nabla \cdot [K(\theta(\psi, c)\nabla(\psi + z)] = 0, \tag{16}$$

$$\frac{\partial}{\partial t}[\theta(\psi, c)c] - \nabla \cdot [D\nabla c - \mathbf{q}c] = R(c), \tag{17}$$

where $\mathbf{q} = -K(\theta(\psi, c)\nabla(\psi + z)$ is the water flux (Darcy velocity) and $R(c)$ is a nonlinear reaction term. Equations (16-17) are coupled in both directions through the nonlinear functions $\theta(\psi, c)$ and $\theta(\psi, c)c$. The pressure equation (16) is solved with the GRW L-scheme described in the previous subsection, with a slight modification due to the dependence of θ on both ψ and c. New algorithms are needed instead to solve the coupled, nonlinear transport equation (17).

3.2.1. Biased GRW algorithm for transport problems

To derive a GRW algorithm for the transport equation, we start with a backward-time central-space finite difference scheme for Eq. (17). Considering a diagonal diffusion tensor with constant components D_1 and D_2, and denoting by U and V the components of the Darcy velocity along the horizontal axis x and the vertical axis z, by Δt the time step, and by Δx and Δz the spatial steps, the scheme reads as

$$\begin{aligned}
&\theta(\psi_{i,j,k}, c_{i,j,k})c_{i,j,k} - \theta(\psi_{i,j,k-1}, c_{i,j,k-1})c_{i,j,k-1} = \\
&-\frac{\Delta t}{2\Delta x}\left(U_{i+1,j,k}c_{i+1,j,k} - U_{i-1,j,k}c_{i-1,j,k}\right) - \frac{\Delta t}{2\Delta z}\left(V_{i,j+1,k}c_{i,j+1,k} - V_{i,j-1,k}c_{i,j-1,k}\right) \\
&+\frac{D_1\Delta t}{\Delta x^2}\left(c_{i+1,j,k} - 2c_{i,j,k} + c_{i-1,j,k}\right) + \frac{D_2\Delta t}{\Delta z^2}\left(c_{i,j+1,k} - 2c_{i,j,k} + c_{i,j-1,k}\right) = \\
&-\left(\frac{2D_1\Delta t}{\Delta x^2} + \frac{2D_2\Delta t}{\Delta z^2}\right)c_{i,j,k} \\
&+\left(\frac{D_1\Delta t}{\Delta x^2} - \frac{\Delta t}{2\Delta x}U_{i+1,j,k}\right)c_{i+1,j,k} + \left(\frac{D_1\Delta t}{\Delta x^2} + \frac{\Delta t}{2\Delta x}U_{i-1,j,k}\right)c_{i-1,j,k} \\
&+\left(\frac{D_2\Delta t}{\Delta z^2} - \frac{\Delta t}{2\Delta z}V_{i,j+1,k}\right)c_{i,j+1,k} + \left(\frac{D_2\Delta t}{\Delta z^2} + \frac{\Delta t}{2\Delta z}V_{i,j-1,k}\right)c_{i,j-1,k} + R(c_{i,j,k}).
\end{aligned} \tag{18}$$

Next, similarly to the scheme for the flow equation, we add a regularization term $L(c^{s+1}_{i,j,k} - c^s_{i,j,k})$ in Eq. (18), define the dimensional parameters

$$r_x = \frac{2D_1\Delta t}{L\Delta x^2}, \quad r_z = \frac{2D_2\Delta t}{L\Delta z^2}, \quad u^s_{i\pm1,j,k} = \frac{\Delta t}{L\Delta x}U^s_{i\pm1,j,k}, \quad v^s_{i,j\pm1,k} = \frac{\Delta t}{L\Delta z}V^s_{i,j\pm1,k}, \tag{19}$$

approximate the concentration by the density of the number of computational particles, $c^s_{i,j,k} \approx n^s_{i,j,k}/\mathcal{N}$, and finally we obtain

$$\begin{aligned}
n^{s+1}_{i,j,k} = &\left[1 - (r_x + r_z)\right] n^s_{i,j,k} \\
&+ \frac{1}{2}\left(r_x - u^s_{i+1,j,k}\right) n^s_{i+1,j,k} + \frac{1}{2}\left(r_x + u^s_{i-1,j,k}\right) n^s_{i-1,j,k} \\
&+ \frac{1}{2}\left(r_z - v^s_{i,j+1,k}\right) n^s_{i,j+1,k} + \frac{1}{2}\left(r_z + v^s_{i,j-1,k}\right) n^s_{i,j-1,k} + \lfloor \mathcal{N} g^s \Delta t \rfloor,
\end{aligned} \tag{20}$$

where $g^s = R(n^s_{i,j,k})/L - \left[\theta(\psi^s_{i,j,k}, n^s_{i,j,k})n^s_{i,j,k} - \theta(\psi_{i,j,k-1}, n_{i,j,k-1})n_{i,j,k-1}\right]/L$, with ψ approximated by the distribution of particles in the flow solver for Eq. (16). Note that the definition of the dimensionless numbers (19) implies that the parameter L has to be a dimensionless number as well.

The contributions to $n^{s+1}_{i,j,k}$ in Eq. (20) are obtained with the biased global random walk algorithm (BGRW)

$$n^s_{l,m,k} = \delta n^s_{l,m|l,m,k} + \delta n^s_{l-1,m|l,m,k} + \delta n^s_{l+1,m|l,m,k} + \delta n^s_{l,m-1|l,m,k} + \delta n^s_{l,m+1|l,m,k}, \tag{21}$$

where, for consistency with the finite difference scheme (20), the quantities δn verify in the mean

$$\overline{\delta n^s_{l,m|l,m,k}} = \left[1 - (r_x + r_z)\right]\overline{n^s_{i,j,k}}, \quad \overline{\delta n^s_{l\pm1,m|l,m,k}} = \frac{1}{2}(r_x \mp u^s_{l,m,k})\overline{n^s_{l,m,k}}, \quad \overline{\delta n^s_{l,m\pm1|l,m,k}} = \frac{1}{2}(r_z \mp v^s_{l,m,k})\overline{n^s_{l,m,k}}. \tag{22}$$

The binomial random variables δn used in the BGRW algorithm are approximated similarly to the algorithms described in the previous sections, by summing up to unity reminders of multiplication and floor operations. A deterministic BGRW algorithm can be obtained, similarly to the flow solver presented in Section 3.1 above, by giving up the particle's indivisibility and using the un-averaged relations (22). However, for the computations presented in the next section, we use a randomized implementation of the BGRW algorithm.

As follows from (22), the BGRW algorithm is subject to the following restrictions

$$r_x + r_z \le 1, \quad \left|u^s_{l,m,k}\right| \le r_x, \quad \left|v^s_{l,m,k}\right| \le r_z. \tag{23}$$

Remark 3. *The constraints (23) impose a limitation on the maximum allowable value of the local Péclet number. Assume a constant flow velocity $-V$ and a constant diffusion coefficient D. Then, according to (23) and (19), the condition $v \le r$ implies $Pé = V\Delta z/D \le 2$.*

Remark 4. *Taking into account that the iterations start with $n_{i,j,k-1}$, setting $L = 1$, $\theta = 1$, and dropping the superscripts s, the relation (20) becomes*

$$\begin{aligned} n_{i,j,k} &= \left[1 - (r_x + r_z)\right] n_{i,j,k-1} \\ &+ \frac{1}{2}\left(r_x - u_{i+1,j,k-1}\right) n_{i+1,j,k-1} + \frac{1}{2}\left(r_x + u_{i-1,j,k-1}\right) n_{i-1,j,k-1} \\ &+ \frac{1}{2}\left(r_z - v_{i,j+1,k-1}\right) n_{i,j+1,k-1} + \frac{1}{2}\left(r_z + v_{i,j-1,k-1}\right) n_{i,j-1,k-1} + \left\lfloor \mathcal{N} R(n_{i,j,k-1}) \Delta t \right\rfloor . \end{aligned} \tag{24}$$

Relation (24), together with (21-23), define a BGRW algorithm for (decoupled) reactive transport described by Eq. (17) with $\theta(\psi, c) = 1$.

3.2.2. Unbiased GRW algorithm for transport problems

The unbiased GRW algorithm is obtained by globally moving groups of particles according to the rule

$$\begin{aligned} n^{s+1}_{i,j,k} &= \delta n^{s}_{i+u^{s}_{i,j,k}, j+v^{s}_{i,j,k} | i,j,k} \\ &+ \delta n^{s}_{i+u^{s}_{i,j,k}+d, j+v^{s}_{i,j,k} | i,j,k} + \delta n^{s}_{i+u^{s}_{i,j,k}-d, j+v^{s}_{i,j,k} | i,j,k} \\ &+ \delta n^{s}_{i+u^{s}_{i,j,k}, j+v^{s}_{i,j,k}+d | i,j,k} + \delta n^{s}_{i+u^{s}_{i,j,k}, j+v^{s}_{i,j,k}-d | i,j,k} + \lfloor \mathcal{N} g^{s} \Delta t \rfloor , \end{aligned} \tag{25}$$

where d is a constant amplitude of diffusion jumps and the dimensionless variables r_x, r_z, u and v are defined similarly to (19) by

$$r_x = \frac{2D_1 \Delta t}{L(d\Delta x)^2}, \quad r_z = \frac{2D_2 \Delta t}{L(d\Delta z)^2}, \quad u^{s}_{i,j,k} = \left\lfloor \frac{\Delta t}{L\Delta x} U^{s}_{i,j,k} + 0.5 \right\rfloor, \quad v^{s}_{i,j,k} = \left\lfloor \frac{\Delta t}{L\Delta z} V^{s}_{i,j,k} + 0.5 \right\rfloor . \tag{26}$$

The particles distribution is updated at every time step by

$$n^{s}_{l,m,k+1} = \delta n^{s}_{l,m,k} + \sum_{i \neq l, j \neq m} \delta n^{s}_{l,m | i,j,k}. \tag{27}$$

The averages over GRW runs of the terms from (25) are now related by

$$\overline{\delta n^{s}_{i+u^{s}_{i,j,k}, j+v^{s}_{i,j,k} | i,j,k}} = \left[1 - (r_x + r_z)\right] \overline{n^{s}_{i,j,k}}, \quad \overline{\delta n^{s}_{i+u^{s}_{i,j,k} \pm d, j+v^{s}_{i,j,k} | i,j,k}} = \frac{r_x}{2} \overline{n^{s}_{i,j,k}}, \quad \overline{\delta n^{s}_{i+u^{s}_{i,j,k}, j+v^{s}_{i,j,k} \pm d | i,j,k}} = \frac{r_z}{2} \overline{n^{s}_{i,j,k}}. \tag{28}$$

Comparing with the BGRW relations (22), we remark that (26) defines unbiased jump probabilities $r_x/2$ and $r_y/2$ on the two spatial directions.

The unbiased GRW algorithm for decoupled transport is obtained by letting $L = 1$ and dropping the superscripts s (see also Remark 4).

The binomial random variables δn used in the unbiased GRW algorithm are approximated by the procedure used for the flow solver and for the BGRW algorithm presented in the previous subsection. For fixed space steps, the time step is chosen such that the dimensionless parameters $u^{s}_{i,j,k}$ and $v^{s}_{i,j,k}$ take integer values larger than unity which ensure the desired resolution of the velocity components [33, Sect. 3.3.2.1]. Further, the jumps' amplitude d is chosen such that the jump probabilities verify the constraint $r_x + r_z \leq 1$, imposed by the first relation (28).

The unbiased GRW, as well as the BGRW algorithm introduced in Section 3.2.1 above, have been tailored to solve problems with constant diffusion coefficients, as those considered in Sections 4.2 and 5.3 below. In case of diagonal diffusion tensors with space-time variable coefficients D_1 and D_2, the algorithms for the transport problem are straightforwardly obtained by assigning to r_x and r_z superscripts s and appropriate subscripts i, j, k.

4. Validation of the two-dimensional GRW algorithms

4.1. GRW flow solutions

For the beginning, we conduct verification tests of the GRW flow code by comparisons with an analytical solution and compute numerical estimates of the order of convergence. The results are further compared with those obtained by a TPFA code implemented in the MRST software [19, 15]. The two codes are tested by solving a problem with manufactured solution previously considered in [25]. The domain is the unit square $[0, 1] \times [0, 1]$ and the final time is $T = 1$. The manufactured solution for the pressure head ψ_m is given by

$$\psi_m(x, z, t) = -t\, x\, (x - 1)\, z\, (z - 1) \; - \; 1. \tag{29}$$

The water content θ and the conductivity K are expressed as

$$\theta(\psi) = \frac{1}{1-\psi}, \quad K(\theta(\psi)) = \psi^2. \tag{30}$$

The analytical solution (29) defines the boundary and initial conditions and induces a source term f, computed analytically from Eq. (12) with parameters given by the expressions (30).

We start the computations on a uniform mesh with $\Delta x = \Delta z = 0.1$ and halve the mesh size step three times successively. The accuracy of the numerical solutions, at the final time $t = T$, is quantified by the L^2 norm $\varepsilon_l = \|\psi^{(l)} - \psi_m\|$, $l = 1, \ldots, 4$, where $l = 1$ corresponds to the original mesh. The estimated order of convergence (EOC) that describes the decrease of the error in logarithmic scale is computed according to

$$EOC = \log\left(\frac{\varepsilon_l}{\varepsilon_{l+1}}\right) / \log(2), \quad l = 1, \ldots, 3. \tag{31}$$

The computations with the TPFA code start with a time step $\Delta t = 0.1$ which is also halved at each refinement of the mesh. The parameters of the convergence indicator (3) are set to $\varepsilon_a = 10^{-6}$ and $\varepsilon_r = 0$. Finally, the linearization parameter L is set equal to $1/2$ and the convergence of the L-scheme is achieved after circa 100 iterations per time step, independently of the mesh size.

In the GRW computations we use the same spatial refinement of the grid and tolerances ε_a and ε_r as above but, according to (5), we have to use adaptive time steps $\Delta t = O(\Delta z^{1/2})$ (see discussion in Section 2.2). The convergence criterion (3) is already fulfilled by the GRW L-scheme with parameter $L = 1$ for numbers of iterations increasing from $s = 2$ to $s = 5$ as the space step decreases. The accuracy ε_l instead is strongly influenced by L. For $L < 800$ the ε_l values may increase with the refinement of the mesh, leading to negative EOC, that is, the GRW solution does not converge to the exact solution ψ_m. However, it is found that the increase of ε_l is prevented by using a sufficiently large parameter L.

The results presented in Table 2 indicate the convergence of order 1 in space for TPFA and of order 2 for the GRW solutions. The higher order of convergence also leads to much smaller errors of the GRW code after the first refinement of the mesh.

Table 2: Estimated order of convergence of the TPFA and GRW flow solvers.

	ε_1	EOC	ε_2	EOC	ε_3	EOC	ε_4
TPFA	8.45e-03	0.94	4.40e-03	0.97	2.25e-03	0.97	1.15e-03
GRW (L=800)	7.20e-03	2.24	1.52e-03	3.21	1.65e-04	0.50	1.17e-04
GRW (L=1000)	9.24e-03	2.22	1.99e-03	2.83	2.80e-04	1.66	8.84e-05
GRW (L=1200)	8.89e-03	2.23	1.90e-03	2.80	2.72e-04	2.14	6.16e-05

Further, we solve the benchmark problem from [21, Sect. 4.2], which describes the recharge of a groundwater reservoir from a drainage trench in a two-dimensional geometry. The groundwater table is fixed by a Dirichlet boundary condition on the right hand side. The drainage process is driven by a Dirichlet boundary condition changing in time on the upper boundary of Ω.

The precise structure of the domain is defined by

$$\begin{aligned}
\Omega &= (0,2) \times (0,3),\\
\Gamma_{D_1} &= \{(x,z) \in \partial\Omega \mid x \in [0,1] \wedge z = 3\},\\
\Gamma_{D_2} &= \{(x,z) \in \partial\Omega \mid x = 2 \wedge z \in [0,1]\},\\
\Gamma_D &= \Gamma_{D_1} \cup \Gamma_{D_2},\\
\Gamma_N &= \partial\Omega \setminus \Gamma_D.
\end{aligned}$$

The Dirichlet and Neumann boundary conditions on Γ_D and Γ_N, respectively, as well as the initial condition consisting of hydrostatic equilibrium are specified as follows:

$$\begin{aligned}
\psi(x,z,t) &= \begin{cases} -2 + 2.2t/\Delta t_D, & \text{on } \Gamma_{D_1}, T \le \Delta t_D,\\ 0.2, & \text{on } \Gamma_{D_1}, T > \Delta t_D,\\ 1 - z, & \text{on } \Gamma_{D_2}, \end{cases}\\
-K(\theta(\psi(x,z,t))\nabla(\psi(x,z,t) + z) \cdot \mathbf{n} &= 0, \quad \text{on } \Gamma_N,\\
\psi(x,z,0) &= 1 - z, \quad \text{on } \Omega,
\end{aligned}$$

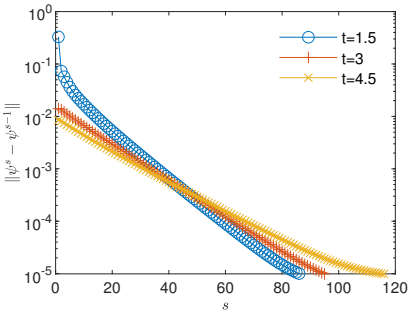


Figure 7: Convergence of the L-scheme implementation of the GRW flow solver for the loam soil problem at three time levels (in hours).

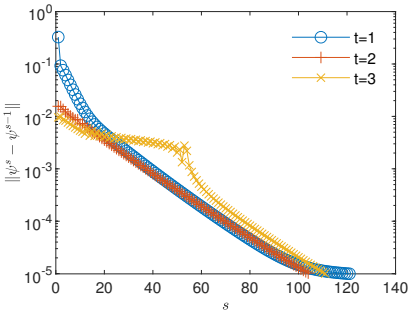


Figure 8: Convergence of the L-scheme implementation of the GRW flow solver for the clay soil problem at three time levels (in days).

where $\mathbf{n}$ represents the outward pointing normal vector.

The relationships defining the water content $\theta(\psi)$ and the hydraulic conductivity $K(\theta(\psi))$ are given by the van Genuchten-Mualem model

$$\theta(\psi) = \begin{cases} \theta_{res} + (\theta_{sat} - \theta_{res})\left(\frac{1}{1+(-\alpha\psi)^n}\right)^{\frac{n-1}{n}}, & \psi < 0 \\ \theta_{sat}, & \psi \geq 0, \end{cases} \tag{32}$$

$$K(\theta(\psi)) = \begin{cases} K_{sat}\Theta(\psi)^{\frac{1}{2}}\left[1 - \left(1 - \Theta(\psi)^{\frac{n}{n-1}}\right)^{\frac{n-1}{n}}\right]^2, & \psi < 0 \\ K_{sat}, & \psi \geq 0, \end{cases} \tag{33}$$

where θ_{res}, θ_{sat}, and K_{sat} represent the same parameters as for the exponential model considered in Section 2.2, $\Theta = (\theta - \theta_{res})/(\theta_{sat} - \theta_{res})$ is the normalized water content, and α and n are model parameters depending on the soil type.

We consider here two sets of soil parameters, presented in Table 3, which correspond to a silt loam and a Beit Netofa clay, respectively.

Table 3: Simulation parameters

	Silt loam	Beit Netofa clay
Vam Genuchten parameters:		
θ_{sat}	0.396	0.446
θ_{res}	0.131	0
α	0.423	0.152
n	2.06	1.17
K_{sat}	$4.96 \cdot 10^{-2}$	$8.2 \cdot 10^{-4}$
Time parameters:		
Δt_D	1/16	1
Δt	1/48	1/3
T	3/16	3

The time unit is 1 day and spatial dimensions are given in meters. Furthermore, we consider a regular mesh consisting of 651 nodes (i.e., $\Delta x = \Delta z = 0.1$).

By setting the stabilization parameters to $L = 0.5$ for loam and for $L = 0.12$ for clay, the convergence criterion (3) with $\varepsilon_a = \varepsilon_r = 5 \cdot 10^{-6}$ is fulfilled after about 120 iterations of the GRW L-scheme, for both soil models (Figs. 7 and 8). The results shown in Figs. 9 and 10 are as expected for this benchmark problem (see [28, 21]): the drainage process in the clay soil is much slower, so that the pressure distribution after three days is similar to that established in the loam soil after 4.5 hours.

The results obtained with the TPFA L-scheme, with $L = 1$ for both soil models, are used as reference to compute the relative errors ε_ψ, ε_θ, ε_{q_x}, and ε_{q_z} shown in Table 4. One remarks that ε_ψ and ε_θ are close to the corresponding errors for the one-dimensional case presented in Table 1, but ε_{q_x} and ε_{q_z} are one order of magnitude

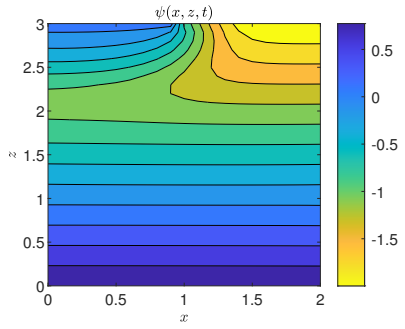


Figure 9: Pressure head solution at $t = 4.5$ hours obtained by the GRW code for the benchmark problem of recharge from a drainage trench through a silt loam soil.

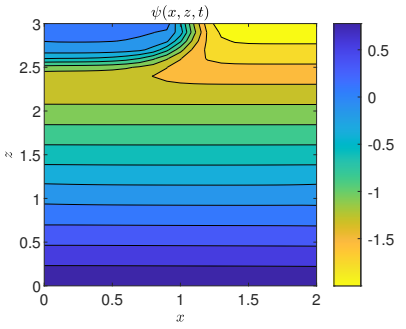


Figure 10: Pressure head solution at $t = 3$ days obtained by the GRW code for the benchmark problem of recharge from a drainage trench through a Beit Netofa clay soil.

larger than ε_q in shown in Table 1. A possible explanation could be the occurrence of the numerical diffusion in the flow TPFA code (see discussion at the end of Section 4.2.3 below). The computational times of the GRW code are 1 second and 1.6 seconds for loam and clay cases, respectively. The times of the TPFA runs, on the same computer, are one order of magnitude larger, i.e., 25 seconds and 38 seconds, respectively.

Table 4: Comparison of GRW and TPFA solutions of the flow benchmark problem.

	ε_ψ	ε_θ	ε_{q_x}	ε_{q_z}
loam	5.73e-02	4.00e-03	2.30e-01	1.04e-01
clay	5.48e-02	6.71e-04	4.73e-01	1.14e-01

4.2. GRW/BGRW solutions for fully coupled flow and transport problems

4.2.1. Code verification tests

The code verification tests for coupled flow and transport problems are conducted similarly to those for the flow solver presented in the previous subsection, by considering, along with the exact flow solution (29), the exact solution for the concentration field given by

$$c_m(x,z,t) = t\, x\, (x-1)\, z\, (z-1) \;+\; 1. \tag{34}$$

After setting $R = 0$ and $D = 1$, the coupled system of equations (16-17) is solved in the unit square for a total time $T = 1$, with source terms, initial conditions, and boundary conditions resulted from the exact solutions (29) and (34) with a new parameterization given by

$$\theta(\psi, c) = \frac{1}{1 - \psi - c/10}, \quad K(\theta(\psi)) = \psi^2. \tag{35}$$

The GRW flow-algorithm (13-15), with θ and K given by (35), is coupled with the BGRW transport-algorithm (20-23) initialized with $\mathcal{N} = 10^{24}$ particles into an alternating splitting scheme [15]. The approach alternates iterations of flow and transport solvers until the convergence criterion (3) with $\varepsilon_a = 10^{-6}$ and $\varepsilon_r = 0$ is fulfilled by the numerical solutions for both ψ and c. In order to highlight the approach to the convergence order 2, the stabilization parameters of the flow and the transport solvers are set to $L_p = L_c = 100$. The GRW results presented in Tables 5 and 6 are compared with results obtained with a TPFA solver applying the same alternating linearized splitting procedure with parameters $L_p = L_c = 1$ which ensure the convergence of order 1.

The GRW flow solver approximates the Darcy velocity by centered differences only in the interior Ω of the computational domain. Therefore, the velocity $\mathbf{q}|_{\partial\Omega}$, needed to compute the number of biased jumps from the boundary $\partial\Omega$ in the BGRW relation (20) has to be provided in some way. The straightforward approach is to compute the velocity by using an approximate forward finite difference discretization of Darcy's law. Another option is to extend on the boundary the velocity from the first neighboring interior site. Thanks to the manufactured solution (29) on which the code verification test is based, we also have the exact velocity computed analytically.

The latter allows accuracy assessments for the above approximations. We note that the GRW results for the pressure solver obtained with analytical, approximate, and extend $\mathbf{q}|_{\partial\Omega}$ are identical in the precision of three significant digits (Table 5). For the concentration solutions (Table 6), we note the remarkably good performance of approximate and extended $\mathbf{q}|_{\partial\Omega}$.

Table 5: Estimated order of convergence of the TPFA and GRW solvers: pressure solutions.

	ε_1	EOC	ε_2	EOC	ε_3	EOC	ε_4
TPFA	8.14e-03	0.93	4.27e-03	0.95	2.20e-03	0.97	1.12e-03
GRW	3.71e-03	2.02	9.18e-04	1.94	2.40e-04	1.45	8.78e-05

Table 6: Estimated order of convergence of the TPFA and GRW solvers: concentration solutions.

	ε_1	EOC	ε_2	EOC	ε_3	EOC	ε_4
TPFA	6.26e-03	0.83	3.52e-03	0.89	1.90e-03	0.91	1.01e-03
GRW (analytical $\mathbf{q}|_{\partial\Omega}$)	3.92e-03	2.00	9.78e-04	1.83	2.74e-04	1.05	1.32e-04
GRW (approximate $\mathbf{q}|_{\partial\Omega}$)	4.72e-03	1.99	1.19e-03	1.85	3.29e-04	1.17	1.46e-04
GRW ($\mathbf{q}|_{\partial\Omega}$ from int(Ω))	5.26e-03	2.00	1.31e-03	1.87	3.59e-04	1.23	1.53e-04

4.2.2. Estimates of numerical diffusion

The small errors shown in Table 6 indicate that the numerical diffusion in solving the transport step of the coupled problem does not play a significant role. This is somewhat expected for the small Péclet numbers of order Pé=10^{-2} encountered in these computations. But for the numerical setup of the benchmark problem presented in Section 4.1 and realistic transport parameters Pé can be significantly larger than unity. Therefore we proceed to estimate the numerical diffusion of the codes compared here by following the procedure used in [24].

Table 7: Estimation of numerical diffusion for BGRW, GRW and TPFA codes.

	Δx	$T/\Delta t$	Pé	ε_{D_x}	ε_{D_z}
BGRW	0.1	2	3.31	7.55e-02	2.60e-01
	0.05	9	1.65	1.90e-16	1.48e-15
	0.01	239	0.33	4.16e-16	1.02e-15
	0.005	960	0.17	2.93e-15	3.63e-15
GRW	0.1	4	3.31	1.94e-16	6.14e-16
	0.05	4	1.65	6.60e-17	8.05e-16
	0.01	19	0.33	1.94e-16	4.79e-16
	0.005	39	0.17	2.10e-15	8.92e-16
TPFA	0.1	5	3.31	9.16e-03	1.99e-01
	0.05	10	1.65	4.69e-03	9.94e-02
	0.01	50	0.33	9.58e-04	1.99e-02
	0.005	100	0.17	5.38e-04	9.89e-03

We consider the analytical Gaussian solution $c(x, z, t)$ of Eq. (17) with $\theta = 1$, $R = 0$, and constant coefficients $D = 0.001$ and $V = -0.0331$, corresponding to the Cauchy problem with a Dirac initial concentration pulse located at the coordinates (1,2.1). The constant velocity V, oriented downwards along the z-axis, is the steady-state solution of the benchmark flow problem from Section 4.1 with $K = K_{sat}$ corresponding to the loam soil, initial condition $\psi(x, z, 0) = 1 - z/3$, Dirichlet boundary conditions $\psi(x, 0, t) = 1$, $\psi(x, 3, t) = 0$, and no-flow Neumann conditions on the vertical boundaries. The initial condition $c(x, z, 0)$ is the same Gaussian function evaluated at $t = 1$ and the final time is $T = 3$. For decreasing mesh sizes Δx and Pé $= V\Delta x/D$, the number of time steps was restricted by the requirement that the support of the numerical solution does not extend beyond the boundaries $\partial\Omega$ (to mimic diffusion in unbounded domains). The effective diffusion coefficients D_x and D_z are computed from the spatial moments along the x- and z-directions of the numerical solution (see [24, Eqs. (38-41)]). The numerical diffusion is estimated by relative errors $\varepsilon_{D_x} = |D_x - D|/D$ and $\varepsilon_{D_z} = |D_z - D|/D$ averaged over the time

interval $[0, T]$. Table 7 shows that while the TPFA results are strongly influenced by the mesh size, similarly to the finite-volume results from [24], the unbiased GRW algorithm is practically unconditionally-free of numerical diffusion. The BGRW algorithm is also free of numerical diffusion provided that Pé ≤ 2 (see also Remark 3). We also note that $\Delta x = 0.05$ defines the coarsest grid acceptable for solving the benchmark problem for coupled flow and transport with BGRW and TPFA codes.

4.2.3. Fully coupled water flow and surfactant transport

In the following we solve the coupled flow and transport problem (16-17) by using the setup of the benchmark flow problem problem from Section 4.1 completed by parameters and initial/boundary conditions modeling a situation of coupled water flow and surfactant transport. The surfactant concentration in the domain Ω has a stratified distribution described by the plane $c(x, z, 0) = z/1.2$. Further, the concentration is set to $c = 1$ on the Dirichlet boundary Γ_{D_1} and to $c = 0$ on Γ_{D_2}, and no-flow Neumann conditions are imposed on the vertical boundaries.

The flow and transport are coupled in both directions through the van Genuchten-Mualem parameterization (32-33) with $\theta(\psi, c) = \theta(\gamma(c)\psi)$, where $\gamma(c) = 1/[1 - b \ln(c/a + 1)]$ models the concentration-dependent surface tension between water and air [17]. The constant parameters of $\gamma(c)$ are set to $a = 0.44$ and $b = 0.0046$ [15]. To describe a more realistic heterogeneous soil, the saturated conductivity K_{sat} is modeled as a log-normal space random function with a small variance $\sigma^2 = 0.5$ and Gaussian correlation of correlation lengths $\lambda_x = 0.1$ m and $\lambda_z = 0.01$ m in horizontal and vertical directions, respectively. The $\ln K$ field is generated by summing up 100 random periodic modes with the Kraichnan algorithm presented in [33, Appendix C.3.1.2]. The diffusion coefficient is set to a constant value, $D = 10^{-3}$ m/day, which is representative for soils and aquifers [24, 28, 33]. Following [15], the nonlinear reaction term is specified as $R(c) = 10^{-3}c/(1 + c)$. Instead of using a fixed number of time steps, as in the flow benchmark presented in Section 4.1, now we fix the total time to $T = 3$ days, set the intermediate time controlling the drainage process to $\Delta t_D = T/3$, and keep the original time steps Δt which ensure the appropriate resolution for contrasting fast and slow processes in loam and clay soils, respectively.

Preliminary tests showed that, in order to obtain an acceptable resolution of the velocity components in the benchmark setup, the unbiased GRW requires extremely fine discretizations with $\Delta x = O(10^{-5})$. Therefore the transport step is solved with the BGRW algorithm for the mesh size $\Delta x = 0.05$ suggested by the above investigations on numerical diffusion. The velocity $\mathbf{q}|_{\partial\Omega}$ on boundaries is approximated by forward finite differences.

The convergence of the flow and transport L-schemes using GRW algorithms requires relatively large linearization parameters, $L_p = L_c = 20$, for loam soil, and $L_p = L_c = 100$ for clay soil models. These are two order of magnitude larger than for the decoupled-flow benchmark presented in Section 4.1, probably due to the increased complexity of the coupled problem. By setting the tolerances of the convergence criterion (3) to $\varepsilon_a = \varepsilon_r = 5 \cdot 10^{-6}$ the convergence is achieved after about 2000 iterations for the loam soil and about 14000 iterations for the clay soil (see Figs. 11 - 14).

The results obtained by coupling the GRW-flow and BGRW-transport solvers are presented in Figs. 15-24. The randomness of K_{sat} is especially felt by the pressure distribution in the more permeable loam soil (Fig. 15), while in the clay soil the pressure remains almost stratified (Fig. 16). The same contrast is shown by the water content, with almost saturated loam soil (Fig. 17) and partially stratified saturation in the clay soil (Fig. 18). Since the Darcy velocity is proportional to the gradient of the random pressure, the heterogeneity of the advective component of the transport process is mainly manifest in the final distribution of the concentration in the loam and clay soils (compare Figs. 19 and Fig. 20). Significant differences between the loan and clay soils are also illustrated by the spatial distribution of the velocity components (Figs. 21 - 24).

The results obtained with the GRW/BGRW flow and transport solvers are compared with those provided by a TPFA code using $L_p = L_c = 1$, for both soils, and $L_c = 2L_p$. The convergence is achieved in reasonable computing times of 263 seconds (loam) and 177 seconds (clay) only when using the Anderson acceleration procedure [2, 37, 7]. Note that the GRW times on the same computer are of the same order of magnitude (526 and 178, respectively), without appealing to the acceleration procedure.

The errors for pressure, water content and velocity components shown in Table 8 are more or less similar to those for the flow benchmark problem given in Table 4. The difference of one order of magnitude between the ε_c values for the two soils can be traced back to the amount of numerical diffusion of the TPFA transport solver (see Table 7). The estimated mean Péclet number for the loam soil, Pé ≈ 1.3, is much larger than the value Pé $\approx 4 \cdot 10^{-3}$ estimated for the clay soil and can partially explain the larger ε_c value in the first case. Since the pressure equation is essentially an advection-diffusion equation with velocity given by the derivatives of the coefficient K [e.g., 11, 34], the errors ε_{q_x} and e_{q_z}, of order 10^{-1} also could be produced by numerical diffusion, in the flow solver. In the setup of the benchmark problems, for both coupled flow and transport and decoupled flow, we estimate a mean Péclet number Pé ≈ 0.9 for both loam and clay soil models (for comparison, in the one

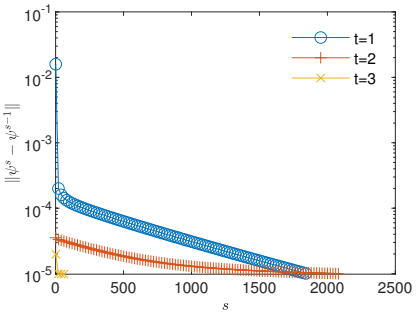


Figure 11: Convergence of the L-scheme implementation of the GRW flow solver for the loam soil problem at three time levels (in days).

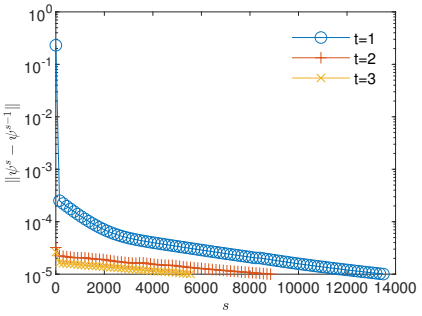


Figure 12: The same as in Fig. 11 for the clay soil problem.

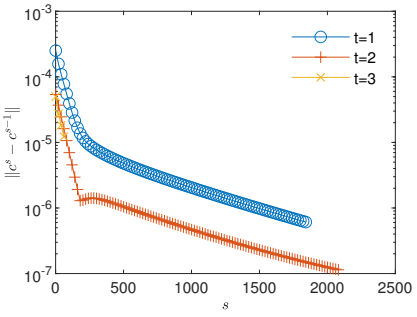


Figure 13: Convergence of the L-scheme implementation of the GRW transport solver for the loam soil problem at three time levels (in days).

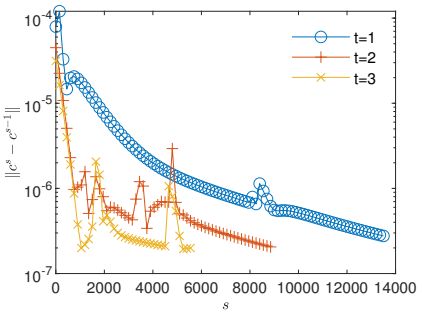


Figure 14: The same as in Fig. 13 for the clay soil problem.

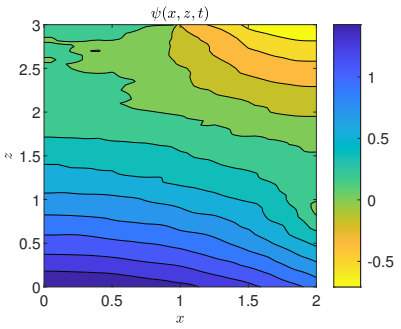


Figure 15: Pressure head solution $\psi(x,z)$ at $t = T$ for the benchmark problem of recharge from a drainage trench through a silt loam soil coupled with advection-dispersion-reaction transport.

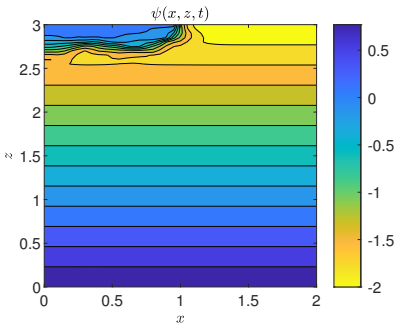


Figure 16: The same as in Fig. 15 for a Beit Netofa clay soil.

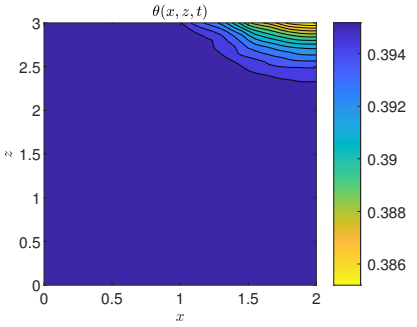


Figure 17: Water content solution $\theta(x,z)$ at $t = T$ for the benchmark problem of recharge from a drainage trench through a silt loam soil coupled with advection-dispersion-reaction transport.

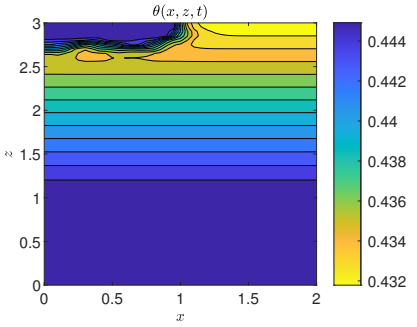


Figure 18: The same as in Fig. 17 for a Beit Netofa clay soil.

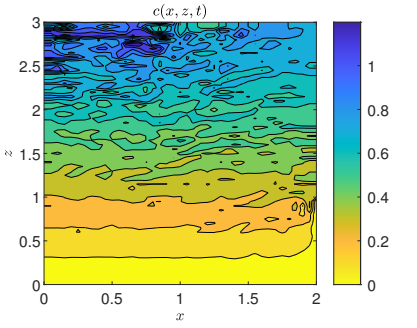


Figure 19: Concentration solution $c(x,z)$ at $t = T$ for the benchmark problem of recharge from a drainage trench through a silt loam soil coupled with advection-dispersion-reaction transport.

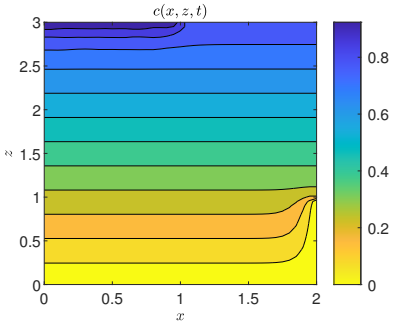


Figure 20: The same as in Fig. 19 for a Beit Netofa clay soil.

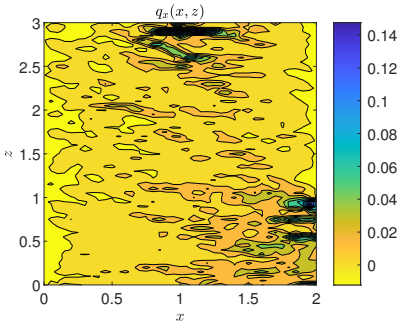


Figure 21: Horizontal water flux $q_x(x,z)$ at $t = T$ for the benchmark problem of recharge from a drainage trench through a silt loam soil coupled with advection-dispersion-reaction transport.

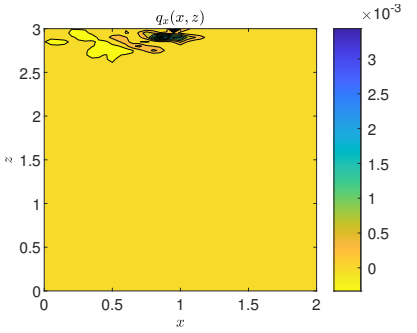


Figure 22: The same as in Fig. 21 for a Beit Netofa clay soil.

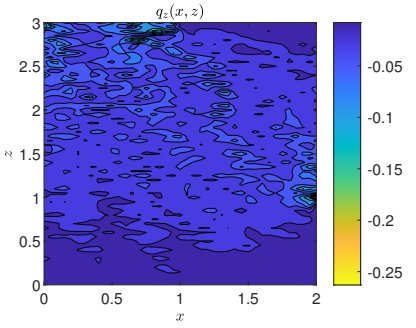


Figure 23: Vertical water flux $q_x(x, z)$ at $t = T$ for the benchmark problem of recharge from a drainage trench through a silt loam soil coupled with advection-dispersion-reaction transport.

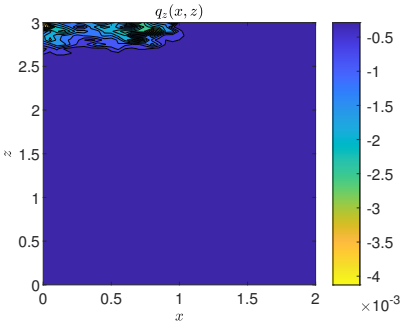


Figure 24: The same as in Fig. 23 for a Beit Netofa clay soil.

Table 8: Comparison of GRW and TPFA solutions of the coupled flow-transport benchmark problem.

	ε_ψ	ε_c	ε_θ	ε_{q_x}	ε_{q_z}
loam	2.89e-02	4.79e-01	7.25e-05	3.15e-01	2.18e-01
clay	5.95e-02	3.77e-02	7.61e-04	3.66e-01	5.36e-01

dimensional case with smaller ε_q, Pé was about 0.03 in Scenario 1 and 0.3 in Scenario 2). Since the flow and transport solvers implemented in MRST basically use the same TPFA finite volume method, we may expect that the flow solver produces a numerical diffusion comparable to that of the transport solver shown in Table 7.

A one-dimensional version of the benchmark problem for flow and surfactant transport can be readily obtained and solved with one-dimensional GRW algorithms [35, Sect. 5.2.4]. Even though the lateral heterogeneity of the two-dimensional benchmark is ignored, the main features are also revealed by the one-dimensional drainage model: the discrepancy between fast-loam and slow-clay flow and transport processes, the same intervals of variation of the solutions, and similar behavior on the vertical direction.

5. Two-dimensional GRW solutions for groundwater flow and transport at regional and field scales

For saturated aquifers ($\theta = const$) Eq. (12) reduces to a linear equation solved by the steady state hydraulic head solution in $h(x, y)$, under time independent boundary conditions. As noted in Remark 2, the GRW L-scheme (13-15) becomes, in this case, a transient scheme for the linear flow equation. In the following examples, we consider flow problems formulated in two-dimensional domains, $(x, y) \in [0, L_x] \times [0, L_y]$, with Dirichlet boundary conditions $h(0, y) = H_1$ and $h(L_x, y) = H_2$ and no-flow Neumann conditions on top and bottom boundaries. In the saturated flow regime, the transport Eq. (17) is also linear and decoupled from the linear flow equation. Decoupled transport problems can be solved by either biased- or unbiased-GRW algorithms (see Remark 4 and Section 3.2.2) on the same lattice as that used to compute the flow velocity.

5.1. Flow in heterogeneous aquifers at regional scale

For the beginning, we follow the setup for regional scale used in [14] to compare approaches for inverse modeling of groundwater flow. The domain and the boundary conditions are specified by $L_x = 4900$ m, $L_y = 5000$ m, $H_1 = 0$ m, $H_2 = 5$ m. The hydraulic conductivity K is a log-normally distributed random field defined by the mean $\langle K \rangle = 12 \cdot 10^{-4}$ m/s, the correlation length $\lambda = 500$ m, and the variance $\sigma^2 = 1$ of the $\ln K$-field. The K-field is generated, as in Section 4.2.3 above, by summing 100 random periodic modes with the Kraichnan algorithm. Besides the exponential correlation considered in [14], we also investigate the behavior of the flow solution for Gaussian correlation of the $\ln K$ field with the same correlation length, as well as in case of the smaller variance $\sigma^2 = 0.1$, for both correlation models.

The two correlation models of the $\ln K$-field are of the form $C(r) = \sigma^2 \exp[-(r/\lambda)^\alpha]$, where $r = (r_x^2 + r_y^2)^{1/2}$ is the spatial lag, the exponent $\alpha = 1$ corresponds to the exponential model, and $\alpha = 2$ to the Gaussian one. Since the correlation functions depend on spatial variables through r/λ, the computation can be done for spatial dimensions

scaled by λ, that is, fields of dimensionless correlation length $\lambda^* = 1$ and a domain $[0, L_x/\lambda] \times [0, L_y/\lambda]$. The results on the original grid are finally obtained after the multiplication by λ of the solution $h(x, y)$ and of the spatial coordinates.

The solutions $h(x, y)$ of the stationary equation (12) corresponding to $\theta = const$, for given realizations of the K-field with $\sigma^2 = 0.1$, are obtained under the initial condition $h_0(x, y)$, which is the plane defined by the Dirichlet boundary conditions $h(0, y) = 0$ and $h(L_x/\lambda, y) = H_2/\lambda$. With space steps set to $\Delta x = \Delta y = 0.2$ m, the steady state is reached after about $4 \cdot 10^5$ iterations of the GRW solver. The relative errors of the solution h obtained with the scaled geometry with respect to the solution of the unscaled problem are of the order 10^{-14}, that is, close to the machine precision [35, Sect. 6.1].

To estimate the order of convergence of the GRW scheme for this particular flow problem, we use manufactured analytical solutions provided in the Git repository https://github.com/PMFlow/FlowBenchmark and, similarly to estimations performed in Section 4.1, we compute the EOC according to (31) by successively halving the space steps from $\Delta x = \Delta y = 2 \cdot 10^{-1}$ up to $\Delta x = \Delta y = 2.5 \cdot 10^{-2}$.

Table 9: Computational order of convergence of the GRW scheme estimated according to (31).

Correlation model	σ^2	ε_1	EOC	ε_2	EOC	ε_3	EOC	ε_4
Exponential	0.1	1.35e+01	3.67	1.06e+00	1.86	2.92e-01	0.66	1.85e-01
	1	1.80e+02	3.24	1.90e+01	2.09	4.47e+00	1.96	1.15e+00
Gaussian	0.1	7.37e-02	1.98	1.87e-02	1.63	6.03e-03	1.14	2.73e-03
	1	1.31e-01	1.59	4.35e-02	1.51	1.53e-02	1.47	5.51e-03

Table 10: Computational order of convergence of the TPFA solver estimated according to (31).

Correlation model	σ^2	ε_1	EOC	ε_2	EOC	ε_3	EOC	ε_4
Exponential	0.1	4.67e+00	1.71	1.43e+00	1.95	3.70e-01	0.48	2.65e-01
	1	1.01e+02	2.23	2.14e+01	3.11	2.48e+00	0.41	1.86e+00
Gaussian	0.1	9.22e-02	2.00	2.30e-02	2.00	5.75e-03	2.00	1.44e-03
	1	1.84e-01	2.00	4.61e-02	2.00	1.16e-02	2.00	2.89e-03

We note that the EOC approach presented here differers somewhat from that used in [1, 34]. The reference solution is now the manufactured solution, instead of the solution on the finest grid, and the error norm is no longer computed after the first iteration but after large numbers of iterations (from 10^5 to more than 10^7), when the GRW solution approaches the stationarity. Due to the limited number of iterations, the solutions are not yet strictly stationary and the order of convergence may be not accurately estimated in some cases. Therefore we also use a TPFA flow solver to compute EOC values for the same scenarios.

The results presented in Tables 9 and 10 show significant differences between the two correlation models. For Gaussian correlation the errors obtained with the two approaches are relatively small in all cases. Instead, for exponential correlation, despite the strong EOC obtained after the first two refinements, the errors are extremely large for $\sigma^2 = 1$ and become smaller than one only for $\sigma^2 = 0.1$, after the second refinement of the grid. These results are consistent with those presented in [1], where similar benchmark problems were solved for a larger range of parameters of the $\ln K$ field.

5.2. Flow in conditions of random recharge

We consider in the following a flow problem formulated for the same geometry and boundary conditions as in the previous subsection, which has been used in [22] to design a new Monte Carlo approach for flow driven by spatially distributed stochastic sources. Now the hydraulic conductivity is constant, $K = 12 \cdot 10^{-4}$ m/s, and the groundwater recharge is described by a source term f in Eq. (12), modeled as a random space function of mean $\langle f \rangle = 362.912$ mm/year, log-normally distributed with exponential correlation specified by different correlation lengths and variances of the $\ln f$ field. Among different scenarios presented in[22], we consider for comparison with the present computations only the case $\lambda = 500$ m and the variance $\sigma^2 = 1$.

As in the previous subsection, we use the setup for the problem's geometry scaled by λ, for which the random recharge problem with $\sigma^2 = 1$ is solved with relative errors of the order 10^{-15} [35, Sect. 6.2].

In a first validation test, we compare the GRW and TPFA solutions of the random recharge problem on the computational domain scaled by $\lambda = 500$ m, for single-realizations of the random recharge with both exponential and Gaussian correlation of the $\ln f$ field and two variances, $\sigma^2 = 0.1$ and $\sigma^2 = 1$. The absolute and relative

differences, $\varepsilon_a = \|h^{GRW} - h^{TPFA}\|$ and $\varepsilon_r = \|h^{GRW} - h^{TPFA}\|/\|h^{TPFA}\|$, presented in Table 11 indicate a good agreement between the two approaches.

Table 11: Comparison of GRW and TPFA solutions of the random recharge problem.

Correlation model	σ^2	ε_a	ε_r
Exponential	0.1	63.44	5.97e-2
	1	101.71	9.82e-2
Gaussian	0.1	84.12	8.72e-2
	1	137.09	1.62e-2

Further, we perform statistical inferences of the mean and variance obtained from an ensemble of 100 Monte Carlo simulations within the setup of [22] for random recharge term with exponential correlation and variance $\sigma^2 = 1$. The mean and the variance of the hydraulic head h are computed as averages over realizations of the $\ln f$ field followed by spatial averages, with standard deviation estimated by spatial averaging. The results presented in Table 12 show, again, that the GRW and TPFA results are in good statistical agreement.

Table 12: Statistical moments of the hydraulic head (Monte Carlo and spatial averages).

	mean	variance
GRW	21.51±9.17	41.11±27.82
TPFA	19.74±7.84	32.09±21.33

Finally, we compare the mean and the variance estimated at the center of the computational domain by GRW and TPFA simulations with the results presented in [22]. As seen in Table 13, the mean values compare quite well but both the GRW and TPFA approaches overestimate the variance computed for the same parameters in [22, Fig. 6]. This discrepancy can be attributed either to the large errors expected for exponential correlation model (see Tables 9 and 10) or to the statistical inhomogeneity of the Monte Carlo ensemble of 100 realizations indicated by the large standard deviations shown in Table 12.

Table 13: Statistical moments of the hydraulic head (MC averages at the center of the domain).

	mean	variance
GRW	31.67	65.14
TPFA	28.31	53.39
(Passeto et al., 2011)	31.05	40.08

5.3. Flow and advection-dispersion transport in aquifers

In the following we consider an incompressible flow in the domain $[0, 20] \times [0, 10]$, driven by Dirichlet boundary conditions $h(0, y) = 1$ and $h(20, y) = 0$ and zero Neumann conditions on top and bottom boundaries. The hydraulic conductivity is a random space function with mean $\langle K \rangle = 15$ m/day, with Gaussian correlation of the $\ln K$ field, correlation length $\lambda = 1$ m, and variance $\sigma^2 = 0.1$, generated by summing 10 random modes with the Kraichnan algorithm. An ensemble of velocity fields corresponding to 100 realizations of the K field is obtained with the flow solver used in Section 5.1, for the resolution of the GRW lattice defined by space steps $\Delta x = \Delta y = 0.1$.

Further, Monte Carlo simulations of advection-diffusion are carried out using the velocity realizations and the isotropic local dispersion coefficient $D = 0.01$ m^2/day. The linear transport equation obtained by setting $\theta = const$ in Eq. (17) is solved with the unbiased GRW algorithm described in Section 3.2.2 by using $\mathcal{N} = 10^{24}$ particles to represent the concentration. The final time $T = 10$ days is chosen such that the support of the concentration does not reach the boundaries during the simulation. Hence, the Monte Carlo inferences can be compared with results of linear theory which provides first-order approximations of dispersion coefficients for small variances σ^2 [5]. In turn, such linear approximations are accurately retrieved by averaging over ensembles of particle tracking simulations of diffusion in realizations of velocity fields approximated to the first-order in σ^2

by a Kraichnan procedure [29]. Following this approach, to infer dispersion coefficients in linear approximation, we use an ensemble of 10^4 realizations of Krainchan velocity fields, computed with 100 random modes by the algorithm described in [33, Appendix C.3.2.2], and the unbiased GRW solver, with $\mathcal{N} = 10^{24}$ particles in each realization. Longitudinal and transverse "ensemble" dispersion coefficients, D_x and D_y, are computed as half the slope of the ensemble average of the second spatial moments of the concentration distribution, centered at the ensemble average center of mass [5, 24, 29]. The results presented in Fig. 25 show a that, in spite of relatively small ensemble of velocity realizations, the ensemble dispersion coefficients obtained with the 100 GRW solutions of the full flow problem are quite close to the reference linear results.

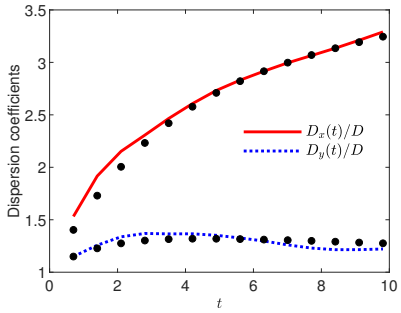


Figure 25: Dispersion coefficients estimated from GRW solutions for 100 realizations of the isotropic hydraulic conductivity K, with Gaussian correlated ln K field of variance $\sigma^2 = 0.1$ and correlation length $\lambda = 1$m, in the domain $[0, 20] \times [0, 10]$, compared to first-order results (dots).

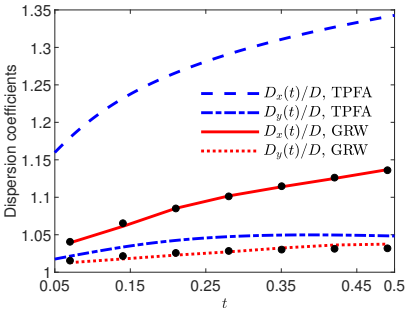


Figure 26: Comparison of dispersion coefficients obtained by GRW, TPFA, and first-order approximation (dots) fron an ensemble of 100 realizations of the isotropic hydraulic conductivity K, with Gaussian correlated ln K field of variance $\sigma^2 = 0.1$ and correlation length $\lambda = 0.1$m, in the domain $[0, 2] \times [0, 1]$.

The computation of the velocity realizations with the transient GRW flow solver requires 10^4 to 10^5 iterations to fulfill the convergence criterion (3) with tolerances $\varepsilon_a = \varepsilon_r = 5 \cdot 10^{-7}$ and about 160 seconds per realization. For the chosen discretization, $\Delta x = \Delta y = 0.1$, the unbiased GRW transport solver requires, according to (26), a relatively rough time discretization of $\Delta t = 0.5$. This leads to a total computation time of about 1.4 seconds for the estimation of the dispersion coefficients by averaging over the 100 realizations of the statistical ensemble. By comparison, the TPFA codes needs about 3.8 seconds to compute a velocity realization and about 13 seconds for a single transport realization, by using the same spatial resolution and a time step $\Delta t = 0.05$. But the TPFA estimates of the dispersion coefficients deviate by more than one order of magnitude from the linear reference solution. Since reducing the spatial steps and the local Pé to reduce the numerical diffusion dramatically increases the computational burden for the TPFA codes, we solved a rescaled problem. So, to preserve the mean and the spatial variability of the velocity field, we chose a smaller domain $[0, 2] \times [0.1]$, correlation length of the ln K field $\lambda = 0.1$, and a new Dirichlet condition, $h(0, y) = 0.1$. Now, the TPFA codes require about 60 seconds to compute one flow realization and about 3 hours for a transport realization, with $\Delta x = \Delta y = 0.001$ and $\Delta t = 0.0005$. The computation times for the GRW codes to solve the rescaled problem by using $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.07$ are practically unchanged. Figure 26 shows that the GRW estimations of the dispersion coefficients are again close to the linear approximation. Instead the TPFA results overestimate the linear approximation by 10% to 20%. The deviations of the TPFA coefficients shown in Fig. 26 are comparable with the numerical diffusion (estimated for constant velocity) in case of the longitudinal coefficient D_x but two orders of magnitude larger for the transverse coefficient D_y [35, Table 17].

6. Conclusions

The GRW schemes for simulating flow in either unsaturated or saturated porous media are equivalent to finite-difference schemes, in their deterministic implementation, or for sufficiently large numbers of particles in randomized implementations. The same, in case of BGRW solver for transport problems. Instead, the unbiased GRW is a superposition of Euler schemes for Itô equation [33], which is no longer equivalent with a finite difference scheme, unless the coefficients of the transport equation are constant. In simulations of reactive transport, GRW algorithms can use huge numbers of computational particles, even as large as the number of molecules involved in reactions, allowing simple and intuitive representations of the process.

While unbiased GRW algorithms are mainly efficient in obtaining fast solutions for large-scale transport in aquifers, BGRW solvers are appropriate for computing solutions of fully coupled flow and transport problems in soil systems with fine variation of the parameters. The algorithms are implemented as iterative L-schemes which linearize the Richards equation and describe the transition from unsaturated to saturated regime. The GRW/BGRW solutions are first-order accurate in time and second-order accurate in space. For saturated regimes, the flow solver becomes a transient scheme solving steady-state flows in aquifers.

Since the GRW algorithms are explicit schemes which do not need to solve systems of algebraic equations, they are simpler and, in some cases, faster than finite element/volume schemes. The GRW L-schemes for non-steady coupled problems for flow and transport in soils, as well as for transport simulations in saturated aquifers, are indeed much faster than the TPFA codes used as reference in this study. However, the flow solutions for saturated porous media in large domains (e.g. field or regional scale) require much larger computing time than classical numerical schemes, due to the large number of iterations needed to achieve the convergence of the transitory scheme used to compute steady-state solutions (see also a detailed analysis in [1]).

The obvious advantage of the GRW schemes is that they are practically free of numerical diffusion. This is demonstrated by the results for decoupled transport presented in Table 7. But, as shown by the discussion at the end of Section 4.2.3, the flow solvers also can be affected by numerical diffusion, which is difficult to isolate from other errors occurring in coupled flow and transport problems. Such errors are avoided by GRW algorithms, which prevent the occurrence of the numerical diffusion by using consistent definitions of the jump probabilities as functions of the coefficients of the flow and transport equations.

Acknowledgements

The authors are grateful to Dr. Emil Cătinaş for fruitful discussions on convergent sequences and successive approximation approaches. Nicolae Suciu acknowledges the financial support of the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant SU 415/4-1 – 405338726 “Integrated global random walk model for reactive transport in groundwater adapted to measurement spatio-temporal scales”. VISTA, a collaboration between the Norwegian Academy of Science and Letters and Equinor, funded the research of Davide Illiano, project number 6367, project name: “Adaptive model and solver simulation of enhanced oil recovery”.

References

[1] Alecsa, C.D., Boros, I., Frank, Nechita, M., Prechtel, A., Rupp, A., Suciu, N., 2019. Numerical benchmark study for fow in heterogeneous aquifers. Adv. Water Resour., 138, 103558. https://doi.org/10.1016/j.advwatres.2020.103558

[2] Anderson, D.G., 1965. Iterative procedures for nonlinear integral equations. J. ACM, 12(4), 547–560. https://doi.org/10.1145/321296.321305

[3] Alt, W., Luckhaus, H., 1983. Quasilinear elliptic-parabolic differential equations, Math. Z., 183(3), 311–341. https://doi.org/10.1007/BF01176474

[4] Bause, M., Knabner, P., 2004. Numerical simulation of contaminant biodegradation by higher order methods and adaptive time stepping. Comput. Visual. Sci., 7(2), 61–78. https://doi.org/10.1007/s00791-004-0139-y

[5] Bellin, A., Salandin, P., Rinaldo, A., 1992. Simulation of dispersion in heterogeneous porous formations: Statistics, first-order theories, convergence of computations. Water Resour. Res. 28(9), 2211–2227. https://doi.org/10.1029/92WR00578

[6] Bellin, A., Fiori, A., Dagan, G., 2020. Equivalent and effective conductivities of heterogeneous aquifers for steady source flow, with illustration for hydraulic tomography. Adv. Water Resour., 142, 103632. https://doi.org/10.1016/j.advwatres.2020.103632

[7] Both, J.W., Kumar, K., Nordbotten, J.M., Radu, F.A., 2019. Anderson accelerated fixed-stress splitting schemes for consolidation of unsaturated porous media. Comput. Math. Appl., 77(6), 1479–1502. https://doi.org/10.1016/j.camwa.2018.07.033

[8] Cătinaş, E., 2019. A survey on the high convergence orders and computational convergence orders of sequences. Appl. Math. Comput., 343, 1–20. https://doi.org/10.1016/j.amc.2018.08.006

[9] Catinas, E., 2020. How many steps still left to x*?, SIAM Rev., to appear.

[10] Gardner, W.R., 1958. Some steady-state solutions of the unsaturated moisture flow equation with application to evaporation from a water table. Soil Sci., 85(4), 228–232. https://journals.lww.com/soilsci/toc/1958/04000

[11] Gotovac, H., Cvetković, V., Andričevicć, R., 2009. Adaptive Fup multi-resolution approach to flow and advective transport in highly heterogeneous porous media: Methodology, accuracy and convergence. Adv. Water Resour. 32(6), 885–905. https://doi.org/10.1016/j.advwatres.2009.02.013

[12] Hajibeygi, H., Olivares, M.B., HosseiniMehr, M., Pop, S., Wheeler, M., 2020. A benchmark study of the multiscale and homogenization methods for fully implicit multiphase flow simulations. Adv. Water Resour., 143, 103674. https://doi.org/10.1016/j.advwatres.2020.103674

[13] Haverkamp, R., Vauclin, M., Touma, J., Wierenga, P. J., Vachaud, G., 1977. A comparison of numerical simulation models for one-dimensional infiltration 1. Soil. Sci. Soc. Am. J., 41(2), 285–294. https://doi.org/10.2136/sssaj1977.03615995004100020024x

[14] Hendricks Franssen, H.J., Alcolea, A., Riva, M., Bakr, M., Van der Wiel, N., Stauffer, F., Guadagnini, A., 2009. A comparison of seven methods for the inverse modelling of groundwater flow. Application to the characterisation of well catchments. Adv. Water Resour., 32(6), 851–872. https://doi.org/10.1016/j.advwatres.2009.02.011

[15] Illiano, D., Pop, I.S., Radu, F.A., 2020. Iterative schemes for surfactant transport in porous media. Comput. Geosci. https://doi.org/10.1007/s10596-020-09949-2

[16] Knabner, P., Angermann, L., 2003. Numerical Methods for Elliptic and Parabolic Partial Differential Equations. Springer, New York.
[17] Knabner, P., Bitterlich, S., Iza Teran, R., Prechtel, A., Schneid, E., (2003). Influence of Surfactants on Spreading of Contaminants and Soil Remediation. In: Jäger W., Krebs H.J. (eds) Mathematics – Key Technology for the Future. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-55753-8_12
[18] Kuzmin, D., 2009. Explicit and implicit FEM-FCT algorithms with flux linearization. J. Comput. Phys., 228(7):2517–2534. http://dx.doi.org/doi:10.1016/j.jcp.2008.12.011.
[19] Lie, K.-A., 2019. An Introduction to Reservoir Simulation Using MATLAB/GNU Octave: User Guide for the MATLAB Reservoir Simulation Toolbox (MRST). Cambridge University Press. https://doi.org/10.1017/9781108591416
[20] Liu, F., Fukumoto, Y., Zhao, X., 2020. Stability Analysis of the Explicit Difference Scheme for Richards Equation. Entropy, 22(3), 352. https://doi.org/10.3390/e22030352
[21] List, F., Radu, F.A., 2016. A study on iterative methods for solving Richards' equation. Comput. Geosci., 20(2), 341–353. https://doi.org/10.1007/s10596-016-9566-3
[22] Pasetto, D., Guadagnini, A., Putti, M., 2011. POD-based Monte Carlo approach for the solution of regional scale groundwater flow driven by randomly distributed recharge. Adv. Water Resour., 34(11), 1450–1463. https://doi.org/10.1016/j.advwatres.2011.07.003
[23] Pop, I.S., Radu, F.A., Knabner, P., 2004. Mixed finite elements for the Richards' equation: linearization procedure, J. Comput. Appl. Math., 168(1), 365–373. https://doi.org/10.1016/j.cam.2003.04.008
[24] Radu, F.A., Suciu, N., Hoffmann, J., Vogel, A., Kolditz, O., Park, C.-H., Attinger, S., 2011. Accuracy of numerical simulations of contaminant transport in heterogeneous aquifers: a comparative study. Adv. Water Resour., 34, 47–61. http://dx.doi.org/10.1016/j.advwatres.2010.09.012.
[25] Radu, F.A., Wang, W., 2014. Convergence analysis for a mixed finite element scheme for flow in strictly unsaturated porous media. Nonlinear Anal. R. World Appl., 15, 266–275. https://doi.org/10.1016/j.nonrwa.2011.05.003
[26] Radu, F.A., Kumar, K. , Nordbotten, J.M., Pop, I.S., 2018. A robust, mass conservative scheme for two-phase flow in porous media including hölder continuous nonlinearities. IMA J. Numer. Anal., 38(2), 884–920. https://doi.org/10.1093/imanum/drx032
[27] Schneid, E., Prechtel, A., Knabner, P., 2000. A comprehensive tool for the simulation of complex reactive transport and flow in soils. Land Contam. Reclamat., 8, 357–365. https://doi.org/10.2462/09670513.570
[28] Schneid, E., 2000. Hybrid-gemischte finite-elemente-diskretisierung der Richards-Gleichung. Doctoral dissertation, Naturwissenschaftliche Fakultät der Friedrich-Alexander-Universität Erlangen-Nürnberg.
[29] Schwarze, H., Jaekel, U., Vereecken, H., 2001. Estimation of macrodispersion by different approximation methods for flow and transport in randomly heterogeneous media. Transport Porous Media, 43(2), 265-287. https://doi.org/10.1023/A:1010771123844
[30] Slodicka, M., 2002. A robust and efficient linearization scheme for doubly non-linear and degenerate parabolic problems arising in flow in porous media. SIAM J. Numer. Anal., 23 (5), 1593–1614. https://doi.org/10.1137/S1064827500381860
[31] Srivastava, R., Yeh, T.C.J., 1991. Analytical solutions for one-dimensional, transient infiltration toward the water table in homogeneous and layered soils. Water Resour. Res., 27(5), 753–762. https://doi.org/10.1029/90WR02772
[32] Strikwerda, J.C., 2004. Finite Difference Schemes and Partial Differential Equations. SIAM. https://doi.org/10.1137/1.9780898717938
[33] Suciu, N., 2019. Diffusion in Random Fields. Applications to Transport in Groundwater. Birkhäuser, Cham. https://doi.org/10.1007/978-3-030-15081-5
[34] Suciu, N., 2020. Global Random Walk Solutions for Flow and Transport in Porous Media, in F.J. Vermolen, C. Vuik (eds.), Numerical Mathematics and Advanced Applications ENUMATH 2019, Lecture Notes in Computational Science and Engineering 139. Springer Nature, Switzerland. https://doi.org/10.1007/978-3-030-55874-1_93
[35] Suciu, N., Illiano, D., Prechtel, A., Radu, F.A., 2020. Global random walk solvers for fully coupled flow and transport in saturated/unsaturated porous media (extended version). arXiv:2011.12889
[36] Vamoş, C., Suciu, N., Vereecken, H., 2003. Generalized random walk algorithm for the numerical modeling of complex diffusion processes. J. Comput. Phys., 186, 527–544. https://doi.org/10.1016/S0021-9991(03)00073-1.
[37] Walker, H.F., Ni, P., (2011). Anderson acceleration for fixed-point iterations. SIAM J. Numer. Anal., 49(4), 1715–1735. https://doi.org/10.1137/10078356X

Graphic design: Communication Division, UiB / Print: Skipnes Kommunikasjon AS

uib.no

ISBN: 9788230866559 (print)
9788230841990 (PDF)