Inaugural-Dissertation

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich-Mathematischen Gesamtfakultät

der

Ruprecht-Karls-Universität

Heidelberg

vorgelegt von	Diplom-Mathematikerin **Rebecca de Cuveland (geb. Neumann)** aus Münchberg
Tag der mündlichen Prüfung	

Two-Phase Compositional Flow Simulation with Persistent Variables

Rebecca de Cuveland

Betreuer: Prof. Dr. Peter Bastian

Acknowledgements

I want to express my deepest gratitude to my main supervisor Peter Bastian[1] and my mentor Olaf Ippisch[2]. They always took the time to talk to me about my work, even when there were so many other things to do. Without their knowledge and valuable suggestions this thesis would not have been possible.

My thanks goes to my colleagues in Heidelberg for the nice work atmosphere and the support with all scientific problems.

This work would not have been possible without the DUNE library, so I want to thank all DUNE and especially fellow PDELab developers for their ongoing work for this worthwhile project.

Furthermore, I would like to gratefully acknowledge the Baden-Württemberg-Stiftung and the Heidelberg graduate school for mathematical and computational modeling. Their financial support gave me the opportunity for the research presented in this thesis.

Most of all, I want to thank my family Jan and Emilia for the patience and support during the last years.

[1]Faculty of Mathematics and Computer Science, Heidelberg University
[2]Faculty of Mathematics/Computer Science and Mechanical Engineering, TU Clausthal

Two-Phase Compositional Flow Simulation with Persistent Variables

Carbon capture and storage (CSS) is a recently discussed new technology, aimed at allowing an ongoing use of fossil fuels while preventing the produced CO_2 from being released into the atmosphere. A suitable mathematical model to simulate this process is compositional multiphase flow with equilibrium phase exchange. It is able to represent the important process of solubility trapping. One of the big problems arising in two-phase two-component flow simulations is the disappearance of the nonwetting phase, where the saturation cannot be used as independent variable.

In this thesis, a persistent variable formulation is presented, which has the important advantage that only one set of primary variables can be used for the biphasic as well as the monophasic case. Using a modified Newton solver, also developed in the course of this work, the convergence at the single-phase/two-phase interface can be greatly improved.

The persistent variable formulation is implemented in the DUNE simulation framework with capillary pressure and nonwetting phase pressure as primary variables. The presented method is verified by numerical test simulations of CO_2 injection in saline aquifers. A fine grid resolution for these large-scale simulations can only be achieved by the use of heavy parallelization. The numerical results for the recent MoMas benchmark agree with the output of other groups. For several test cases, grid convergence and scalability are analyzed numerically. The method scales well and converges with the optimal order of convergence.

Zweiphasen-Strömungssimulation unter Berücksichtigung von Löslichkeitseffekten mit persistenten Variablen

Carbon capture and storage (CSS) ist eine viel diskutierte Technologie, die freigesetztes CO_2 daran hindert, in die Atmosphäre zu gelangen, um so eine längere Nutzung fossiler Energieträger zu ermöglichen. Unter Annahme eines Gleichgewichts des Phasenaustauschs wird eine Mehrphasen-Strömung unter Berücksichtigung von Löslichkeitseffekten für die Simulation von CSS verwendet. Ein großes Problem bei der Simulation von Mehrphasen-Strömung mit Löslichkeitseffekten tritt auf, wenn eine Phase verschwindet. In diesem Fall kann die Sättigung nicht als eigenständige Variable verwendet werden.

In dieser Arbeit wird eine Formulierung mit persistenten Variablen vorgestellt. Diese Herangehensweise hat den Vorteil, dass die gleichen Primärvariablen sowohl im Zweiphasen- als auch im Einphasebereich verwendet werden können. Ein verbessertes Newton-Verfahren liefert deutlich bessere Konvergenzergebnisse für den Übergang zwischen Ein- und Zweiphasenbereich.

Dieses Verfahren ist in der DUNE-Simulationsumgebung implementiert. Dabei werden Kapillardruck und der Druck der CO_2 Phase als Primärvariablen verwendet, um die Verpressung von CO_2 in salzhaltigen Wasserschichten numerisch zu simulieren. Damit die großflächigen Rechengebiete genau aufgelöst werden können, wird die Simulation stark parallelisiert. Es wird eine sehr gute Übereinstimmung zu den Ergebnissen anderer Teilnehmer des vor kurzem durchgeführten MoMas Benchmark erreicht. Anhand mehrerer Testprobleme werden Gitterkonvergenz- und Skalierbarkeitseigenschaften des Verfahrens numerisch untersucht. Die Formulierung liefert die optimale Konvergenzordnung.

Contents

List of Figures

List of Tables

1 Introduction

Carbon Capture and Storage (CCS) describes the process of capturing carbon dioxide from power plants or other industrial sources. The compressed CO_2 is then injected into suitable geological formations with the aim to store it there permanently (see Figure 1.1). The injection takes usually place at depths of at least 750 m to ensure that the injected CO_2 is in a supercritical aggregation state, i. e., it has a very high density compared to gaseous CO_2.

Even in supercritical state, CO_2 has a lower density than the brine present in the formation. This causes an upward migration of the CO_2. Therefore, it is important, that a geological layer with a significantly lower permeability is present, to stop the upward movement. This layer is usually called a caprock.

Suitable storage sites are, e. g., deep saline aquifers or depleted gas or oil fields (see [48]). Examples for CCS pilot projects are Ketzin in Germany (since 2008, see [47]), Hontomín in Spain (since 2013, see [63]) and the first CCS site Sleipner in Norway (since 1996, see [53]).

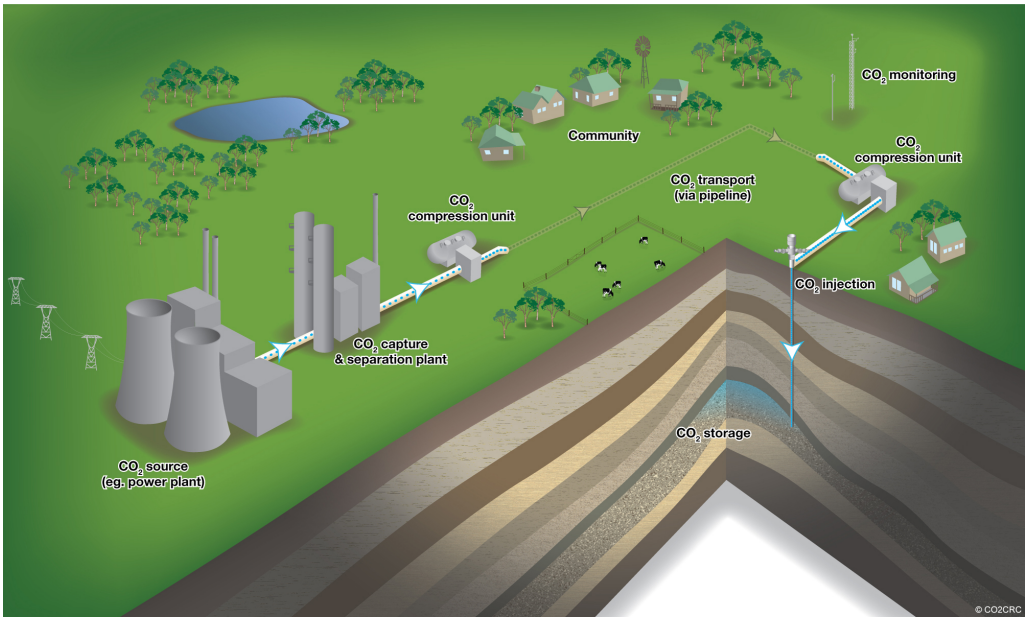


Figure 1.1: Schematic showing geological sequestration of CO_2 from a power plant (source: www.co2crc.com.au)

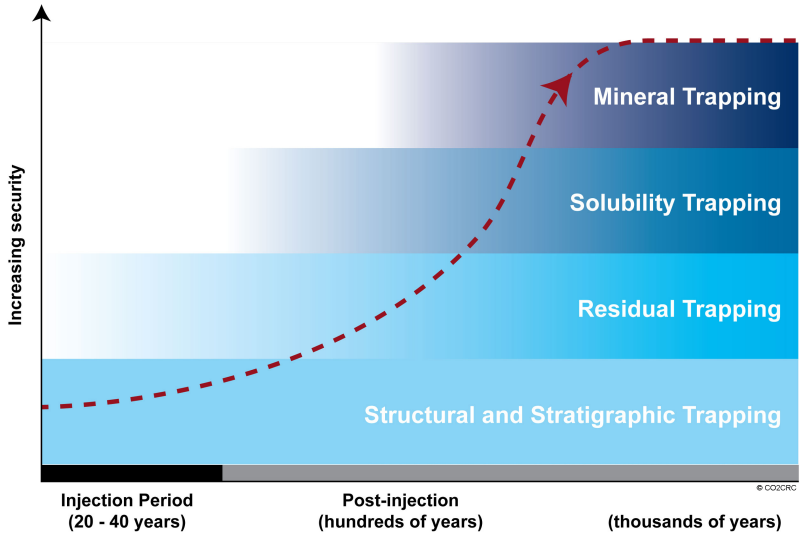


Figure 1.2: Schematic representation of CCS trapping mechanisms at different time scales (source: www.co2crc.com.au)

CCS is a highly debated topic in Germany and worldwide. Carbon Dioxide (CO_2) is the most significant greenhouse gas, which affects the atmosphere causing global warming. Many people consider CO_2 storage as an important factor in the effort to reduce the emission of greenhouse gases. With CCS the CO_2 emissions of suitable power plants can be reduced by up to 90 %. A detailed evaluation of the potential of CCS is published in the IPCC Special Report [48].

Major concerns with the CCS technology are, e. g., CO_2 leakage or migration of brine that may lead to pollution of freshwater aquifers. There is also a risk of structural failing due to large pressure peaks caused by the high injection rates. A risk assessment study for CO_2 storage can be found in [60]. One of the biggest problems especially for depleted gas or oil fields in Northern America is the existence of a lot of wells (confer the contribution of Celia in [38]). These wells provide vertical pathways to the surface for the CO_2.

Due to the risks described above, reliable simulation data is crucial for all stages of CCS projects. The mathematical model has to include all relevant physical processes during the injection and storage of CO_2.

1.1 Trapping Mechanisms

When CO_2 is injected into the subsurface several mechanisms lead to an entrapment of the CO_2 underground. All those mechanisms operate on different time scales.

Figure 1.2 demonstrates the different trapping mechanisms and their temporal appearance. The dominant and most important trapping mechanism directly after injection is the **structural trapping**. CO_2 has a lower density than water and percolates upwards

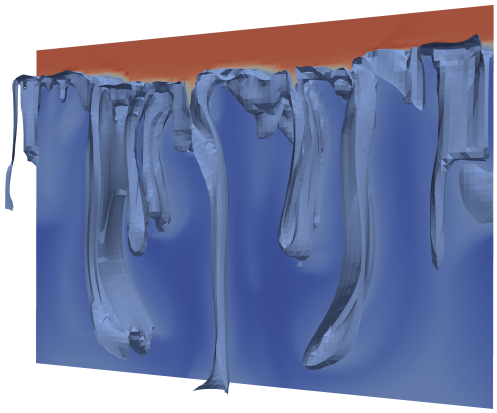


Figure 1.3: CO_2 fingering caused by density driven flow (picture from [19])

(a) Residual Trapping **(b)** Mineral Trapping

Figure 1.4: Residual and mineral trapping (source: www.co2captureproject.com)

through the porous medium until it reaches the caprock. This impermeable layer keeps the CO_2 from further upward movement.

During the movement of the CO_2 phase some of the CO_2 is left behind as disconnected, immobile droplets due to capillary forces. This process is called **residual** or **capillary trapping** (see Figure 1.4a).

CO_2 dissolves into the brine, that is already present in the subsurface. The amount of dissolved CO_2 depends on pressure, temperature and the salinity of the brine. This is called **solubility trapping**. Additionally the density of brine with dissolved CO_2 is higher than that of pure brine. This leads to convective flow and gravity fingers of dense fluids that sink into the underlying brine. Figure 1.3 shows gravity fingers from numerical experiments conducted by Burchardt in his diploma thesis [19].

Over a long time period CO_2 could also bind with the surrounding rock and form solid carbonate minerals, called **mineral trapping** (see Figure 1.4b). The rate of mineralization is heavily dependent on the chemistry of rock and water at the storage site.

1.2 Challenges in Modeling Compositional Multiphase Flow

A very important part of modeling CCS is to decide on the capability and complexity of the mathematical model. A simple model is immiscible two-phase flow which is a system of two coupled nonlinear partial differential time-dependent equations. It describes the flow of two immiscible phases in a porous medium (the two phases for modeling CCS are water and CO_2). However, the drawback of this model is, that it does not include solubility. As solubility trapping is an important factor in determining the storage capability of a given site this simple model is not chosen for this work.

A suitable mathematical model describing CO_2 injection in geologic reservoirs including solubility trapping is *two-phase two-component flow*. As an extension of immiscible two-phase flow it is also a system of two partial differential equations. In contrast to the latter model each phase consists of two components (again water and CO_2). The components are exchanged across the phases and with that this model takes into account solubility trapping. Residual trapping is included in both models by the shape of the water retention curve (described in Section 3.1). Mineral trapping is not considered in this work.

For both models two primary variables have to be chosen. Together with a number of additional algebraic relations the primary variables close the system. For immiscible two-phase flow a standard choice of primary variables is the pressure of one phase and the saturation of the other phase. A great challenge in modeling compositional multiphase flow with the assumption of equilibrium phase exchange is the treatment of disappearing phases, because the saturation cannot be used as a primary variable. In the last years this topic has gained much attention, and many different papers show approaches to deal with this problem. Chapter 4 of this work presents the different methods for the treatment of disappearing phases. A special method is presented, which uses nonwetting phase pressure and capillary pressure as primary variables.

A further challenge in modeling CCS is the fact that huge domains and very large time spans (up to thousands of years) have to be taken into account. Nordbotten and Celia [31] propose upscaled models on 2D grids for CO_2 injection. For upscaled models accuracy must be sacrificed to obtain improvements in computational speed. In this work another approach is chosen. Heavy parallelization is used as a possibility to speed up the computation of simulations with a fine grid resolution in 3D (confer Chapter 9).

1.3 Outline

In this work, a special set of primary variables is chosen that is valid for the monophasic as well as the biphasic case and can easily handle the disappearance of one phase. As an advantage of this approach, the chosen variables are continuous over material heterogeneities if both phases are present. The mathematical model for this is presented in Chapter 2.

To simulate CSS, constitutive relations between physical properties like pressure and density are necessary. The selected models are summarized in Chapter 3.

Various approaches to choosing primary variables and to deal with the phase appearance problem are presented in Chapter 4.

In Chapter 5 crucial parts of the numerical solution and implementation of the model are discussed.

A large part of this work consists of the numerical simulation of different test cases and the evaluation of the results. In Chapter 6–9, various test cases from 1D to 3D are investigated.

Major contributions of this work are:

- Development of a persistent variable formulation for two-phase two-component flow.
- Dimension independent implementation of CO_2 injection in the DUNE framework.
- Development of an improved Newton solver for compositional two-phase flow.
- Parallel simulation of large scale CO_2 injection examples

Some material presented here has been previously published in [51]. These parts are cited in the following without explicit mention.

2 Mathematical Model of Isothermal Two-Phase Two-Component Flow

In this chapter, a system of partial differential equations describing two-phase two-component flow in a porous medium is derived. For meaningful results, it is important that the mathematical model for carbon capture and storage (CCS) supports solubility trapping. Two-phase two-component flow, the model presented here, does this by taking the composition of the phases into account.

For the sake of simplicity the temperature is assumed to be constant. However, thermodynamic effects can be included in the model in a straightforward manner. The salinity of the water phase is also considered to be constant.

2.1 Fundamental Terms

Different length scales have to be taken into account for modeling flow in porous media. On the macroscopic scale for example regions with finer and coarser sand can be identified. On the microscopic scale one can see individual sand grains. The smallest scale is the molecular scale, where single molecules can be identified. For a continuous model on the macroscopic scale possible heterogeneities and characteristics on the smaller scale have to be taken into account. In this work the averaging procedure by Bear [10] is used, where an average value for each point in the continuum is determined through a representative elementary volume (REV) on the microscopic scale. All quantities defined in this section are summarized in Table 2.1.

A porous medium consists of a solid part and interconnected void space called pores, that may be filled with different fluids. In a porous medium the **porosity** $\phi(x)$ is defined as the ratio between the volume of the void space and the total volume of a given REV (see [5]). Per definition the value of $\phi(x)$ lies between 0 and 1, e. g., gravel typically has a porosity of about 0.3 and a typical value for sandstone is 0.1. In this work, ϕ is assumed to be constant throughout the domain.

The fluids in the void space establish one or more phases. The ratio between the volume of phase α and the total volume of pore space in a given REV is defined as $S_\alpha(x,t)$, the **saturation** of phase α. From this definition it is clear that:

$$0 \leq S_\alpha(x,t) \leq 1, \quad \sum_\alpha S_\alpha(x,t) = 1. \tag{2.1}$$

For compositional flow, each phase consists of several components. In this work two-phase two-component flow is modeled, so there are two phases $\alpha \in \{\mathrm{w}, \mathrm{n}\}$, wetting and nonwetting, and two components $\kappa \in \{\mathrm{a}, \mathrm{b}\}$, wetting and nonwetting component. In the context of CCS the wetting phase is equivalent to the water phase and the nonwetting phase to the CO_2 phase. CO_2 can dissolve into the water phase and the water phase consists of the two components water and CO_2. Correspondingly the CO_2 phase also consists of the CO_2 component and a small water component.

The **mass fraction** X_α^κ of component κ in phase α is the ratio of the mass of component κ in phase α and the mass of phase α in a REV. For two components, the **molar fraction** x_α^κ, which is used in the mathematical model, is derived from the mass fraction by

$$x_\alpha^\mathrm{b} = \frac{X_\alpha^\mathrm{b} M^\mathrm{a}}{X_\alpha^\mathrm{b} M^\mathrm{a} + (1 - X_\alpha^\mathrm{b}) M^\mathrm{b}}$$

where M^κ is the molar mass of component κ. By definition, the molar fractions in one phase sum up to one:

$$x_\alpha^\mathrm{a} + x_\alpha^\mathrm{b} = 1. \tag{2.2}$$

The **absolute permeability** K measures the ability of a fluid to flow through a porous medium. The tensor K solely depends on the porous medium, not the fluid. If the porous medium has a preferred flow direction, K is anisotropic. In this work, the porous medium is considered isotropic, so the value of K is the same in every direction.

If more than one phase flows in the porous medium, the flow of one phase is inhibited by the presence of another one. This is measured by $k_{\mathrm{r}\alpha}$, the **relative permeability** of phase α, which depends on the saturation of the phase. If only phase α is present, $k_{\mathrm{r}\alpha} = 1$. Models for the relative permeability based on experimental data are described in Chapter 3.

The **molar density** ρ_α is defined as the number of moles per volume in phase α. The density depends on pressure and temperature and also on the composition of the phase. For example, water with dissolved CO_2 is more dense than pure water, which results in a density-driven convection.

The **viscosity** μ_w describes the resistance of a fluid to deformation by shear stress. Dynamic viscosity heavily depends on the pressure and temperature. The influence of the composition of the phase is not considered in this work.

description	symbol	SI unit
porosity	ϕ	
wetting and nonwetting phase pressure	p_w, p_n	[Pa]
wetting and nonwetting phase saturation	S_w, S_n	
wetting and nonwetting phase mass density	$\rho_\mathrm{mass,w}$, $\rho_\mathrm{mass,n}$	$[\mathrm{kg\,m^{-3}}]$
wetting and nonwetting phase molar density	ρ_w, ρ_n	$[\mathrm{mol\,m^{-3}}]$
wetting and nonwetting phase viscosity	μ_w, μ_n	$[\mathrm{Pa\,s^{-1}}]$
molar fraction of component in wetting phase	x^a_w, x^b_w	
molar fraction of component in nonwetting phase	x^a_n, x^b_n	
relative permeability of wetting and nonwetting phase	k_rw, k_rn	
absolute permeability	K	$[\mathrm{m^2}]$
molar mass of wetting and nonwetting component	M^a, M^b	$[\mathrm{g\,mol^{-1}}]$

Table 2.1: Quantities in two-phase two-component flow

2.2 Balance Equations

2.2.1 Phase Velocities

Darcy's Law is a constitutive equation that describes flow through a porous medium. It was first determined experimentally. It can also be derived by averaging techniques from the Navier-Stokes equation. The momentum conservation of Navier-Stokes on the microscopic scale reduces to Darcy's Law on the macroscopic scale (confer [36] for the derivation). The phase velocities u_α are given by an extended Darcy's Law:

$$\begin{aligned} u_\mathrm{w} &= -K\frac{k_\mathrm{rw}}{\mu_\mathrm{w}}(\nabla p_\mathrm{w} - \rho_\mathrm{mass,w}\cdot g), \\ u_\mathrm{n} &= -K\frac{k_\mathrm{rn}}{\mu_\mathrm{n}}(\nabla p_\mathrm{n} - \rho_\mathrm{mass,n}\cdot g), \end{aligned} \tag{2.3}$$

where g is the gravity vector.

2.2.2 Diffusive Fluxes

Following Fick's Law, the diffusive flux of a component κ in the phase α is given by

$$j^\kappa_\alpha = -D^\kappa_{\mathrm{pm},\alpha}\,\rho_\alpha \nabla x^\kappa_\alpha, \tag{2.4}$$

where $D^\kappa_{\mathrm{pm},\alpha}$ is the diffusion coefficient of component κ in phase α in a porous medium.

In many previous works, such as [21] and [16], it is assumed that

$$j^\mathrm{a}_\alpha + j^\mathrm{b}_\alpha = 0 \tag{2.5}$$

holds. This assumption is also used in this work, so instead of four only two diffusion coefficients are necessary.

2.2.3 Two-Phase Two-Component Flow

For the two-phase two-component model, chemical reactions inside each phase are neglected. Local equilibrium phase exchange of the components in the phases is assumed. The conservation of the amount of substance of each component is given by

$$\phi\partial_t(\rho_\alpha\, x_\alpha^\kappa S_\alpha) + \nabla\cdot(\rho_\alpha\, x_\alpha^\kappa u_\alpha + j_\alpha^\kappa) - q_\alpha^\kappa - r_\alpha^\kappa = 0$$

for each component κ in each phase α. Here q_α^κ is the source/sink term for component κ in phase α and r_α^κ the term that models the exchange of mass of component κ with other phases. The equation states that the flow of component κ in phase α over the surface of the domain together with the sources or sinks within the domain equals the change of molar mass of component κ in phase α.

It is assumed that no intraphase chemical reactions take place, that means $\sum_\alpha r_\alpha^\kappa = 0$ (confer [5]). By summing up the balance equation for one component over all phases the reaction terms cancel out and the final form of the two-phase two-component model in porous media is achieved (see also [54] for a detailed derivation):

$$\begin{aligned}
&\phi\partial_t(\rho_\mathrm{w}\, x_\mathrm{w}^\mathrm{a} S_\mathrm{w} + \rho_\mathrm{n}\, x_\mathrm{n}^\mathrm{a} S_\mathrm{n}) + \nabla\cdot(\rho_\mathrm{w}\, x_\mathrm{w}^\mathrm{a} u_\mathrm{w} + \rho_\mathrm{n} x_\mathrm{n}^\mathrm{a} u_\mathrm{n}) + \nabla\cdot(j_\mathrm{w}^\mathrm{a} + j_\mathrm{n}^\mathrm{a}) - q^\mathrm{a} = 0,\\
&\phi\partial_t(\rho_\mathrm{w}\, x_\mathrm{w}^\mathrm{b} S_\mathrm{w} + \rho_\mathrm{n}\, x_\mathrm{n}^\mathrm{b} S_\mathrm{n}) + \nabla\cdot(\rho_\mathrm{w}\, x_\mathrm{w}^\mathrm{b} u_\mathrm{w} + \rho_\mathrm{n} x_\mathrm{n}^\mathrm{b} u_\mathrm{n}) + \nabla\cdot(j_\mathrm{w}^\mathrm{b} + j_\mathrm{n}^\mathrm{b}) - q^\mathrm{b} = 0,
\end{aligned} \tag{2.6}$$

with the phase velocities u_α as given by Equation (2.3) and q^κ being the source/sink term for the component. To complete System (2.6), suitable initial and boundary conditions have to be chosen. Example conditions for different test setups are presented in Chapters 6–9. An expansion of this model including kinetic interphase mass transfer that does not assume a local equilibrium phase exchange can, e. g., be found in [35].

Equations (2.6) represent an isothermal model, so the temperature in the whole simulation domain is assumed to be constant. If required, a variable temperature can be included in

symbol	variable	count
p_α	phase pressures	2
S_α	phase saturations	2
$\rho_{\mathrm{mass},\alpha}$	mass densities	2
ρ_α	molar mass densities	2
μ_α	phase viscosities	2
x_α^κ	molar fractions of components in phases	4
$k_{\mathrm{r}\alpha}$	relative permeabilities of phases	2
j_α^κ	diffusive fluxes	4
u_α	phase velocities	2
sum		22

Table 2.2: Number of unknowns for two-phase two-component flow

relations		count
component mass balance summed over phases	(2.6)	2
extended Darcy's law	(2.3)	2
$\sum_\alpha S_\alpha = 1$	(2.1)	1
$S_\mathrm{w} = S_\mathrm{w}(p_\mathrm{c} = p_\mathrm{n} - p_\mathrm{w})$	Section 3.1	1
$k_{\mathrm{r}\alpha} = k_{\mathrm{r}\alpha}(S_\alpha)$	Section 3.1	2
$\sum_\kappa x_\alpha^\kappa = 1$	(2.2)	2
$x_\alpha^\kappa = x_\alpha^\kappa(p_\mathrm{n}, T, s_\mathrm{sal})$	Section 3.2	2
$\rho_\mathrm{mass,w} = \rho_\mathrm{mass,w}(x_\mathrm{w}^\mathrm{b}, T), \rho_\mathrm{mass,n} = \rho_\mathrm{mass,n}(p_\mathrm{n}, T)$	Section 3.3	2
$\rho_\mathrm{w} = \rho_{\mathrm{mass},\alpha}/(x_\alpha^\mathrm{b} M^\mathrm{b} + x_\alpha^\mathrm{a} M^\mathrm{a})$	(3.3)	2
$\mu_\mathrm{w} = \mu_\mathrm{w}(T)$, $\mu_\mathrm{n} = \mu_\mathrm{n}(p_\mathrm{n}, T)$	Section 3.3	2
$\sum_\kappa j_\alpha^\kappa = 0$	(2.5)	2
$j_\alpha^\kappa = -D_{\mathrm{pm},\alpha}^\kappa \rho_\alpha \nabla x_\alpha^\kappa$, $D_{\mathrm{pm},\alpha} = D_{\mathrm{pm},\alpha}(\phi, S_\alpha, D_\alpha^\kappa)$	(2.4), (3.4)	2
sum		22

Table 2.3: Constitutive relations for two-phase two-component flow

the model in a straightforward manner (see, e. g., [11]). This also holds for the salinity of the water s_sal, that is supposed to be constant in this work.

Table 2.2 lists all unknowns. Porosity ϕ, absolute permeability K, temperature T and salinity of water s_sal are constants in this work. Molar masses M^a and M^b are physical constants depending on the specific components used for the model. The two phase pressures are related through the capillary pressure $p_\mathrm{c} = p_\mathrm{n} - p_\mathrm{w}$. In Table 2.3 all algebraic relations are summarized. Having the same number of unknowns and relations, these relations are sufficient to determine all unknowns.

Chapter 3 discusses certain algebraic relations for a water CO_2 system in detail. If the model uses other components than water and CO_2, some of the relations in Table 2.3 have to be adapted to the physical properties of these components and dependencies on other variables may change.

For the two equations (2.6), one chooses two independent primary variables. With the algebraic relations in Table 2.3 all other variables (called secondary or derived variables) are derived from the primary variables. The choice of the primary variables is crucial for the efficient numerical solution of the equation system. It is discussed in detail in Chapter 4.

In this Chapter a suitable mathematical model for CCS is derived, and all important physical properties are introduced. Two-phase two-component flow is preferred over the simple two-phase flow, because the model is able to support solubility trapping with the composition of the phases.

3 Constitutive Relations for a Water/CO2 System

In this thesis most of the test setups are taken from the context of CCS, where a liquid water phase and a liquid, gaseous or supercritical CO_2 phase are present. The components are water and CO_2. For meaningful simulation results, the physical properties of CO_2 and water are essential. Typically, experimental data is gathered and fitted to a curve. In the following, different functions and equation of state (EOS) relating several physical quantities are presented and their dependence on other variables is given.

3.1 Capillary Pressure Saturation Curves and Relative Permeabilities

For the relation between capillary pressure and saturation

$$S_{\mathrm{w}} = S_{\mathrm{w}}(p_{\mathrm{c}}) = S_{\mathrm{w}}(p_{\mathrm{n}} - p_{\mathrm{w}}),$$

there exist different approaches.

In this work, two important models are used for capillary pressure-saturation curves: *Brooks-Corey* and *van Genuchten*. Both of them also provide a constitutive relation between relative permeability and saturation.

Brooks-Corey model (see [18], [5])

In the Brooks-Corey model, the constitutive relation between saturation and capillary pressure is:

$$\begin{aligned} S_{\mathrm{w}}(p_{\mathrm{c}}) &= \left(\frac{p_{\mathrm{entry}}}{p_{\mathrm{c}}}\right)^{-\lambda} \\ S_{\mathrm{w}} &= 1 \qquad\qquad \text{if } S_{\mathrm{w}}(p_{\mathrm{c}}) > 1, \end{aligned}$$

where the parameters have to be chosen depending on the characteristics of the porous media. For CCS in saline aquifers an adequate example is the Brooks-Corey curve from [11] with values $\lambda = 2$ and an entry pressure $p_{\mathrm{entry}} = 1000\,\mathrm{Pa}$. The entry pressure is the minimum pressure that has to be applied before the nonwetting phase appears (i. e., the minimum pressure before $S_{\mathrm{n}} > 1$). Figure 3.1 shows the resulting curve.

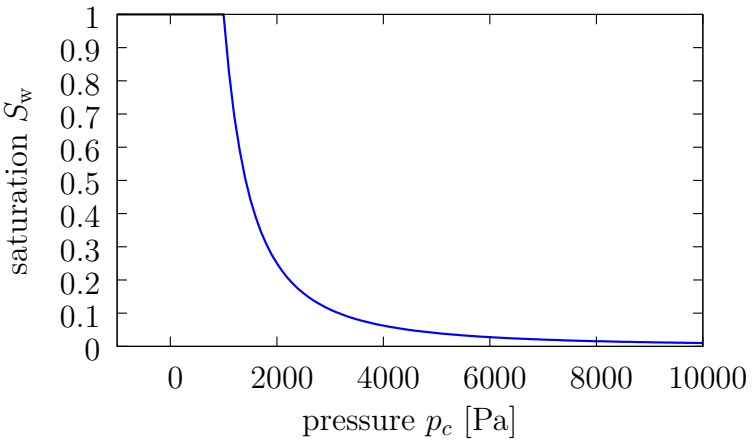


Figure 3.1: The Brooks-Corey capillary pressure/saturation function for parameters $\lambda = 2$ and $p_{\text{entry}} = 1000\,\text{Pa}$. Notice the steep descent at the entry pressure p_{entry}

The special form of the curve (the steep descent at the entry pressure) presents numerical challenges, as discussed in detail in Section 6.3.

The model also provides a constitutive relation between the saturation and the relative permeabilities (see Figure 3.2 for the resulting curve with $\tau = 0.5$):

$$k_{\text{rw}}(S_{\text{w}}) = S_{\text{w}}^{2+\tau+\frac{2}{\lambda}}$$
$$k_{\text{rn}}(S_{\text{n}}) = S_{\text{n}}^{\tau}\left(1-(1-S_{\text{n}})^{\frac{1+\lambda}{\lambda}}\right)^2$$

van Genuchten model (see [32], [5])

The van Genuchten model predicts the saturation and capillary pressure relationship as:

$$S_{\text{w}}(p_{\text{c}}) = \left(1+(\alpha_{\text{VG}}\,p_{\text{c}})^n\right)^{-m}$$
$$S_{\text{w}} = 1 \qquad \text{if } S_{\text{w}}(p_{\text{c}}) > 1.$$

The parameters for the van Genuchten curve in Figure 3.3 are $\alpha_{\text{VG}} = 6.5 \cdot 10^{-4}\,\text{Pa}^{-1}$, $n = 6$ and $m = \frac{5}{6}$. These parameters were chosen to obtain a similar curve like for the Brooks-Corey model above.

The constitutive relation between the saturation and the relative permeabilities for van Genuchten is (see Figure 3.4):

$$k_{\text{rw}}(S_{\text{w}}) = S_{\text{w}}^{\tau}\left(1-\left(1-(S_{\text{w}})^{\frac{1}{m}}\right)^m\right)^2$$
$$k_{\text{rn}}(S_{\text{n}}) = S_{\text{n}}^{\tau}\left(1-(1-S_{\text{n}})^{\frac{1}{m}}\right)^{2m}$$

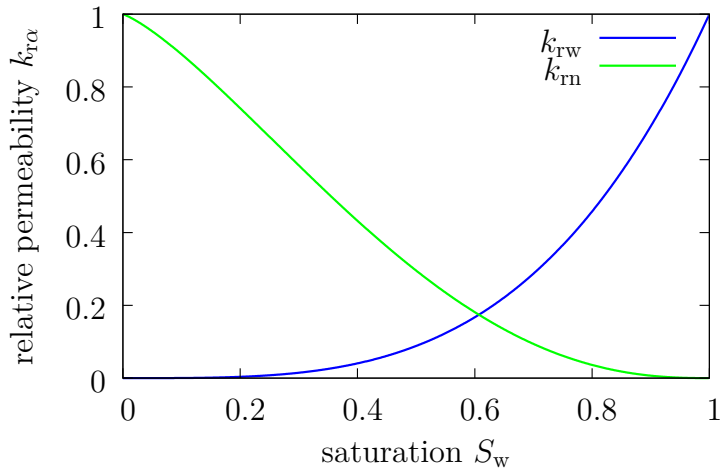


Figure 3.2: The Brooks-Corey relative permeability/saturation function for parameters $\lambda = 2$, $\tau = 0.5$ and $p_{entry} = 1000\,\mathrm{Pa}$

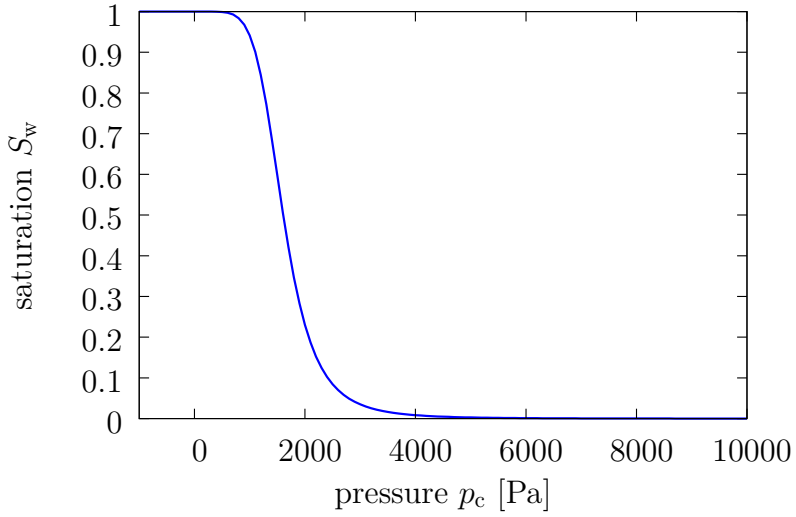


Figure 3.3: The van Genuchten capillary pressure/saturation function for parameters $\alpha_{VG} = 6.5 \cdot 10^{-4}\,\mathrm{Pa}^{-1}$, $n = 6$ and $m = \frac{5}{6}$

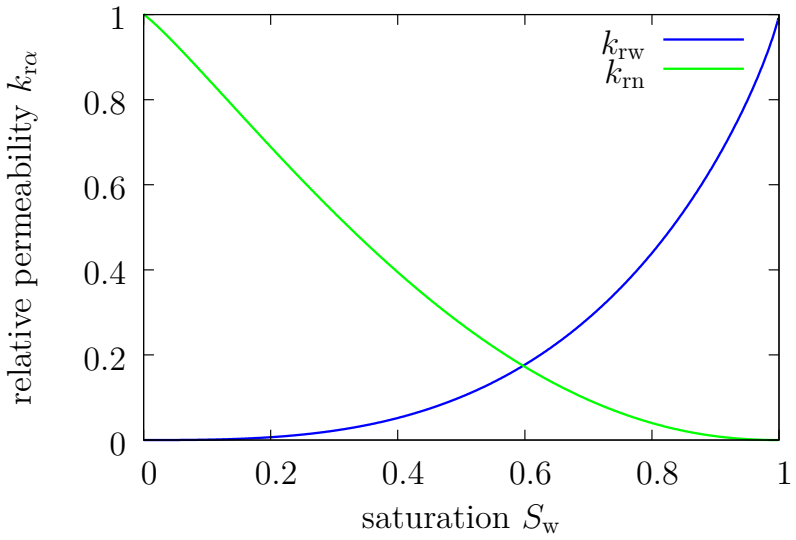


Figure 3.4: The van Genuchten relative permeability/saturation function for parameters $\tau = 0.5$, $n = 6$ and $m = \frac{5}{6}$

The van Genuchten model does not have an explicit entry pressure like the Brooks-Corey model. From definition it is evident that $p_c = 0$ is the minimum pressure required before S_w drops below 1. So the corresponding entry pressure for the van Genuchten model is $p_{entry} = 0$.

Both models can be modified to include residual saturations $S_{\alpha,res}$. The saturation of phase α cannot be reduced below $S_{\alpha,res}$. By using the effective saturation:

$$\bar{S}_\alpha = \frac{S_\alpha - S_{\alpha,res}}{1 - \sum_\beta S_{\beta,res}}$$

instead of the saturation in the equations above the effect of residual trapping can be included into the models. In general the capillary pressure saturation curves are different for a drainage or imbibition cycle. For this hysteresis effect the history of the pressure evolution has to be taken into account. Hysteresis is not part of the simulations in this work.

For numerical simulation, the water retention curves have to be evaluated several times in each cell in every time step. Because of the expensive power function evaluations this is quite time consuming. A solution for this is the use of interpolation tables (see Section 5.7).

3.2 Solubility of Components

There are several approaches to quantify the amount of dissolved CO_2 in water and water in CO_2. For all EOS, the authors try to fit curves to experimental data points. The solubility of CO_2 in water is much higher than the solubility of water in CO_2. Therefore less complex EOS give good results for the solubility of CO_2 but only use rough approximations for the water solubility.

A very simple approach is Henry's law:

$$X_{\mathrm{w}}^{\mathrm{b}} = \frac{H}{\rho_{\mathrm{w}}} p_{\mathrm{n}}^{\mathrm{b}}. \tag{3.1}$$

Here H is the Henry constant that depends on the temperature, the solute and solvent of the system. Assuming the amount of CO_2 in water can be neglected, the partial pressure of the nonwetting component in the nonwetting phase is $p_{\mathrm{n}}^{\mathrm{b}} = p_{\mathrm{n}}$. This very simple, linear relationship between the mass fraction and the nonwetting phase pressure is only valid for very low solute mass fractions (see [4]). Though, for CCS the pressure reaches relatively high values and with that high solubilities. Therefore, Henry's law is not used for the CO_2 simulations. The advantage of Henry's law is the easy implementation, thus it is used for the benchmark in Appendix A.

The EOS of Duan and Sun (confer [27]) is one example for an EOS with good results for the solubility of CO_2 in water. In this work, the EOS by Spycher and Pruess (see [59] and [58]) is preferred because it also represents the solubility of water in CO_2 well. Figures 3.6 and 3.7 show the solubility curves for different temperatures.

For a CO_2 water system the solubility of the components is influenced by the pressure p_{n}, the temperature T of the system and the salinity s_{sal} of water:

$$x_{\mathrm{w}}^{\mathrm{b}} = x_{\mathrm{w}}^{\mathrm{b}}(p_{\mathrm{n}}, T, s_{\mathrm{sal}}), \quad x_{\mathrm{n}}^{\mathrm{a}} = x_{\mathrm{n}}^{\mathrm{a}}(p_{\mathrm{n}}, T, s_{\mathrm{sal}}). \tag{3.2}$$

The complete EOS by Spycher and Pruess is described in detail in [58]. First the Redlich Kwong cubic equation is solved to determine the molar volume of the compressed gas phase. After that the solubilities can be computed via a combination of polynomials and exponential functions.

Figure 3.5 depicts a phase diagram for pure CO_2. The different aggregation states depend on pressure and temperature. For temperatures below the critical temperature $T_{\mathrm{crit}} = 304.15\,\mathrm{K}$, the state of the CO_2 changes from gaseous to liquid with rising pressure. For temperatures above T_{crit} the CO_2 is gaseous up to the saturation pressure and then changes to a supercritical state. Supercritical CO_2 has properties between those of gas and liquid.

The state of the CO_2 also has an influence on the solubility. Figure 3.6 shows, that the solubility of CO_2 in the water phase increases fast with rising pressure up to the saturation pressure, above which it rises with a smaller rate. For temperatures below the critical temperature $T_{\mathrm{crit}} = 304.15\,\mathrm{K}$, the state of the CO_2 changes from gaseous (below saturation

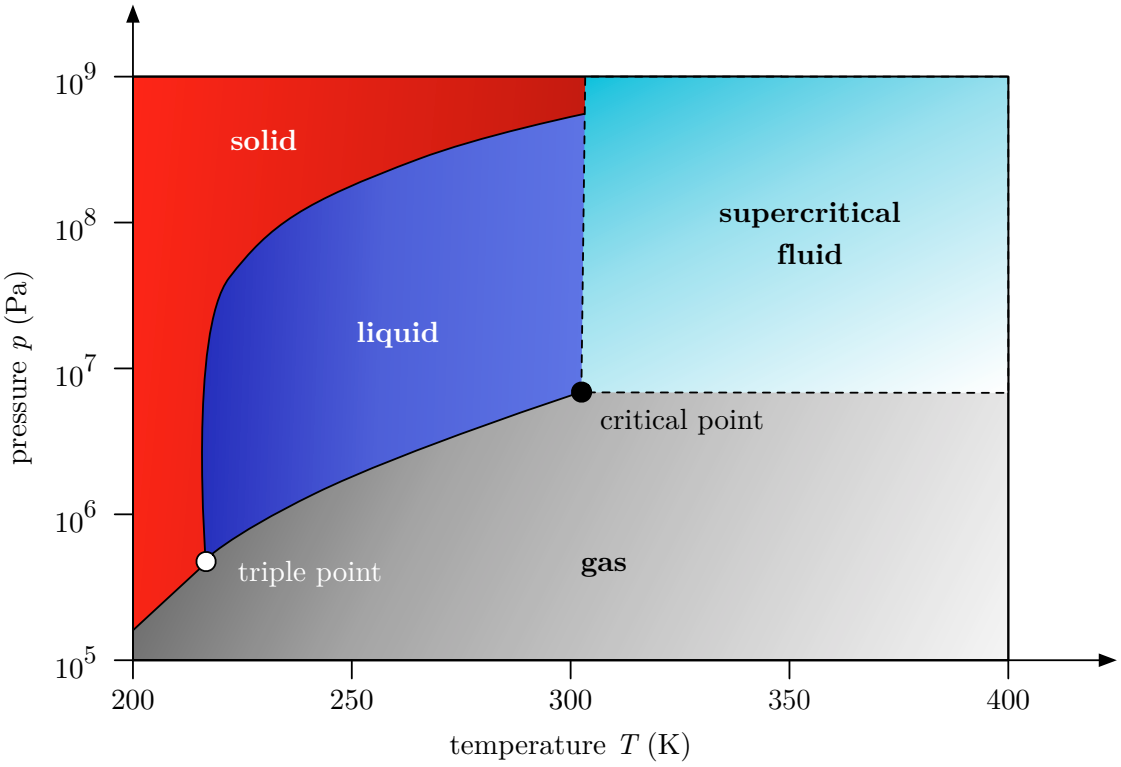


Figure 3.5: Pressure temperature phase diagram for CO_2

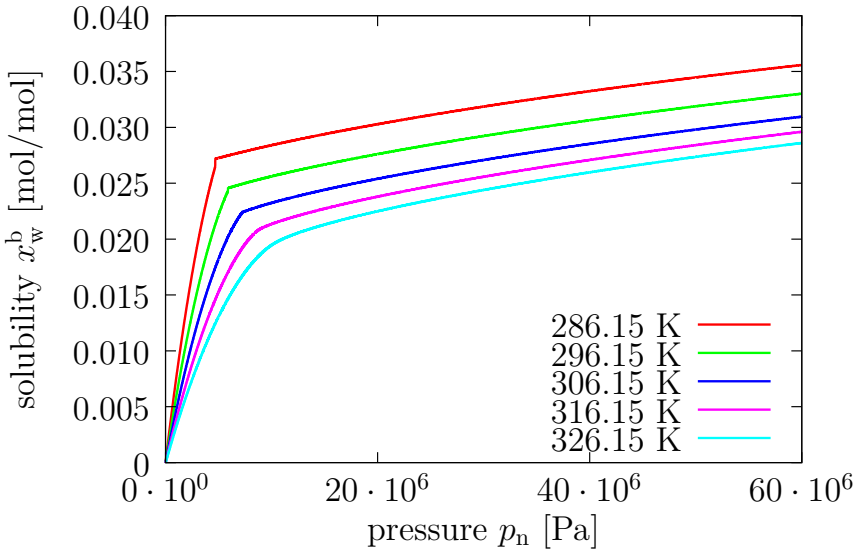


Figure 3.6: x_w^b/pressure function by Spycher and Pruess (with $s_{sal} = 0$)

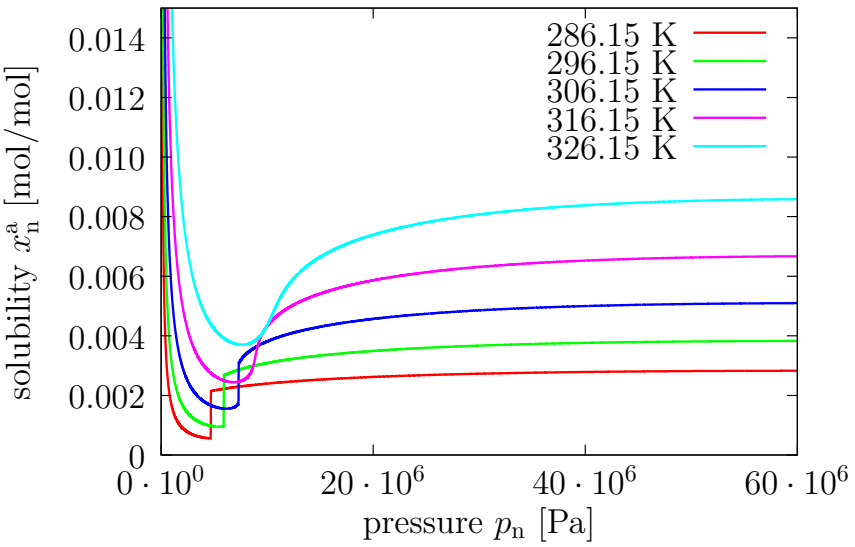


Figure 3.7: x_n^a/pressure function by Spycher and Pruess (with $s_{sal} = 0$)

pressure) to liquid. This phase transition results in a not continuously differentiable sharp break at the transition point.

Suitable geological storage sites are located very deep under the surface, where the temperature is normally higher than the critical temperature T_{crit}. Therefore, the non-differentiabilities for $T < T_{crit}$ are not relevant to the simulations in this work. Furthermore, for CCS the phase pressure is quite high if a CO_2 phase appears. That means for the case that not all the CO_2 is dissolved in the water CO_2 exists in a supercritical state.

3.3 Densities and Viscosities

For the density of the water phase, the approach of Garcia [30] is applied. The density increases slightly for a larger fraction of CO_2 in the water phase. Experimental data show, that for temperatures below 573.15 K the density is almost independent of pressure (confer [30]) and is therefore neglected in Garcia's approach. The EOS of Duan [26] is used to calculate the density of the CO_2 phase, which strongly depends on the CO_2 phase pressure:

$$\rho_{mass,w} = \rho_{mass,w}(x_w^b, T), \quad \rho_{mass,n} = \rho_{mass,n}(p_n, T).$$

Figure 3.8 shows the density of CO_2 for different temperatures.

Like for the solubility (see Section 3.2), the density below the critical temperature shows a distinct kink where the CO_2 changes from gaseous to liquid state.

Often molar density instead of mass density is used. To convert mass density to molar density the phase composition has to be taken into account:

$$\rho_\alpha = \frac{\rho_{mass,\alpha}}{x_\alpha^b M^b + x_\alpha^a M^a}. \tag{3.3}$$

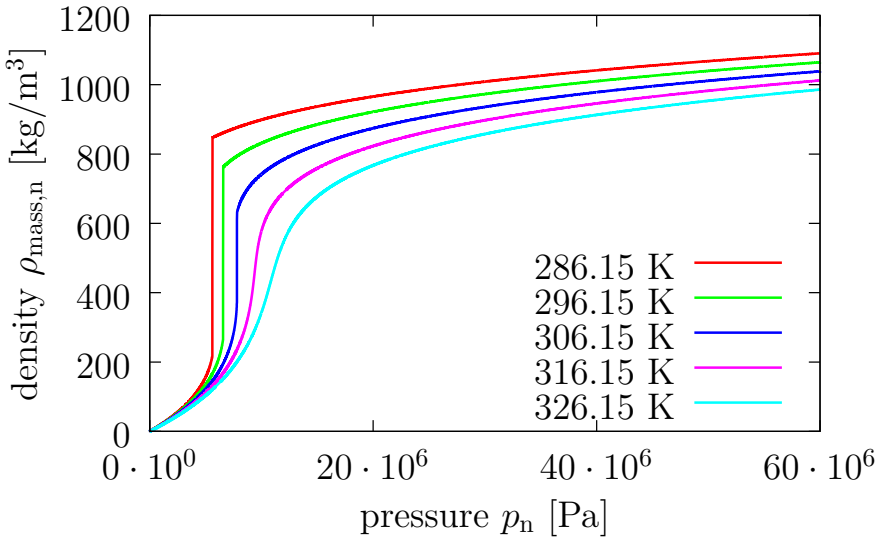


Figure 3.8: CO_2 density/pressure function by Garcia

The viscosity of the water phase is computed using a function by Atkins [4], for the CO_2 phase the approach of Fenghour and Vesovic [28] is employed. Again the CO_2 phase viscosity also depends on the CO_2 phase pressure,

$$\mu_{\mathrm{w}} = \mu_{\mathrm{w}}(T), \quad \mu_{\mathrm{n}} = \mu_{\mathrm{n}}(p_{\mathrm{n}}, T).$$

Figure 3.9 shows the viscosity of CO_2 for different temperatures.

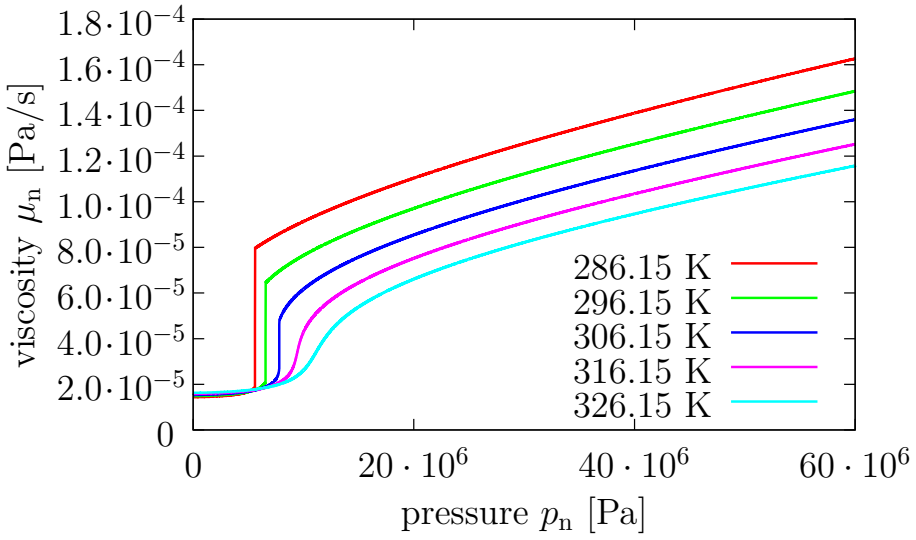


Figure 3.9: CO_2 viscosity/pressure function by Fenghour and Vesovic

3.4 Diffusion

Following [42] an approach suggested by Millington and Quirk is used to obtain the diffusion coefficient in the porous medium,

$$D^{\kappa}_{\mathrm{pm},\alpha} = \frac{(\phi S_\alpha)^{10/3}}{\phi^2} D^{\kappa}_{\alpha}. \tag{3.4}$$

where D^{κ}_{α} is the binary diffusion coefficient of component κ in phase α.

In this Chapter all missing algebraic relations for System (2.6) were introduded. The constitutive relationssships for, e. g., density and solubility are approximations of experimental data. The computation of these variables is quite expensive, therefore interpolation tables are used to speed up the computations (confer Section 5.7).

4 Choice of Primary Variables

In compositional two-phase flow the choice of primary variables is crucial for the numerical performance. A standard choice for two-phase flow simulations is one phase pressure and the saturation of the other phase. However, this approach is not valid if there are regions where only one phase exists. In this chapter, existing solutions to this problem are discussed. As selected solution, a choice of variables is presented, where capillary pressure and nonwetting phase pressure are used as primary variables.

4.1 Existing Approaches

A standard choice for the primary variables are one phase pressure and the saturation (confer [5]). In all examples in this work, the nonwetting component is injected into a fully water saturated domain and at the beginning of the process only the wetting phase is present. During injection the nonwetting phase appears and both phases are present.

In the one phase region ($S_\mathrm{n} = 0$), the system (2.6) degenerates to

$$\phi\partial_t(\rho_\mathrm{w}\, x_\mathrm{w}^\mathrm{a}) + \nabla\cdot(\rho_\mathrm{w}\, x_\mathrm{w}^\mathrm{a} u_\mathrm{w} + j_\mathrm{w}^\mathrm{a}) - \; q^\mathrm{a} \; = 0, \tag{4.1}$$

$$\phi\partial_t(\rho_\mathrm{w}\, x_\mathrm{w}^\mathrm{b}) + \nabla\cdot(\rho_\mathrm{w}\, x_\mathrm{w}^\mathrm{b}\, u_\mathrm{w} + j_\mathrm{w}^\mathrm{b}) - \; q^\mathrm{b} = 0. \tag{4.2}$$

Adding equations (4.1) and (4.2) and using (2.2) and (2.5) the system can be rewritten as a coupled groundwater-flow and transport problem

$$\phi\partial_t\,(\rho_\mathrm{w}) \qquad + \nabla\cdot(\rho_\mathrm{w} u_\mathrm{w}) \qquad\qquad = q^\mathrm{a} + q^\mathrm{b}, \tag{4.3}$$

$$\phi\partial_t\left(\rho_\mathrm{w}\, x_\mathrm{w}^\mathrm{b}\right) + \nabla\cdot\left(\rho_\mathrm{w}\, x_\mathrm{w}^\mathrm{b}\, u_\mathrm{w} + j_\mathrm{w}^\mathrm{b}\right) = q^\mathrm{b}. \tag{4.4}$$

With the disappearance of the nonwetting phase the saturation can no longer be used as primary variable and the standard choice of variables cannot be applied here. One natural set of variables for the one-phase region would be the wetting phase pressure p_w and the solubility x_w^b.

There are several approaches to solve the problem at the phase transition. A classification of several methods can also be found in [46].

4.1.1 Extended and Nonstandard Variables

The presented methods in this section use nonstandard primary variables or extend the scope of variables. Several approaches deal with the special case that Henry's Law is used to couple solubility and pressure in the context of nuclear waste management. Often the nonwetting phase is assumed to consist only of the nonwetting component.

Bourgeat et al. [16] describe several possibilities for sets of primary variables. They use Henry's Law to determine the saturations. One possibility is to use the total nonwetting concentration

$$\rho_{\mathrm{tot}}^{\mathrm{n}} = S_{\mathrm{w}}\rho_{\mathrm{mass,w}}X_{\mathrm{w}}^{\mathrm{n}} + S_{\mathrm{n}}\rho_{\mathrm{mass,n}}$$

together with the wetting phase pressure p_{w}. Another approach is to exchange $\rho_{\mathrm{tot}}^{\mathrm{n}}$ by the nonwetting concentration in the wetting phase $\rho_{\mathrm{w}}^{\mathrm{n}} = \rho_{\mathrm{mass,w}}X_{\mathrm{w}}^{\mathrm{n}}$. These primary variables have the advantage that they are continuous across material heterogeneities.

Angelini et al. [3] use the two phase pressures as primary variables and extend the definition of the nonwetting phase pressure in the case that only the wetting phase is present $S_{\mathrm{w}} = 1$.

Amaziane et al. [2] introduce two new variables, the global pressure variable and the total hydrogen mass density (following the idea of [15]). They show a fully equivalent fractional flow formulation for compositional compressible two-phase flow in porous media. With the concept of global pressure the equations are partially decoupled. The global pressure equals p_{w} when only the wetting phase is present. Both global pressure and total hydrogen mass density are valid for the saturated and unsaturated case and Amaziane et al. use them as primary variables.

Abadpour and Panfilov [1] extend the saturation to artificial negative values and values higher than 1. This causes System (2.6) to not degenerate in the one-phase region. Thus, one of the saturations can be used as a primary variable. Abadpour and Panfilov use S_{n} and p_{n} as primary variables. According to the authors, an advantage of this formulation is that existing two-phase simulators can be easily adapted to work for two-phase two-component-models.

4.1.2 Switching Primary Variables

A common method to deal with phase appearance and disappearance is primary variable switching. Here in addition to the classic variables, saturation and phase pressure, an additional variable like the total concentration or mass/mole fraction is chosen.

Depending on the phases existence at a certain location, different sets of primary variables are used. In the two-phase region, a phase pressure and saturation are used as variables, while in the one-phase region, the saturation is replaced by total concentration or mass/mole fraction. Switching the primary variables is a nondifferentiable process that can potentially lead to numerical difficulties.

Forsyth and Simpson [29] use the pressure and temperature as primary variables for the non-isothermal two-phase two-component flow. The state of the system is determined in each Newton iteration. Depending on the present phases the additional primary variable is chosen as:

- S_w, if both phases are present
- x_w^b, if only the wetting phase is present ($S_\mathrm{w} = 1$)
- x_n^b, if only the nonwetting phase is present ($S_\mathrm{w} = 0$)

Bielinski [11] uses p_n and S_w for the two phase region and p_n and X_w^b in the one phase region. **Helmig et al. [23]** describe primary variable switching for a dense non-aqueous phase liquid (DNAPL) system with three phases and three components (see Table 2 in [23] for details).

4.1.3 Using Complementarity Constraints with Semi-Smooth Newton Method

The phase transition problem can be circumvented by the use of nonlinear complementarity constraints that describe the composition of the phases and decide about the number of present phases. One additional primary variable has to be chosen for the complementarity constraint.

Gharbia and Jaffré [41] use S_w and p_w together with the molar fraction x_w^b as main variables. Additionally, nonlinear complementarity constraints describing the transition from one-phase to two-phase region are used to close the system. These complementarity constraints consist of equations and inequalities. By expressing the constraints equivalently through a complementarity function, the resulting system is free of inequalities.

The introduction of a complementarity function means that the problem is no longer C^1, and convergence is not assured for the standard Newton algorithm. So instead of the standard Newton method, Gharbia and Jaffré use a semi-smooth Newton method. Using their method for the MoMas benchmark (see Chapter 6), they observe a quadratic convergence for the semi-smooth Newton method for this problem (confer [33]).

Lauser et al. [43] present a method that is capable of handling an arbitrary number of phases and components. Their model includes temperature changes throughout the domain and therefore has an additional unknown (the temperature T).

Lauser et al. use M nonlinear complementarity conditions for the $N+1$ balance equations (for N components and the temperature) and choose $M+N+1$ primary variables. For the case of two phases and two components, which is regarded in this work, the five primary variables are: p_n, f^1, f^2, S_w and T. Instead of f^i (the fugacity of the component i) also the mole fractions x_n^a and x_n^b can be used.

Like Gharbia and Jaffré [33] the resulting complementarity functions are linearized with a semi-smooth Newton method. The authors show that with local static condensation the

size of the resulting Jacobian matrix can be reduced to three local unknowns instead of five.

4.1.4 Using Extended Variables and Complementarity Constraints with Semi-Smooth Newton

Marchand et al. [46] and [45] use a combination of complementarity constraints with a semi-smooth Newton solver together with nonstandard primary unknowns and a standard Newton solver. Here the primary variables are extended versions of molar fraction and pressure. X is the total molar fraction of the nonwetting component and P a mean pressure:

$$X = \frac{\rho_{\mathrm{n}} x_{\mathrm{n}}^{\mathrm{b}} S_{\mathrm{n}} + \rho_{\mathrm{w}} x_{\mathrm{w}}^{\mathrm{b}} S_{\mathrm{w}}}{\rho_{\mathrm{n}} S_{\mathrm{n}} + \rho_{\mathrm{w}} S_{\mathrm{w}}},$$

$$P = \varrho(S_{\mathrm{n}}) p_{\mathrm{n}} + (1 - \varrho(S_{\mathrm{n}})) p_{\mathrm{w}}$$

with a weight function ϱ, e. g., $\varrho(S_{\mathrm{n}}) = S_{\mathrm{n}}$.

The primary variables P and X always remain meaningful even when one phase is missing. Depending on the choice of ϱ, this model can handle the disappearance of the nonwetting as well as the wetting phase.

Marchand et al. call the constitutive relations that are needed to reconstruct all variables from the two primary variables a system of static equations. This system of static equations is reformulated as complementarity problem.

A Newton method is used to solve the global balance equations (2.6). The global Newton method calls local solvers for each cell that solve the system of static equations with a semi-smooth Newton method.

4.1.5 Compositional Modeling Using Flash Calculation

A very different approach to deal with the phase appearance problem is a method named flash calculation. In this case the balance equations use global compositions of each component (summed up over all phases) instead of component concentrations in each phase. The calculation of the thermodynamical equilibrium state splits global compositions of components into phase compositions. This process is called flash calculation. A drawback of this method is that the flash calculation is quite complex and iterative methods have to be used for the solution on every grid cell.

The work of **Heimsund [37]** covers the non-isothermal black oil model, which uses three phases (water, gas, oil) and three components (aquatic, light, heavy). The water phase consists only of the aquatic component. Heimsund chooses water pressure p_{w}, phase saturations of water and oil phase S_{w} and S_{oil} and temperature T as primary variables for the black oil model.

He uses a decoupled solution technique and splits the mass balance equations into a pressure equation and equations for the saturations.

The solution algorithm for each time step is then as follows:

- Solve p_{w} from the black oil pressure equation
- Solve S_{w} and S_{oil} from the black oil phase volume balance equations
- Compute Darcy velocities with determined pressures and saturations
- Determine the total number of moles (N^k) for each component k from mole balance equations
- Solve T from conservation of energy equation
- Calculate thermodynamical equilibrium state. Split the global compositions N^k into phase compositions (flash calculation)
- Repeat above steps until convergence is achieved

The phase saturations computed in the second step of the algorithm are in general different to those computed from the thermodynamical equilibrium state. Heimsund solves this by overwriting the saturation with those found from thermodynamics.

During flash calculation, the global compositions N^k are split into phase compositions. Then, phase volumes and densities are determined from pressure p_{w}, total component mass N^k and temperature T. Therefore, a fitting equation of state (EOS) together with equations for the chemical potential and fugacity are used. The flash algorithm uses Newton's method or successive substitution (or a combination of both) to solve the complex equations.

Polívka and Mikyška [57] use a two-phase multicomponent system and solve the problem of phase appearance and disappearance by a flash calculation. In contrast to the mathematical model used in this work, Polívka and Mikyška neglect diffusion and capillarity, therefore only one total pressure instead of two phase pressures are present.

For known temperature T and total molar concentrations of each component $c_1, \ldots, c_{n_c}$, they perform a constant volume phase stability test (called VT-stability) (described in [50]) to test whether a phase splitting occurs. If the VT-stability indicates that the system is in two phases, a VT-flash calculation (described in [49]) is performed, which computes saturations S_α and molar concentrations $c_{\alpha,\kappa}$ of all components in both phases. Afterwards the pressure is determined by the EOS of Peng and Robinson (confer [55]).

The authors conclude, that the advantage of their method over a standard PT-flash is that there is no need to invert the EOS in order to determine the pressure.

Polívka and Mikyška use a fully implicit combination of RT_0 mixed finite elements for the total flux and the finite volume method for the discretization of the transport equations. The resulting nonlinear system of $n_k n_c + n_e$ algebraic equations is then linearized with a Newton method. (Here n_k is the number of mesh-elements, n_e the number of edges and n_c the number of components.) The primary variables are the $n_k n_c$ overall molar concentrations of each component in each mesh-cell and n_e pressures on the edges. The overall molar concentrations and edge-pressures are well defined for both the one-phase

and two-phase region. For the special case of two components this means there are three primary variables, two overall molar concentrations c_1 and c_2 and the pressure p.

The computational algorithm is then as follows for each time step:

Repeat Newton iterations until the convergence criterion is satisfied:

- Perform the stability and flash calculations (with values from last time step or last Newton step) to obtain a number of phases and their compositions locally on all elements
- Compute the cell-averaged pressures and edge pressures with the EOS of Peng-Robinson
- Assemble and solve the final system for molar concentrations and edge pressures

In another article, Polívka and Mikyška ([56]) show, that with a semi-implicit approach they can reduce the size of the system to n_e equations, a number that does not depend on the number of components in the system.

4.2 Selected Approach: $p_\mathrm{n}/p_\mathrm{c}$ Formulation

In this work, an approach based on extended variables is developed using the pressure of the nonwetting phase and the capillary pressure as primary variables. In the absence of the nonwetting phase, p_n is defined as the corresponding pressure to the solubility x_w^b. This idea was first presented by Ippisch [39] for coupled transport processes.

This approach has the advantage that there are only two primary variables in contrast to the complementarity constraints method, where an additional variable is needed. With this constant set of variables, switching of the primary variables is avoided, which is a non-differentiable process that can lead to numerical difficulties.

In Chapters 6–9 this approach is applied to a recent benchmark study on nuclear waste management and also to the very challenging field of CCS. In contrast to nuclear waste management and the work of Bourgeat et al. (see [16]) and Angelini et al. (see [3]), it is not possible to use Henry's Law for the solubility because the approximation is not valid for CO_2. In this work, a nonlinear function (see Section 3.2) is necessary to describe the dependency between the nonwetting pressure and the molar fraction.

4.2.1 Interpretation as Algebraic Transformation

The entry pressure p_entry is the critical capillary pressure that must be applied so that the nonwetting phase appears. One has to distinguish two cases:

1. $p_\mathrm{c} \leq p_\mathrm{entry}$ where $S_\mathrm{n} = 0$ and only the wetting phase exists;
2. $p_\mathrm{c} > p_\mathrm{entry}$ where $S_\mathrm{n} > 0$ and both wetting and nonwetting phase exist.

Case 1: $p_\mathrm{c} \leq p_\mathrm{entry}$
As mentioned in the beginning of the section the natural set of variables for the one phase system (4.3) would be p_w and x_w^b. Consider the following transformation of variables

$$\begin{aligned} p_\mathrm{w} &= p_\mathrm{n} - p_\mathrm{c} \\ x_\mathrm{w}^\mathrm{b} &= \psi(p_\mathrm{n}) \end{aligned} \tag{4.5}$$

where ψ is a continuous and invertible function. The solubility relation for x_w^b (3.2) satisfies these demands (see Figure 3.6 and Spycher and Pruess [58]). The mapping between p_n and x_w^b is hence unique and p_n and p_c is a valid set of primary variables.

The relation between the capillary pressure and the saturation $p_\mathrm{c}(S_\mathrm{w})$ (see Section 3.1) is a strictly decreasing function for $S_\mathrm{w} \in [0, 1]$ and can therefore be inverted

$$S_\mathrm{w} = \eta(p_\mathrm{c}).$$

The dependent variables are then obtained through

$$\begin{aligned} S_\mathrm{w} &= \eta(p_\mathrm{c}) \quad & S_\mathrm{n} &= 1 - \eta(p_\mathrm{c}) \\ x_\mathrm{w}^\mathrm{b} &= \psi(p_\mathrm{n}) \quad & x_\mathrm{w}^\mathrm{a} &= 1 - \psi(p_\mathrm{n}) \\ x_\mathrm{n}^\mathrm{a} &= \gamma(p_\mathrm{n}) \quad & x_\mathrm{n}^\mathrm{b} &= 1 - \gamma(p_\mathrm{n}) \end{aligned}$$

where γ is the solubility curve given in (3.2). All other variables are computed as given in Chapter 2 and 3.

This choice of primary variables is not unique, another possible set would be p_w/p_c or $p_\mathrm{w}/p_\mathrm{n}$. Using p_n as a primary variable has the advantage, that the highly nonlinear density and viscosity functions are directly dependent on a primary variable. p_c is preferred over p_w as additional primary variable, because then the saturation only depends on the primary variable p_c through the nonlinear capillary pressure-saturation relationship.

Instead of the nonwetting phase pressure p_n the molar fraction $x_\mathrm{w}^\mathrm{b} = \psi(p_\mathrm{n})$ could also be used as primary variable, which is very similar to the hydrogen concentration used by Bourgeat et al. [16].

Case 2: $p_\mathrm{c} > p_\mathrm{entry}$
The common choice of primary variables in the two-phase region is one pressure and the saturation. With the $p_\mathrm{n}/p_\mathrm{c}$ formulation the saturations are obtained through the retention curve $S_\mathrm{w} = \eta(p_\mathrm{c})$, the other variables are computed accordingly.

p_w and x_w^b are continuous at the interface between the one-phase and the two-phase region. Through the transformation (4.5), p_n and p_c are continuous at the interface too. Furthermore, a formulation using $p_\mathrm{n}/p_\mathrm{c}$ as primary variables can be consistently used in the presence or absence of the nonwetting phase. One advantage of the $p_\mathrm{n}/p_\mathrm{c}$ formulation is, that the pressures, in contrast to the saturations, are continuous across material heterogeneities if both phases exist.

5 Numerical Solution

This Chapter focuses on the numerical solution of compositional two-phase flow. It highlights several important parts of the implementation. The applications of compositional two-phase flow (e. g., CO_2 sequestration) typically take place on huge domains. Additionally, large time spans have to be covered by the simulation. Therefore, the aim of the implementation is to have an efficient code, that is able to achieve an adequate precision of the solution. The numerical solution of the problem must not take longer than some days or weeks. Hence, extensive parallelization of the code is necessary.

5.1 The DUNE Framework

All simulations present in this work are based on the C++ libraries of the DUNE (Distributed and Unified Numerics Environment) simulation framework (confer [61], [8], [7]). DUNE provides data structures and algorithms for the grid-based solution of partial differential equations. While DUNE core-modules facilitate basic functionality, it can be extended to further applications by additional modules. The author of this work co-developed the high-level discretization module dune-pdelab (confer [9]) in the context of this thesis.

In PDELab the user provides a *local operator* that describes the residual assembly of the balance equations (see Equations (2.6)) on a local degree of freedom. The local operator splits the assembly into volume and face integrals. With the *grid operator*, PDELab assembles the global matrix. The algebraic system can then be solved using a suitable combination of nonlinear and linear solvers.

As a nonlinear solver, PDELab already implements an inexact Newton solver (see Section 5.5). For the linear part, there exists a great variety of preconditioners and sequential and parallel linear solvers. Here, libraries from the DUNE core module dune-istl (see [13]) are used. A very efficient and scalable preconditioner is the algebraic multigrid (AMG) method (see Section 5.6). A detailed description and introduction to PDELab can be found in [62].

Figure 5.1: Logo of DUNE simulation framework

5.2 Spatial Discretization with Finite Volumes

The domain Ω is discretized with a cell-centered finite volume method on a structured grid using a two-point flux approximation. The grid only consists of rectangles of the same size, so the simple cell-centered finite volume scheme is a suitable choice. One advantage of this method is, that it is local mass conservative.

The grid $E_h = e_1, \dots, e_n$ consists of rectangular elements e_i and the boundary of each element is $\partial e_i = \bigcup_{j \in \Sigma(i)} \gamma_{ij}$ where $\Sigma(i)$ contains the indices of the neighbor cells of e_i and γ_{ij} denotes the boundary between elements e_i and e_j. F_h contains all faces in the grid. The center of each cell e_i is x_i.

Based on the grid the non-conforming space W_h is defined as

$$W_h = \{w \in L_2(\Omega) : \, u\,|_{\Omega_e} = \text{const} \quad \forall e \in E_h\}.$$

The weak formulation of System (2.6) is: $\forall v \in W_h$

$$\begin{aligned}
&\sum_{e_i \in E_h} \int_{e_i} \Big(\phi \partial_t \left(\rho_\mathrm{w} \, x_\mathrm{w}^\kappa S_\mathrm{w} + \rho_\mathrm{n} \, x_\mathrm{n}^\kappa S_\mathrm{n} \right) \\
&\quad + \nabla \cdot \left(\rho_\mathrm{w} \, x_\mathrm{w}^\kappa u_\mathrm{w} + \rho_\mathrm{n} \, x_\mathrm{n}^\kappa u_\mathrm{n} + j_\mathrm{w}^\kappa + j_\mathrm{n}^\kappa \right) - q^\kappa \Big) \cdot v \, \mathrm{d}e \\
&= \sum_{e_i \in E_h} \Big\{ \int_{e_i} \Big(\phi \partial_t \left(\rho_\mathrm{w} \, x_\mathrm{w}^\kappa S_\mathrm{w} + \rho_\mathrm{n} \, x_\mathrm{n}^\kappa S_\mathrm{n} \right) - q^\kappa \Big) \cdot v \, \mathrm{d}e \\
&\quad + \sum_{j \in \Sigma(i)} \int_{\gamma_{ij}} \Big(\rho_\mathrm{w} \, x_\mathrm{w}^\kappa u_\mathrm{w} + \rho_\mathrm{n} \, x_\mathrm{n}^\kappa u_\mathrm{n} + j_\mathrm{w}^\kappa + j_\mathrm{n}^\kappa \Big) \cdot n_{ij} \, v \, \mathrm{d}\gamma \Big\} \\
&= \sum_{e_i \in E_h} \int_{e_i} \Big(\phi \partial_t \left(\rho_\mathrm{w} \, x_\mathrm{w}^\kappa S_\mathrm{w} + \rho_\mathrm{n} \, x_\mathrm{n}^\kappa S_\mathrm{n} \right) - q^\kappa \Big) \cdot v \, \mathrm{d}e \\
&\quad + \sum_{\gamma_{ij} \in F_h} \int_{\gamma_{ij}} \Big(\rho_\mathrm{w} \, x_\mathrm{w}^\kappa u_\mathrm{w} + \rho_\mathrm{n} \, x_\mathrm{n}^\kappa u_\mathrm{n} + j_\mathrm{w}^\kappa + j_\mathrm{n}^\kappa \Big) \cdot n_{ij} \, [v] \, \mathrm{d}\gamma = 0.
\end{aligned}$$

Here n_{ij} denotes the unit outer normal to face γ_{ij} pointing from e_i to e_j and $[v] = v(x_i) - v(x_j)$.

The integrals over the faces are approximated by the midpoint rule:

$$\begin{aligned}
r_h^\kappa(p_h, v) = &\sum_{e_i \in E_h} \Big(\phi(x_i) \partial_t \Big(\sum_{\alpha \in \{\mathrm{w,n}\}} \rho_\alpha(x_i) x_\alpha^\kappa(x_i) S_\alpha(x_i) \Big) - q^\kappa(x_i) \Big) v(x_i) |e_i| \\
&+ \sum_{\gamma_{ij} \in F_h} \Big(\sum_{\alpha \in \{\mathrm{w,n}\}} \rho_{\alpha,ij} \, x_{\alpha,ij}^\kappa u_{\alpha,ij} + \frac{j_\alpha^\kappa(x_i) - j_\alpha^\kappa(x_j)}{\|x_i - x_j\|} \Big) \cdot n_f \, [v] \, \mathrm{d}\gamma = 0.
\end{aligned}$$

$\rho_{\alpha,ij}$, $x_{\alpha,ij}^\kappa$ and $u_{\alpha,ij}$ are computed depending on the upwind element (see 5.3).

The cell-centered finite volume method for System (2.6) for each component κ (p_h is one of the primary variables p_c and p_n depending on κ) then reads:

$$\text{Find } p_h \in W_h : r_h^\kappa(p_h, v) = 0 \quad \forall v \in W_h \tag{5.1}$$

5.3 Upwinding

In general, the finite volume approximation will not produce stable results. An upwinding scheme can be utilized to stabilize the problem. In [40] Ippisch presents a special upwinding scheme for coupled transport in porous media. This scheme is used for the calculation of the phase fluxes at the interface between two elements. It can handle material discontinuities, which result in different capillary-pressure saturation curves and relative permeability functions in neighboring elements. This upwinding scheme is adapted here for two-phase two-component flow.

The direction of the flux of phase α at the interface between two elements i and j can be obtained from the sum of the pressure gradient and the force of gravitation

$$w_{\alpha,ij} = \left(\frac{p_\alpha(x_i) - p_\alpha(x_j)}{|x_i - x_j|} + \rho_{\text{mass},\alpha,ij} \cdot g \right) \cdot n_{ij},$$

where $\rho_{\text{mass},\alpha,ij}$ is computed as the arithmetic average of cells e_i and e_j. Depending on the sign of $w_{\alpha,ij}$ the upwind element is determined as:

$$\mathsf{upwind}_\alpha = \begin{cases} i & w_{\alpha,ij} \geq 0, \\ j & \text{else.} \end{cases}.$$

The capillary pressure of the upwind element is used to calculate the relative permeability in each element. The obtained relative permeabilities are multiplied by the absolute permeabilities and the viscosities in each element. A harmonic average of the values is used to calculate the flux at the interface:

$$\begin{aligned} K_{\alpha,i} &= K_i \frac{k_{r\alpha,i}(p_{c,\mathsf{upwind}_\alpha})}{\mu_{\alpha,i}} \\ K_{\alpha,j} &= K_j \frac{k_{r\alpha,j}(p_{c,\mathsf{upwind}_\alpha})}{\mu_{\alpha,j}} \\ u_{\alpha,ij} &= \frac{K_{\alpha,i} K_{\alpha,j}}{K_{\alpha,i} + K_{\alpha,j}} w_{\alpha,ij} \end{aligned}$$

For homogeneous porous media, this upwinding scheme corresponds to an upwinding of saturation.

For the calculation of the convective component transport a full upwinding of the molar fractions and the molar densities based on the upwind direction is used with

$$x^\kappa_{\alpha,ij} = x^\kappa_{\alpha,\mathsf{upwind}_\alpha}, \quad \rho_{\alpha,ij} = \rho_{\alpha,\mathsf{upwind}_\alpha}.$$

5.4 Time Discretization with Implicit Euler

In time Equation (5.1) is discretized with the implicit Euler scheme. The time interval $[0, T]$ is split into parts with time points: $0 = t_0, \ldots, t_N = T$. Time step n is then the

interval of length $\Delta t_n = t_{n+1} - t_n$ starting at t_n. The evaluation of a variable at time t_n is abbreviated with a superscript n, e. g., $p_\mathrm{w}(t_n) = p_\mathrm{w}^n$.

With the implicit Euler method, Equation 5.1 reads in time step n

$$\sum_{e_i \in E_h} \Bigg\{ \int_{e_i} \Big(\frac{\phi}{\Delta t_n} \Big\{ \nu_\mathrm{w}^{\kappa,n} S_\mathrm{w}^n + \nu_\mathrm{n}^{\kappa,n} S_\mathrm{n}^n - \nu_\mathrm{w}^{\kappa,n-1} S_\mathrm{w}^{n-1} - \nu_\mathrm{n}^{\kappa,n-1} S_\mathrm{n}^{n-1} \Big\} - q^{\kappa,n} \Big) \mathrm{d}e + \frac{1}{\|e_i\|} \sum_{j \in \Sigma(i)} \int_{\gamma_{ij}} \Big(\{ \nu_\mathrm{w}^{\kappa,n} u_\mathrm{w}^n + \nu_\mathrm{n}^{\kappa,n} u_\mathrm{n}^n \} + \{ j_\mathrm{w}^{\kappa,n} + j_\mathrm{n}^{\kappa,n} \} \Big) \cdot n_{ij} \, \mathrm{d}\gamma \Bigg\} = 0,$$

with $\nu_\alpha^\kappa = \rho_\alpha \, x_\alpha^\kappa$.

5.5 Inexact Newton Method with Line Search

5.5.1 Standard Method

The system of equations arising from numerical discretization is highly nonlinear. Newton's method is an efficient way to solve the difficult, nonlinear equations. It is an iterative scheme that uses a linear approximation to find roots of the nonlinear system. A description of the inexact Newtons Method can be found in [5].

After discretization, there is a set of nonlinear algebraic equations with solution vector u

$$R(u) = 0,$$

to be solved per time step. The Jacobian matrix is defined as the local derivative of R in u:

$$A_{i,j} = (\nabla R(u))_{i,j} = \frac{\partial (R)_i}{\partial u_j}(u)$$

The exact computation of the Jacobian for compositional two-phase flow is very difficult. In this work an approximation is used and the Jacobian is derived by one-sided numerical differentiation through:

$$\frac{\partial (R)_i}{\partial u_j}(u) = \frac{R_i(u + \Delta u_j e_j) - R_i(u)}{\Delta u_j} + \mathcal{O}(\Delta u_j)$$

where $\Delta u_j = \epsilon(1 + \|u_j\|)$. In this work $\epsilon = 1 \cdot 10^{-7}$ is used for all computations.

For strongly nonlinear functions line search increases the convergence sphere by adding only a fraction of the correction until convergence is achieved. A theoretical motivation of the line search strategy can be found in [17]. Here the line search strategy of Hackbusch and Reusken is employed (confer [34]).

The algorithm below solves the equation $R(u) = 0$ with initial guess u^0 and relative accuracy ϵ_nl or absolute error $\epsilon_\mathrm{nl}^\mathrm{abs}$:

With a given u^0, compute residual $r^0 = R(u^0)$. Do in each Newton iteration k:

- Assemble Jacobian system $A = (\nabla R(u^k))$
- Solve $Az^k = r^k$ with a linear solver with accuracy ϵ_{lin}: $\|Az^k - r^k\|_2 \leq \epsilon_{\text{lin}} \|r^k\|_2$
- Throw exception if linear solver does not converge
- Use z^k as correction and update for $l = 0$: $u^{k+1,0} = u^k - z^k$
- $\sigma = 1$. Do in each line search iteration l:
  - Accept $u^{k+1,l}$ as u^{k+1} if $\|R(u^{k+1,l})\|_2 > (1 - \frac{\sigma}{4})) \cdot \|R(u^k)\|_2$
  - Scale with dampening factor λ: $\sigma = \lambda\sigma$
  - Update $u^{k+1,l+1} = u^k - \sigma z^k$
  - Throw exception if $l > \max_{\text{linesearch}}$
- Compute residual $r^{k+1} = R(u^{k+1})$
- Accept u^{k+1} as solution if $\|r^{k+1}\|_2 \leq \epsilon_{\text{nl}} \|r^0\|_2$ or $\|r^{k+1}\|_2 \leq \epsilon_{\text{nl}}^{\text{abs}}$
- Set $k = k + 1$
- Throw exception if $k > \max_{\text{newtoniterations}}$

A typical dampening factor for line search is $\lambda = 0.5$. In the simulations in this work either $\lambda = 0.5$ or $\lambda = 0.7$ are chosen. A maximum number of line searches $\max_{\text{linesearch}} = 10$ and a maximum number of newton iterations $\max_{\text{newtoniterations}} = 10$ have been proven appropriate. The choice of the stopping criteria ϵ_{nl} and $\epsilon_{\text{nl}}^{\text{abs}}$ is crucial for the performance and accuracy of the simulation.

The resulting linear equation system is solved with a suitable solver preconditioned by an algebraic multigrid method (see Section 5.6.1) or a direct solver like SuperLU for problems with a small number of unknowns.

5.5.2 Modified Newton Method for Pressure/Pressure Formulation

Using the pressure/pressure formulation from Section 4.2, the typical Newton scheme does not always converge as expected. During the appearance of a phase in a cell, it only converges if very small time steps are chosen.

For test case 1 from Chapter 6, Figure 5.2 shows the pressure evolution in a cell during phase appearance for the $p_{\text{n}}/p_{\text{c}}$ formulation: Before the nonwetting phase appears (until approximately 13000 years), the derivative of p_{c} is very high. After the appearance of the second phase, p_{c} is almost constant. This kink in the capillary pressure is a problem for the standard Newton solver: Close to the kink, the solver uses the high derivative of the capillary pressure and overestimates the update for p_{c} dramatically. Thus, convergence is only achieved for relatively small time steps.

Part of this work is the development of a modified Newton solver that is able to handle larger time steps. The modified Newton solver identifies the cells where a second phase appears. In these cells a different, more suitable starting point for the primary variables p_{c} and p_{n} is chosen. As new p_{c} a value slightly above the entry pressure is used, where the derivative has a reasonable value. The nonwetting phase pressure p_{n} is adjusted to the new capillary pressure and is set to $p_{\text{c}} + p_{\text{w}}$.

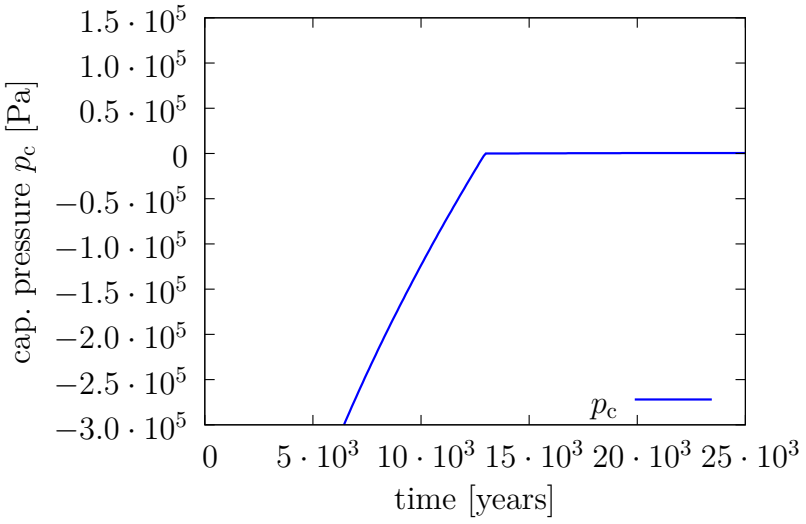


Figure 5.2: Capillary pressure evolution during phase appearance: the capillary pressure rises approximately linear until the entry pressure 0 is reached and the second phase appears.

Newton step k with phase appearance:
if $\left(p_c^k < p_{\text{entry}} \left(S_n^k = 0 \right) \quad \& \quad p_c^{k+1} > p_{\text{entry}} \left(S_n^{k+1} > 0 \right) \right)$:
Assemble new Jacobian system with

- $p_c^k = p_{\text{entry}} + \epsilon$
- $p_n^k = p_{\text{entry}} + \epsilon + p_w^k$

This check for an appearance of a new phase is carried out in every Newton iteration k just before the line search takes place. If no modifying of the primary variables is necessary the Newton algorithm just continues as discussed in the standard algorithm. Otherwise the solver restarts step k by assembling a new Jacobian system with the modified $u^k = \left(p_c^k, p_n^k \right)$.

ϵ has to be chosen depending on the parameters of the van Genuchten or Brooks-Corey function. For the van Genuchten curve in Section 6.3, $p_{\text{entry}} = 0$ and $\epsilon = 10$ is a reasonable choice.

5.6 Parallelization

The huge size and the long time spans needed for the applications require the use of parallelization. Otherwise, the numerical simulations of complex test problems is not feasible in a reasonable time. Especially the use of an efficient and fast linear solver is crucial, as the linear solver claims a lot of the overall simulation time. DUNE uses a domain decomposition approach with a disjoint partitioning, where each process is responsible for one partition of the grid (confer [13]).

5.6.1 Overlapping Solvers

In this work, a BiCGStab solver preconditioned with an AMG method (confer [12]) is used. This solver is already implemented in the modules dune-istl and dune-pdelab. Each process is liable for one partition of the grid. A layer of overlap cells adjacent to the boundary of the subdomain ensures that each process has the relevant information to assemble its part of matrices independently.

The AMG preconditioned solver was developed by Blatt [12]. The scalability of this solver has been shown for computations with up to 917504 processes on the supercomputer JUQUEEN in Jülich by Ippisch [40].

The solver needs an appropriate grid that supports the domain decomposition approach. For this work structured rectangular grids are used. YaspGrid is an efficient grid manager for structured grids in the module dune-grid that can be used with as well as without overlap. In all applications YaspGrid with an overlap of one cell is chosen.

5.6.2 Nonoverlapping Solvers

There are two grids available in DUNE that support unstructured or tetrahedral meshes: ALUGrid (see [25]) and UG (see [6]). Both of them are nonoverlapping grids, which means that the additional overlap cells used for AMG in [12] are not available.

In the context of this work, a nonoverlapping AMG method was introduced in dune-pdelab and dune-istl. The existing AMG clearly separates parallel data composition together with the communication interfaces from the data structures. Thus, the implementation of a nonoverlapping version is straightforward. It is only necessary to provide nonoverlapping versions of scalar product, preconditioner and linear operator.

The missing overlap implies some extra work for the cells at the border of each process' subdomain. For cells on the border of a process, in contrast to the overlapping version, several additional processes have to contribute to the matrix line assembly of those cell. The nonoverlapping linear operator solves this problem by additional communication.

The code is verified for a linear test problem obtained from the Poisson equation. The results for different grid size and number of processes are summarized in Table 5.1. For YaspGrid, overlapping and nonoverlapping domain decomposition can be used (denoted as OY and NY in the table), for ALUGrid only the nonoverlapping solver is applicable

Table 5.1 demonstrates the efficiency of AMG in contrast to other preconditioners. It also shows that the nonoverlapping and overlapping version achieve similar results. Compared to the YaspGrid versions, the total simulation time for ALUGrid is only slightly larger. This is due to the overhead of ALUGrid as an unstructured grid compared to the structured YaspGrid.

Nonoverlapping AMG is part of the DUNE library since the DUNE 2.0 release.

	OY	NY	ALU
BiCGStab SSOR			
iterations	27.5	29	49.5
convergence rate	0.415	0.448	0.625
TIT [s]	0.45	0.39	0.40
total time [s]	12.5	11.2	19.6
BiCGStab AMG			
iterations	8.5	8.0	8.0
convergence rate	0.063	0.046	0.055
TIT [s]	0.56	0.53	0.60
total time [s]	4.74	4.28	4.78

	OY	NY	ALU
BiCGStab SSOR			
iterations	114	101.5	84
convergence rate	0.851	0.796	0.752
TIT [s]	1.0	1.8	1.4
total time [s]	114.5	184.2	117.0
BiCGStab AMG			
iterations	11.5	10.5	11.5
convergence rate	0.132	0.100	0.120
TIT [s]	3.2	3.9	2.3
total time [s]	36.9	40.9	26.0

Table 5.1: Comparison of different preconditioners for BiCGStab applied to a Poisson model problem using overlapping and nonoverlapping partitions on a grid with $60 \times 60 \times 60$ elements and 6 processes (left) and $120 \times 120 \times 120$ elements and 16 processes (right)

5.7 Interpolation Tables

In every time step, the program has to evaluate different parameters like solubility or permeability in every grid cell. The evaluation of the nonlinear EOS and the many power functions is quite time consuming. A good solution to this problem is the usage of interpolation tables, where precomputed values in in certain intervals are interpolated. This uses more memory but saves computation time. Linear splines with equidistant intervals as recommended and implemented in [40] are used in this work.

Additional to interpolation tables for saturation and relative permeabilities (confer Section 3.1) also tables for density and viscosity of CO_2 (confer Section 3.3) and especially the solubilities (confer Section 3.2) are used. A description of the interpolation class can be found in Appendix B.

With a simple test the time savings achieved through interpolation tables is measured. A test case with a setup similar to test case 2 (see Section 7.1) is used, one with interpolation tables, the other without. For 113 time steps, the simulation without interpolation tables takes 457 s, while with interpolation tables, the simulation takes only 249 s (180% faster). For every further time step this difference increases, so that the time saving for simulations with many time steps (for example test case 4, see Chapter 9) is enormous.

6 Test Case 1 (MoMas Benchmark): Gas Injection in a Fully Water Saturated Domain (1D)

The first test case is part of a recent benchmark study in the context of nuclear waste management focusing on appearance and disappearance of phases. As there are no analytical solutions for two-phase two-component flow systems, results of other research groups are used as possibility to validate the results. The test case is an example from the MoMas benchmark on multiphase flow in porous media (see [64]).

6.1 Parameters and Setup

The descriptions from the MoMas benchmark are converted to match the variables used in this work. For the MoMas benchmark the considered nonwetting component is hydrogen and the wetting component is water. The solubility of water in the nonwetting phase is neglected: $x_{\mathrm{n}}^{\mathrm{a}} = 0$. Hydrogen is injected into the left part of a rectangular domain Γ_{in} (see Figure 6.1) with a flux of $w_{\mathrm{in}}^{\mathrm{b}}$ for $5 \cdot 10^5$ years. Gravitation is neglected, which leads to a quasi-1D problem.

The domain is initially fully saturated by the water phase, consisting only of pure water with initial conditions $p_\alpha = p_\alpha^{\mathrm{i}}$ (see Table 6.1). The boundary conditions are no-flux conditions at the top and bottom (see Figure 6.1). The Dirichlet boundary conditions for the outflow boundary Γ_{out} are the same as the initial conditions: $p_\alpha|_{\Gamma_{\mathrm{out}}} = p_\alpha^{\mathrm{i}}$.

The relationship between p_{n} and $X_{\mathrm{w}}^{\mathrm{b}}$ (where X_α^κ is the mass fraction in contrast to the molar fraction x_α^κ) is given through Henry's law (confer (3.1)):

$$X_{\mathrm{w}}^{\mathrm{b}} = \frac{H}{\rho_{\mathrm{w}}} p_{\mathrm{n}}^{\mathrm{b}}.$$

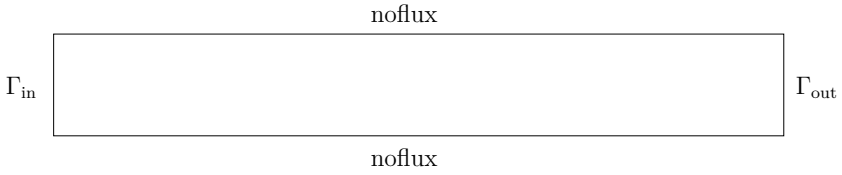


Figure 6.1: Domain setup for test case 1

parameter	symbol	value	unit
porosity	ϕ	0.15	
van Genuchten parameter	$S_{\mathrm{w,res}}$	0.4	
van Genuchten parameter	$S_{\mathrm{n,res}}$	0	
van Genuchten parameter	n	1.49	
van Genuchten parameter	α_{VG}	$5 \cdot 10^{-7}$	Pa^{-1}
absolute permeability	K	$5 \cdot 10^{-20}$	m^2
molar mass	M^{a}	$1 \cdot 10^{-2}$	$\mathrm{kg\,mol}^{-1}$
molar mass	M^{b}	$2 \cdot 10^{-3}$	$\mathrm{kg\,mol}^{-1}$
initial pressure	$p_{\mathrm{w}}^{\mathrm{i}}$	$1 \cdot 10^{6}$	Pa
initial pressure	$p_{\mathrm{n}}^{\mathrm{i}}$	0	Pa
source/sink term	q^{κ}	0	$\mathrm{kg\,m}^{-2}\,\mathrm{s}^{-1}$
molecular diffusion coefficient	$D_{\mathrm{w}}^{\mathrm{b}}$	$3 \cdot 10^{-9}$	$\mathrm{m}^2\,\mathrm{s}^{-1}$
molecular diffusion coefficient	$D_{\mathrm{n}}^{\mathrm{a}}$	0	$\mathrm{m}^2\,\mathrm{s}^{-1}$
standard water mass density	ρ^{a}	$1 \cdot 10^{3}$	$\mathrm{kg\,m}^{-3}$
viscosity	μ_{w}	$1 \cdot 10^{-3}$	$\mathrm{Pa\,s}^{-1}$
viscosity	μ_{n}	$9 \cdot 10^{-6}$	$\mathrm{Pa\,s}^{-1}$
Henry's law parameter	H	$7.65 \cdot 10^{-6}$	$\mathrm{mol\,m}^{-3}\,\mathrm{Pa}^{-1}$
flux of hydrogen at Γ_{in}	$w_{\mathrm{in}}^{\mathrm{b}}$	$1.77 \cdot 10^{-13}$	$\mathrm{kg\,m}^{-2}\,\mathrm{s}^{-1}$
flux of water at Γ_{in}	$w_{\mathrm{in}}^{\mathrm{a}}$	0	$\mathrm{kg\,m}^{-2}\,\mathrm{s}^{-1}$

Table 6.1: Parameters for test case 1

Here, the partial pressure of hydrogen in the nonwetting phase is $p_{\mathrm{n}}^{\mathrm{b}} = p_{\mathrm{n}}$ because there is no water in the nonwetting phase for this example.

The nonwetting phase density is determined by the ideal gas law, wetting phase density is obtained through Henry's Law as

$$\rho_{\mathrm{mass,n}} = p_{\mathrm{n}} M^{\mathrm{b}} (RT)^{-1}, \quad \rho_{\mathrm{mass,w}} = \rho^{\mathrm{a}} + H(T) M^{\mathrm{n}} p_{\mathrm{n}},$$

where $\rho_{\mathrm{w}}^{\mathrm{std}}$ is the standard water mass density.

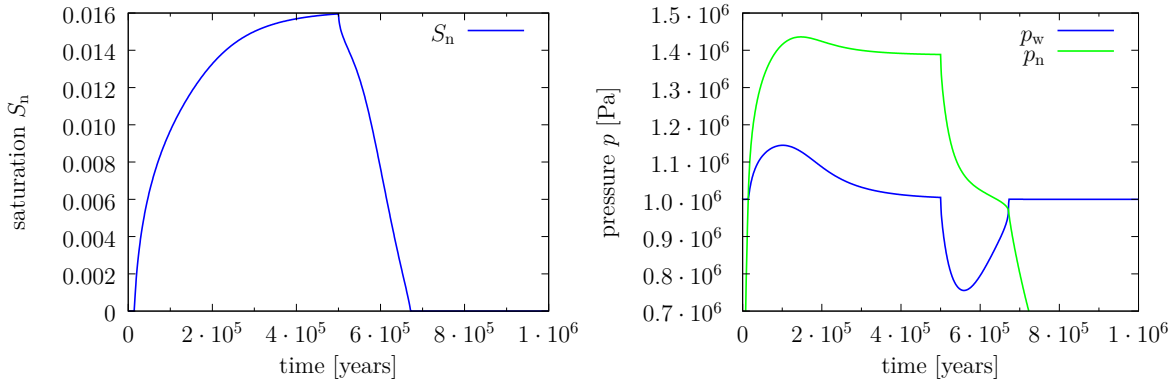
Here, the diffusion coefficient is defined as

$$D_{\mathrm{pm},\alpha}^{\kappa} = \phi S_{\alpha} \left(\frac{X_{\alpha}^{\mathrm{a}}}{M^{\mathrm{a}}} + \frac{X_{\alpha}^{\mathrm{b}}}{M^{\mathrm{b}}} \right) D_{\alpha}^{\kappa}.$$

A van Genuchten model (confer Section 3.1) with the parameters n, α_{VG} and $S_{\alpha,\mathrm{res}}$ as given in Table 6.1 is used for the saturation/capillary pressure and relative permeability relationships. All other parameters used in the simulation are also noted in Table 6.1.

6.2 Numerical Results

The choice of the right time step size is very important for the performance of the code. If the time step is too large, the Newton solver might not converge. If it is too small,



(a) Nonwetting phase saturation

(b) Phase pressures

Figure 6.2: Saturation and pressure evolution at influx boundary (Γ_{in}) of test case 1

the simulation takes too much time. Unfortunately, the right time step size can often not be determined beforehand. Additionally, it may change over time (e. g., the time step must not be too large, when a new cell is filled with the nonwetting phase). Therefore, an adaptive time stepping is commendable. For that an initial time step size $\mathrm{d}t_{\text{initial}}$ is chosen. Then, if the Newton solver converges the size for the next time step is increased by a certain factor $\mathrm{d}t_{\text{scale}}^{\text{up}}$ (for test case 1: $\mathrm{d}t_{\text{scale}}^{\text{up}} = 1.2$). Otherwise, i. e., if the Newton solver does not converge, the time step computation is unsuccessful. In this case, the computation is repeated with a time step that is decreased by the factor $\mathrm{d}t_{\text{scale}}^{\text{down}}$ (for test case 1: $\mathrm{d}t_{\text{scale}}^{\text{down}} = 0.5$). $\mathrm{d}t_{\text{scale}}^{\text{up}}$ and $\mathrm{d}t_{\text{scale}}^{\text{down}}$ have to be chosen carefully depending on the setup of the simulation.

Figure 6.2 shows the evolution of the nonwetting phase saturation and phase pressures at Γ_{in} over time. S_n is zero at the beginning. All injected hydrogen dissolves into the wetting phase and no nonwetting phase is present. At $t \approx 13000$ years, a nonwetting phase starts to appear at the injection point Γ_{in}. The maximum hydrogen saturation is reached at about $t \approx 5 \cdot 10^5$ years. After that the injection is stopped and the hydrogen saturation decreases fast. A structured grid of 400 cells is used for the computations.

The output of the simulation corresponds well to the results of other groups. The results of the benchmark and a comparison between all participating groups can be found in [14]. The test case described here corresponds to test case 1 in the article. A detailed comparison with the results of Marchand and Knabner is available in [44]. The good correspondence to the results of various research groups indicates that the model and implementation in this work provides accurate results.

6.3 Preventing Time Step Breakdown during Phase Appearance

In the course of this work it became clear, that the standard Newton solver is not sufficient for the efficient simulation of all test cases. For some applications (i. e., test case 2 in

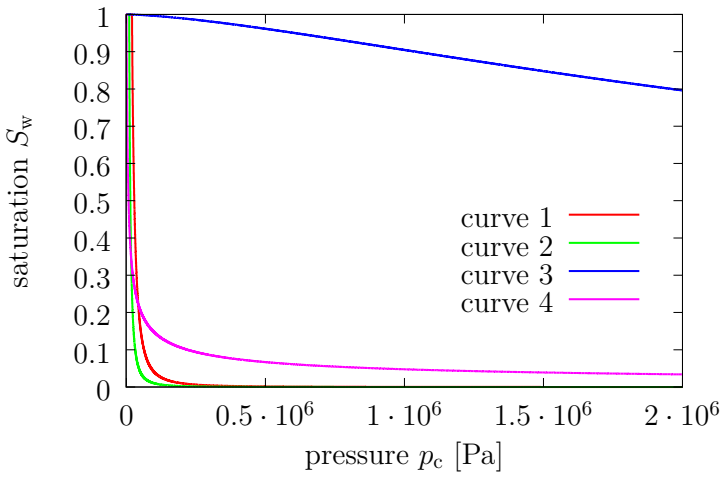


Figure 6.3: Different water retention curves:
curve 1: from Svalbard CO_2 benchmark (Brooks-Corey) (see [24] and [52])
curve 2: from Stuttgart CO_2 benchmark (Brooks-Corey) (see [22])
curve 3: from MoMas benchmark 1a (van Genuchten) (see [64])
curve 4: test curve for CO_2 (van Genuchten)

Chapter 7) the time step size occasionally dropped to a very small value due to non-convergence in the Newton solver.

For test case 1 this behavior did not appear. One big difference between test case 1 and 2 is the underlying physical process. Test case 1 models hydrogen injection into a very dense material, whereas the application of test case 2 is injection of CO_2 into an aquifer consisting of sandstone. The water retention curves (see Section 3.1) are very different due to different soil structure. Figure 6.3 shows typical water retention curves for recent CO_2 benchmarks and for the MoMas benchmark.

To investigate the problem, the setup of the simple 1D MoMas test was chosen to directly compare the influence of different water retention curves. A van Genuchten curve (curve 4 in Figure 6.3) resembling the CO_2 benchmark curves is chosen for the CO_2 setup. The only difference between the CO_2 and MoMas curve is the van Genuchten parameter α_{VG}, which is $\alpha_1 = 5 \cdot 10^{-7}$ for MoMas and $\alpha_2 = 5 \cdot 10^{-4}$ for CO_2.

Large values of α_{VG} lead to a higher saturations in fewer cells compared to the smaller α_{VG}. Figure 6.4 shows the nonwetting phase saturation distribution in the domain at 100 000 years.

Numerical simulations with the exact same parameters (see Table 6.1) and a maximum time step size of 5 000 years are performed, where the only difference is the parameter α_{VG}. The results are summarized in Table 6.2.

For the CO_2 case, the average time step size (ADT) is only a third of the ADT for the MoMas case and the simulation takes almost 20 times longer because the number of time

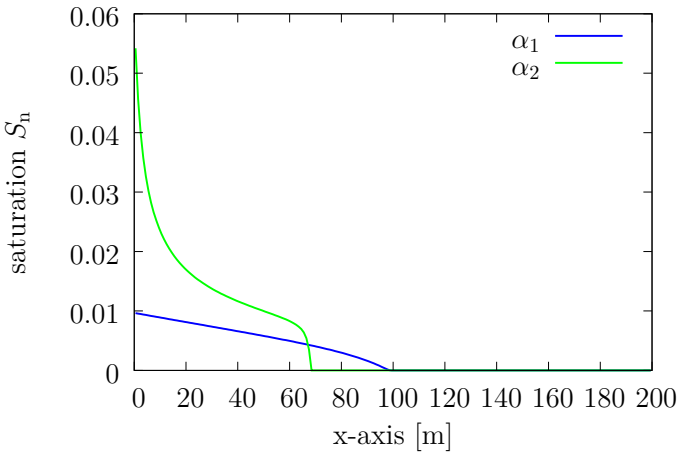


Figure 6.4: Nonwetting phase saturation distribution at 100 000 years

	α_1 (MoMas)	α_2 (CO_2)
ADT	4350 y	1420 y
TS	228	708
total simulation time	1.7 min	33.1 min

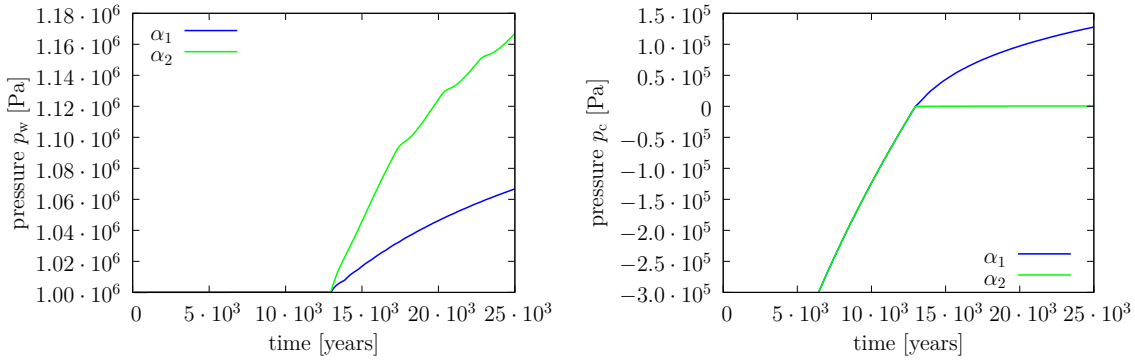
Table 6.2: Comparison between simulations for different α_{VG}

steps (TS) is larger. Unsuccessful time steps consume a lot of computation time. This is due to the fact that the Newton solver uses the maximum number of allowed Newton iterations (in this case 10) for the computation of an unsuccessful time step, while for a successful time step usually only few iterations are needed.

Using a smaller maximum number of allowed Newton iterations reduces the computation time spent on unsuccessful time steps. On the other hand it could cause time steps to not converge that would otherwise be successful. Then the computation has to be repeated with a smaller time step. Therefore, the maximum number of Newton iterations and the time stepping control have to be chosen carefully for each problem.

Figure 6.5 shows wetting phase pressure and capillary pressure over time at the leftmost cell directly at the influx boundary. At the beginning (until $\approx 13\,000$ y) only the water phase exists in the cell. After that, the nonwetting phase appears and the behavior of the pressure curves changes significantly. The capillary phase pressure for α_1 rises steadily, while for α_2 there is a distinct kink where the nonwetting phase appears. After the kink the capillary pressure is almost constant for the CO_2 case in contrast to the MoMas case.

This difference in the pressure evolution for different choices of α_{VG} causes convergence problems. For large α_{VG} the derivative of p_c is very high before the nonwetting phase appears. During the appearance, the Newton solver overestimates the update for p_c dramatically and does not converge. The standard Newton solver only converges for very



(a) wetting phase pressure

(b) capillary pressure

Figure 6.5: Pressure evolution for different α in the leftmost cell directly at the influx boundary Γ_{in}

	Standard Newton Solver	Modified Newton Solver
ADT	1420 y	2956 y
TS	709	341
total simulation time	33.1 min	10.7 min

Table 6.3: Comparison between standard and modified Newton solver for simulation with α_2 (CO_2 case)

small time steps. Instead, a modified Newton solver (confer Section 5.5) was developed, which uses a more suitable starting point for p_c in these cases.

The same simulation for α_2 with the modified Newton solver is three times faster than the standard Newton solver, see Table 6.3. The reason for this large speedup is the increased time step size which results in only half as many time steps. There are fewer unsuccessful time steps, thus saving a lot of computing time.

6.4 Grid Convergence

A grid convergence study can verify the experimental order of convergence in the implementation. As the grid is refined and the time step is reduced the spatial and temporal discretization errors should asymptotically approach zero. The order of this error reduction depends on the discretization methods and can be determined experimentally.

For a better understanding of the challenges in compositional two-phase flow, a second benchmark is proposed in this work. This benchmark concentrates on phase appearance and disappearance and includes two different test cases for different van Genuchten α_{VG} (corresponding to Section 6.3).

The simulation setup is almost identical to the MoMas benchmark in Section 6.1. Although, the system is written in terms of molar mass balances. A minor difference is the definition of the density of the wetting phase. The complete setup is presented in Appendix A.

There are two test cases that are identical apart from different values of α_{VG} for the van Genuchten curves:

- $\alpha_1 = 5 \cdot 10^{-7}$ easy case (original value)
- $\alpha_2 = 5 \cdot 10^{-4}$ difficult case (new value)

Note the slightly different notation in Appendix A. Mass densities are denoted by ρ_α and ν_α is the molar density. The molar fraction of a component in a phase x_α^κ can be computed via the molar density concentration $\nu_\alpha^\kappa = x_\alpha^\kappa \nu_\alpha$.

Parallelity would introduce an additional error through runs with different numbers of processes. Therefore a setup is chosen that allows all simulations to run sequentially. The simple 1D setup is a very good example for this application as the amount of grid cells is small.

The Newton solver uses a relative error of $\epsilon_{\mathrm{nl}} = 1 \cdot 10^{-6}$, the absolute error is $\epsilon_{\mathrm{nl}}^{\mathrm{abs}} = 0$. To minimize the influence of the time discretization error the time step size must not be too large. For all simulations an initial and maximum time step size of $\mathrm{d}t = 4$ years is used.

For the coarsest level (level 0), 100 cells are used. For each level, the amount of grid cells in x direction is doubled, so there are $2^i \cdot 100$ cells for level i. The solution on level 7 is used as reference solution, because no analytical solution is available. The resulting experimental order of convergence (EOC) is then computed through

$$\mathrm{EOC}_{i+1} = \frac{1}{\log(2)} |\log\left(\frac{e_i}{e_{i+1}}\right)|,$$

where e_i is the $L2$ error between the solution on level i and the reference solution. The relative $L2$ error e_i^{rel} is defined as e_i divided by the $L2$ norm of the reference solution.

Tables 6.4 and 6.5 show EOC and relative error to the reference solution for saturation and wetting phase pressure at $t = 1 \cdot 10^5$ years. In order to reduce the time discretization error a maximum time step of only $\mathrm{d}t = 0.5$ years is used.

The convergence study shows that the numerical solution with the finite volume discretization converges with an optimal EOC of one for α_1. For α_2, the EOC is slightly smaller.

N	S_n		p_w	
	e^rel	EOC	e^rel	EOC
$2^0 \cdot 100$	$2.0 \cdot 10^{-2}$		$5.9 \cdot 10^{-4}$	
$2^1 \cdot 100$	$1.0 \cdot 10^{-2}$	0.97	$2.9 \cdot 10^{-4}$	1.02
$2^2 \cdot 100$	$5.1 \cdot 10^{-3}$	0.98	$1.5 \cdot 10^{-4}$	0.96
$2^3 \cdot 100$	$2.5 \cdot 10^{-3}$	1.01	$7.3 \cdot 10^{-5}$	1.02
$2^4 \cdot 100$	$1.2 \cdot 10^{-3}$	1.04	$3.6 \cdot 10^{-5}$	1.01
$2^5 \cdot 100$	$5.7 \cdot 10^{-4}$	1.10	$1.8 \cdot 10^{-5}$	1.00
$2^6 \cdot 100$	$2.5 \cdot 10^{-5}$	1.21	$9.5 \cdot 10^{-6}$	0.94

Table 6.4: Grid convergence study for α_1 at $t = 1 \cdot 10^5$ years

#elements	S_n		p_w	
N	e^rel	EOC	e^rel	EOC
$2^0 \cdot 100$	0.104		$2.1 \cdot 10^{-3}$	
$2^1 \cdot 100$	0.060	0.79	$1.0 \cdot 10^{-3}$	0.99
$2^2 \cdot 100$	0.036	0.76	$6.7 \cdot 10^{-4}$	0.64
$2^3 \cdot 100$	0.022	0.68	$3.6 \cdot 10^{-4}$	0.91
$2^4 \cdot 100$	0.014	0.65	$1.9 \cdot 10^{-4}$	0.92
$2^5 \cdot 100$	0.008	0.69	$9.8 \cdot 10^{-5}$	0.94
$2^6 \cdot 100$	0.005	0.80	$5.0 \cdot 10^{-5}$	0.96

Table 6.5: Grid convergence study for α_2 at $t = 1 \cdot 10^5$ years

6.5 Performance

Apart from grid convergence the benchmark in Appendix A also aims to compare the performance of different code implementations. For the simulations following values were chosen for the Newton solver (confer 5.5) and the time stepping scheme:

- $\epsilon_\mathrm{nl} = 10^{-6}$
- $\epsilon_\mathrm{nl}^\mathrm{abs} = 0$
- $\epsilon_\mathrm{lin} = 0$ (a direct solver (SuperLU) is used for this 1D problem)
- $\max_\mathrm{newtoniterations} = 10$
- $\max_\mathrm{linesearch} = 30$
- $\lambda = 0.5$ (dampening factor)
- $\mathrm{d}t_\mathrm{initial} = 5\,000$ years
- $\mathrm{d}t_\mathrm{max} = 50\,000$ years
- $\mathrm{d}t_\mathrm{scale}^\mathrm{down} = 0.5$, $\mathrm{d}t_\mathrm{scale}^\mathrm{up} = 1.2$

Table 6.6 and 6.7 display the number of successful and failed time steps and Newton steps. Additionally the wall time for the simulation is given. All simulations were performed sequentially.

N	time steps (failed)	Newton steps (failed)	wall clock time [s]
$2^0 \cdot 100$	48 (7)	309 (70)	1.8
$2^1 \cdot 100$	62 (11)	422 (110)	4.9
$2^2 \cdot 100$	76 (15)	511 (150)	12.6
$2^3 \cdot 100$	89 (18)	630 (180)	38.7
$2^4 \cdot 100$	92 (19)	643 (190)	61.6
$2^5 \cdot 100$	89 (18)	629 (180)	121.3

Table 6.6: Benchmark performance results for $\alpha_1 = 5e^{-7}$

N	time steps (failed)	Newton steps (failed)	wall clock time [s]
$2^0 \cdot 100$	213 (55)	1335 (550)	8.5
$2^1 \cdot 100$	403 (107)	2641 (1070)	37.5
$2^2 \cdot 100$	781 (206)	5535 (2060)	141.1
$2^3 \cdot 100$	1477 (390)	10701 (3990)	518.0
$2^4 \cdot 100$	2827 (746)	20751 (7440)	2043.6
$2^5 \cdot 100$	5488 (1441)	38942 (13699)	7611.3

Table 6.7: Benchmark performance results for $\alpha_2 = 5e^{-4}$

Table 6.8 and 6.9 show the results obtained at Friedrich-Alexander University of Erlangen-Nuremberg FAU (Fabian Brunner and Peter Knabner).

They use capillary pressure p_c and the molar density concentration ν_w^b as primary variable. The domain is discretized with a combination of linear finite elements and vertex centered finite volume scheme. They perform a 2D simulation with $(n_x + 1) \cdot (n_y + 1)$ degrees of freedom and use $n_y = 1$. Thus, they have approximately two times the number of degrees of freedom compared to the cell centered finite volume scheme used in this work, where the number of freedoms corresponds to the number of cells. Furthermore, they use the following parameters for their simulation:

- $\epsilon_{\text{nl}} = 10^{-6}$
- $\epsilon_{\text{nl}}^{\text{abs}} = 0$
- $\max_{\text{newtoniterations}} = 30$
- $\max_{\text{linesearch}} = 2$
- $\mathrm{d}t_{\text{initial}} = 10$ years
- $\mathrm{d}t_{\max} = 1\,000\,000$ years
- $\mathrm{d}t_{\text{scale}}^{\text{down}} = 0.5$, $\mathrm{d}t_{\text{scale}}^{\text{up}} = 10/s$ with s number of Newton steps in last time step

The numbers in Table 6.8 and 6.9 show a very similar behavior compared to the results in this work (Table 6.6 and 6.7). The simulations for test case α_2 show that significantly more time steps and Newton steps are needed compared to test case α_1. For α_2 the number of Newton steps approximately doubles for each level, where it only increases very slowly for α_1.

N	time steps (failed)	total Newton steps
$2^0 \cdot 100$	23 (5)	156
$2^1 \cdot 100$	30 (9)	220
$2^2 \cdot 100$	47 (17)	370
$2^3 \cdot 100$	63 (26)	523
$2^4 \cdot 100$	53 (16)	415
$2^5 \cdot 100$	78 (35)	654

Table 6.8: FAU (Fabian Brunner and Peter Knabner): Benchmark performance results for $\alpha_1 = 5e^{-7}$

N	time steps (failed)	total Newton steps
$2^0 \cdot 100$	107 (25)	1100
$2^1 \cdot 100$	175 (38)	1717
$2^2 \cdot 100$	300 (74)	3167
$2^3 \cdot 100$	540 (56)	5584
$2^4 \cdot 100$	1000 (204)	10816
$2^5 \cdot 100$	2048 (620)	22823

Table 6.9: FAU (Fabian Brunner and Peter Knabner): Benchmark performance results for $\alpha_2 = 5e^{-4}$

FAU reports results where the improved Newton solver developed in Section 5.5 is used instead of the standard one. They also observe much better results for the modified version for α_2, confer Table 6.11.

In total, the simulations of FAU clearly need less time steps and Newton steps. Instead of deriving the Jacobian by numerical differentiation, FAU uses the exact analytical Jacobian. Furthermore, the exact Newton and time stepping scheme customization is very different from the simulations in this work and can have a very big impact on the simulation results. A predefined Newton setup and time stepping scheme configuration would be helpful for a better comparison.

N	time steps (failed)	total Newton steps
$2^0 \cdot 100$	21 (2)	153
$2^1 \cdot 100$	20 (1)	144
$2^2 \cdot 100$	18 (3)	123
$2^3 \cdot 100$	29 (9)	246
$2^4 \cdot 100$	31 (12)	253
$2^5 \cdot 100$	32 (13)	263

Table 6.10: FAU (Fabian Brunner and Peter Knabner): Benchmark performance results for $\alpha_2 = 5e^{-7}$ with modified Newton solver

N	time steps (failed)	total Newton steps
$2^0 \cdot 100$	93 (11)	994
$2^1 \cdot 100$	167 (25)	1756
$2^2 \cdot 100$	273 (46)	3060
$2^3 \cdot 100$	454 (64)	4923
$2^4 \cdot 100$	883 (166)	9659
$2^5 \cdot 100$	1490 (380)	16626

Table 6.11: FAU (Fabian Brunner and Peter Knabner): Benchmark performance results for $\alpha_2 = 5e^{-4}$ with modified Newton solver

In this Chapter a simple 1D setup from the MoMas benchmark is used as application for the code. The experimental order of convergence for the implemented finite volume method is optimal. Other research groups' output matches the results of the simulation in this work very well. This indicates that the implementation in this work provides accurate solutions. Additionally, a further benchmark is proposed. This benchmarks is based on the MoMas benchmark, but considers two different van Genuchten curves. Performance markers and a grid convergence study for this benchmark are provided. A comparison with the results from FAU shows a very similar behavior. FAU also observes much better results with the modified Newton solver developed in this work.

7 Test Case 2: CO_2 Injection into a Fully Water Saturated Domain (2D)

In the second test case, CO_2 is injected into a 2D domain. Here, the physical properties of CO_2 and water from Chapter 3 are employed and tested. A strong scalability test shows the parallel performance of the simulation code.

7.1 Parameters and Setup

In the second test case, CO_2 is injected into the lower left part of a rectangular geometry ($600\,\mathrm{m} \times 100\,\mathrm{m}$) with a flux of $w_{\mathrm{in}}^{\mathrm{b}}$. The domain is located 800 m under the surface. As in test case 1, the top and bottom of the domain have no-flux boundary conditions (see Figure 7.1). For the Dirichlet boundary on the right side, hydrostatic pressure for the water phase and zero pressure for the CO_2 phase is chosen (that corresponds to $x_{\mathrm{w}}^{\mathrm{n}} = 0$, i. e., the water phase contains no CO_2):

$$p_{\mathrm{w}}|_{\Gamma_{\mathrm{out}}} = 10^5\,\mathrm{Pa} + (900 - z)\rho_{\mathrm{mass,w}} \cdot g\,\mathrm{Pa}, \quad p_{\mathrm{b}}|_{\Gamma_{\mathrm{out}}} = 0\,\mathrm{Pa},$$

where z is the z coordinate in the domain and g the gravity in z-direction. Again, the values on Γ_{out} are taken as initial values.

Densities, viscosities and solubilities are chosen as suggested in Chapter 3. All other parameters are given in Table 7.1. For the computations, a structured grid with 240×40 cells is used.

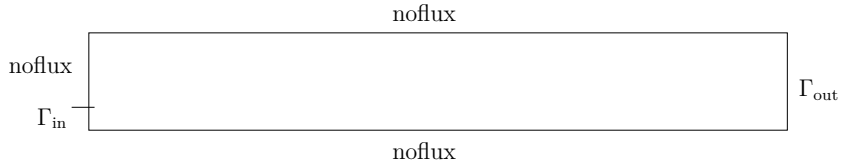


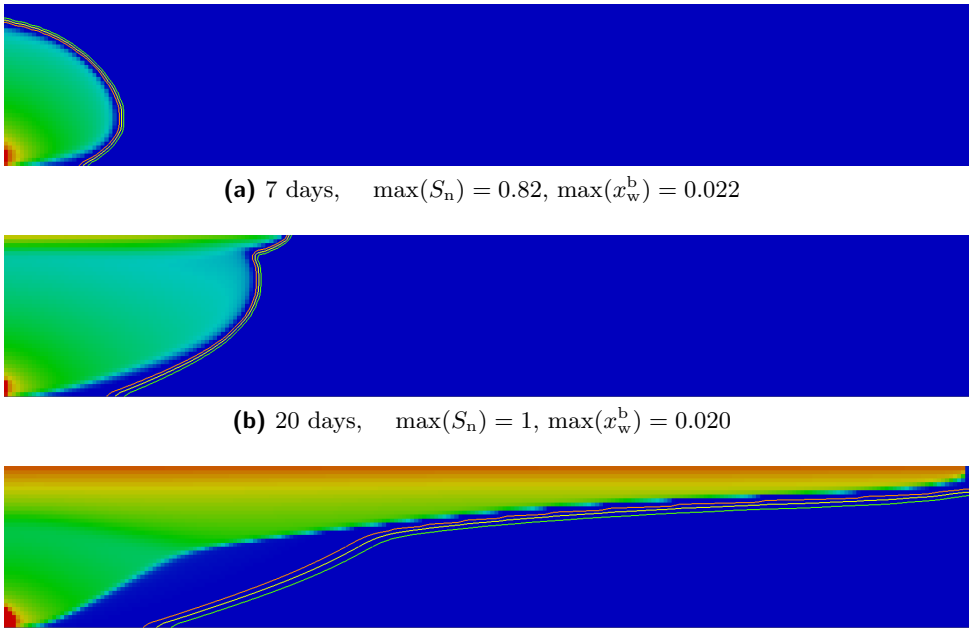
Figure 7.1: Domain setup for test case 2

parameter	symbol	value	unit
porosity	ϕ	0.2	
salinity	s_{sal}	0	
temperature	T	313.15	K
van Genuchten parameter	$S_{\alpha,\text{res}}$	0	
van Genuchten parameter	n	6	
van Genuchten parameter	α	$6.5 \cdot 10^{-4}$	Pa^{-1}
absolute permeability	K	$1 \cdot 10^{-12}$	m^2
source/sink term	q^κ	0	$\text{kg}\,\text{m}^{-2}\,\text{s}^{-1}$
molecular diffusion coefficient	D_{w}^{b}	$2 \cdot 10^{-9}$	$\text{m}^2\,\text{s}^{-1}$
molecular diffusion coefficient	D_{n}^{a}	$2 \cdot 10^{-9}$	$\text{m}^2\,\text{s}^{-1}$
molar mass	M^{a}	$1.8 \cdot 10^{-2}$	$\text{kg}\,\text{mol}^{-1}$
molar mass	M^{b}	$4.4 \cdot 10^{-2}$	$\text{kg}\,\text{mol}^{-1}$
flux of CO_2 at Γ_{in}	w_{in}^{b}	$4 \cdot 10^{-2}$	$\text{kg}\,\text{m}^{-2}\,\text{s}^{-1}$
flux of water at Γ_{in}	w_{in}^{a}	0	$\text{kg}\,\text{m}^{-2}\,\text{s}^{-1}$

Table 7.1: Parameters for test case 2

7.2 Numerical Results

The results of test case 2 are shown in Figure 7.2. Each picture shows the CO_2 phase saturation at a specific time point. The contour lines depict the molar fraction of CO_2 in the water phase. The CO_2 migrates upwards until it reaches the top of the domain with the no-flux conditions and is then driven to the right by advective forces.



(a) 7 days, $\max(S_n) = 0.82$, $\max(x_w^b) = 0.022$

(b) 20 days, $\max(S_n) = 1$, $\max(x_w^b) = 0.020$

(c) 65 days, $\max(S_n) = 1$, $\max(x_w^b) = 0.020$

Figure 7.2: Test case 2: CO_2 phase saturation and molar fraction of dissolved CO_2 in water (contour lines for $x_w^b = 0.005, 0.011, 0.016$). Color scale ranges from $S_n = 0$ (blue) to $S_n = \max(S_n)$ (red).

7.3 Strong Scalability

All simulations are performed in parallel. To analyze the parallel performance of the simulations, a strong scalability test is conducted, where the global problem size stays fixed and the number of processes is increased. For a weak scalability test (see for example in Section 8.3), the global problem size in each direction is doubled in each level. Simultaneously the number of processes in each direction is doubled, so the number of cells per process stays constant.

The parallel efficiency is defined as

$$E = \frac{T_1}{pT_p},$$

#processes	total time [s]	efficiency
1	13975	1
2	7763	0.90
4	4151	0.84
8	2658	0.65

Table 7.2: Strong scalability test

#processes	TS	ADT [s]	MDT [s]	ANI
1	2249	3579.7	156.25	3.9
2	2276	3527.6	156.25	3.9
4	2205	3593.1	312.5	3.9
8	2282	3514.4	312.5	3.9

Table 7.3: Performance indicators for strong scalability test

where T_1 is the time for the sequential method, p the number of processes and T_p the time for the parallel method with p processes. The efficiency for one process is $E = 1$ by definition. With more processes, the problem size for one process becomes smaller but due to the parallel overhead (interprocess communication), the efficiency decreases. It is important that the problem size for one process does not get too small, otherwise the communication overhead dominates.

Table 7.2 shows the results for test case 2 for a simulation time of 65 days. The total time needed for solving the problem scales very well with the number of processes and the efficiency only slowly decreases with the number of processes.

To facilitate a comparison with other implementations, important performance indicators are listed in Table 7.3. TS is the number of time steps performed (both successful and unsuccessful). For average and minimum time step sizes only the successful time steps are regarded. Table 7.3 shows that average time step size (ADT), average number of Newton iterations (ANI) and minimum time step size (MDT) stay almost constant for different numbers of processes.

8 Test Case 3: CO_2 Injection into a Fully Water Saturated Domain (3D)

This section shows numerical results for a CO_2 injection setup in 3D. The setup is very close to that of Chapter 7. Naturally, the 3D test has a large number of unknowns in comparison to a 2D test. A Weak Scalability test, which is an important tool to measure the parallel performance of the code, is part of this chapter.

8.1 Parameters and Setup

For test case 3, we use the same parameters and a very similar setup as in test case 2 in Section 7.1. The difference is that we look at a 3D domain as shown in Figure 8.1.

The domain is a cube with dimensions $120\,\mathrm{m} \times 120\,\mathrm{m} \times 120\,\mathrm{m}$. The source Γ_{in} is $10\,\mathrm{m} \times 10\,\mathrm{m}$. For the computations, a structured grid with $60 \times 60 \times 60$ cells is used.

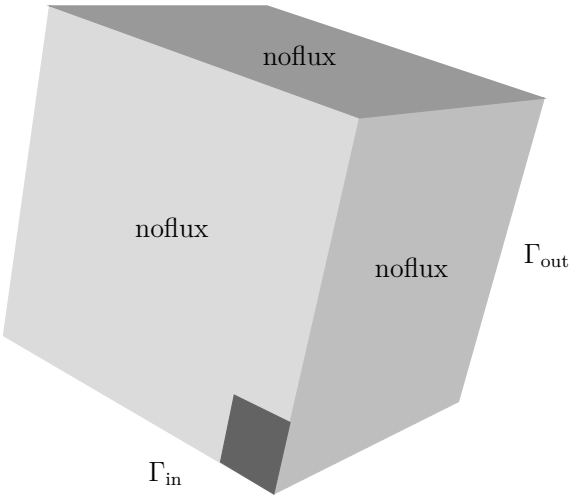


Figure 8.1: Domain setup for test case 3

8.2 Numerical Results

The results of test case 3 are shown in Figure 8.2. As in test case 2, each picture shows the CO_2 phase saturation. The CO_2 rises until it reaches an impermeable layer at the top of the domain. Then it spreads throughout the top. The solubility of CO_2 in the water phase is given by the contour lines. After 18 days a noticeable amount of water contains dissolved CO_2.

8.3 Weak Scalability

Weak scale tests on different parallel computers and for different simulation setups are used to ensure the parallel efficiency of the code. In contrast to the strong scalability test in Section 7.3, the problem size for one process stays fixed, whereas the global problem size increases.

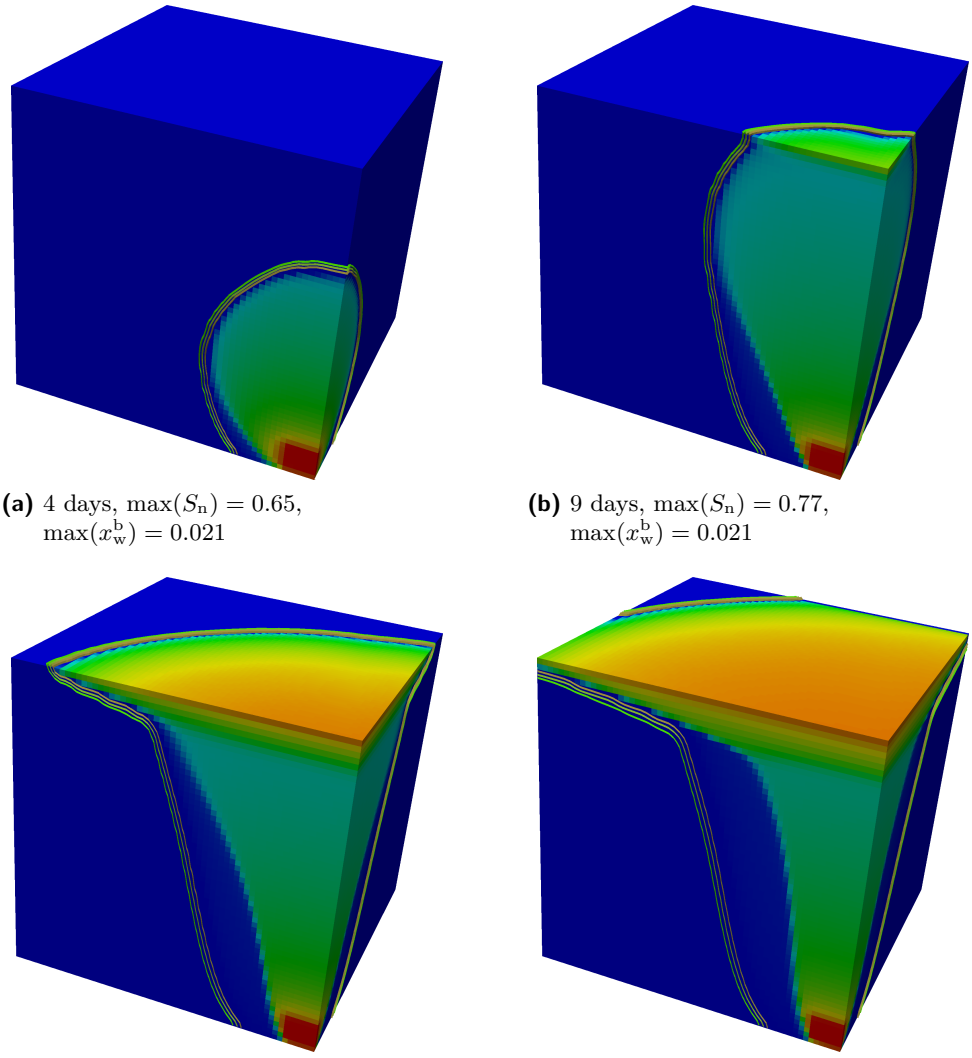
As computation domain, a cube of $100\,\mathrm{m} \times 100\,\mathrm{m} \times 100\,\mathrm{m}$ is used. For the sequential version, a cube grid of $50 \times 50 \times 50$ elements is used. The grid is then globally refined everywhere (that correlates to eight times more grid cells in total). For the sake of accuracy, the time step size has to be chosen such that the CO_2 front travels about one grid cell layer per time step. To achieve this, the allowed maximum time step size is halved for each level.

Weak scale tests were performed on two different parallel computers, helics3a located at IWR, University of Heidelberg, and hermit (CrayXE6) at the HLRS, University of Stuttgart.

The technical details of the two supercomputers are summarized in Table 8.1. Table 8.2 shows the results of the weak scaling test for a fixed wall clock time of 20 500 s (5.7 h).

Machine name	helics3a	hermit
# nodes	32	3552
CPU sockets per node	4	2
RAM per node	128 GB	32 GB
total # cores	1024	113 664
CPU socket	AMD Opteron 6216 at 2.6 GHz	AMD Opteron 6276 at 2.3 GHz
# cores per CPU	8	16
Launch date	May 2012	December 2011
InfiniBand	Mellanox 40G QDR single port PCIe Interconnect QSFP	Cray Gemini

Table 8.1: Supercomputers used for the scalability tests



(a) 4 days, $\max(S_n) = 0.65$, $\max(x_w^b) = 0.021$

(b) 9 days, $\max(S_n) = 0.77$, $\max(x_w^b) = 0.021$

(c) 14 days, $\max(S_n) = 0.82$, $\max(x_w^b) = 0.021$

(d) 18 days, $\max(S_n) = 0.82$, $\max(x_w^b) = 0.021$

Figure 8.2: CO_2 phase saturation and molar fraction of dissolved CO_2 in water for test case 3 (contour lines for $x_w^b = 0.005, 0.010, 0.016$)
Color scale ranges from $S_n = 0$ (blue) to $S_n = \max(S_n)$ (red).

#p	N	TS	ADT[s]	ST[s]	ANI	ALI	TIT[s]
1	50^3	310	279	86400	3.0	4.7	0.6
8	100^3	176	140	24430	3.8	5.6	1.1
64	200^3	116	68	7892	4.0	5.7	1.2
512	400^3	82	33	2690	4.3	6.2	1.5

Table 8.2: Weak scalability test on helics3a for test case 3

#p	N	TS	ADT[s]	ST[s]	ANI	ALI	TIT[s]
1 (only 4.5h)	50^3	310	279	86400	3.0	5.0	0.4
8	100^3	165	140	23030	3.7	5.6	1.2
64	200^3	99	67.3	6668	4.0	5.7	1.4
512	400^3	78	31.7	2417	4.2	6.8	1.5

Table 8.3: Weak scalability test on hermit for test case 3

The symbols in the table have the following meaning:

N number of cells
TS number of time steps
ADT average time step size
ST simulation time
ANI average number of Newton iterations
ALI average number of linear iterations per Newton iteration
TIT average time for a linear iteration

In this weak scalability test, in each level the mesh is refined by a factor of two in each spatial dimension and correspondingly the number of processors is increased by eight. The time step size needs to be decreased for smaller cells. Thus, the time that is simulated for the same number of time steps is less. In total, 512 processors compute 82 time steps compared to 310 for 1 processor (factor of 3.8 less).

For an optimal method without communication, ANI, ALI and TIT would stay constant. For 512 processors, the number of Newton steps and the number of linear iterations do only slightly increase compared to 1 processor. However, the average time for a linear iteration increases almost by a factor of three. This increase is the reason that less time steps are performed for 512 processors.

An explanation for the big increase of the average time per linear iteration could be the Bulldozer microprocessor architecture of the AMD Opteron 6200 series, that is used in both supercomputers (confer Table 8.1). Bulldozer shares one floating point execution unit between two threads (confer [20]) which reduces the floating point performance.

The same weak scalability test performed on hermit gives similar results (see Table 8.3). In summary: Going from the sequential (1 process) to the parallel version (8 processes) there is an expectable drop, especially for the TIT. After that, the code scales very well on both helics3a and hermit.

9 Test Case 4: Svalbard Benchmark

As last example a recent benchmark studied at a workshop in Svalbard is chosen. The Svalbard benchmark is chosen as a possibility for a setup with a large number of unknowns.

One of the main goals is to test the parallel capability of the code on the supercomputer hermit (CrayXE6) at the HLRS, University of Stuttgart and to look at the spreading of the CO_2 plume for different grid resolutions.

A weak scalability test is presented to evaluate the parallel efficiency of the code. Furthermore, a new benchmark is proposed. This new benchmark uses a very simple 2d setup and concentrates on the evolution of time step sizes after injection stop.

9.1 Parameters and Setup

The complete description of the benchmark can be found in [24]. Results of different groups are available (see [52]). The CO_2 injection stops after a certain time span. The aim of the benchmark is to track the movement of the CO_2 plume long after injection stop.

In contrast to the MoMas benchmark discussed in Chapter 6, many modeling decisions are left open. Fluid properties like density or viscosity are not defined. Additionally, even miscibility is not mandatory. One of the aims of the Svalbard benchmark is to compare the influence of the specific modeling on the numerical results. But the great liberty in modeling choices also means, that the complexity and with that the specific numerical challenges of the different implementations are very different.

In order to cover the large time span required by the benchmark (10 000 years), the participating groups used different approaches to simplify the numerical simulation. Many groups do not include solubility or compressibility in their simulations and use a simple two-phase flow model. One group (University of North Carolina) uses vertical upscaling as a possibility to reduce the complexity to that of a 2D problem. Furthermore, many groups simply use a very small grid. Only Potsdam reports results for a relatively large grid (1 million cells) that includes solubility of CO_2 in water.

The emphasis of this work is a very large grid resolution together with a model that includes solubility trapping. Therefore, solubility of CO_2 in water is included in the simulations. Furthermore, the complex physical behavior of, e. g., density and viscosity is part of the model. A fine grid is chosen to resolve the large domain and large parallelization is employed to make the numerical simulation feasible in a reasonable time.

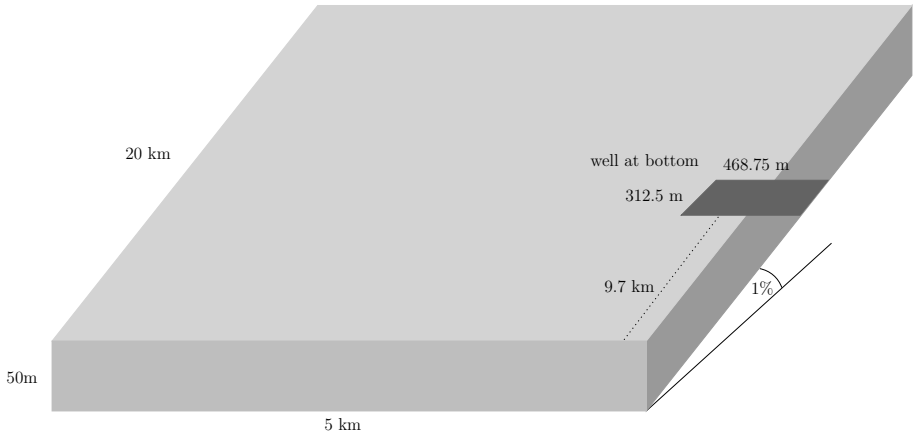


Figure 9.1: Domain setup for test case 4

One difficulty of this particular benchmark is the very anisotropic domain. The original benchmark definition has an extent of 100 km × 200 km but only 50 m in z-direction. A well is located at the center of the bottom of the domain.

Initial tests show a close relation between the size of the cells and the time step size. For a very fine grid resolution the time step size gets very small. In order to use a large number of cells only the first 20 years of the benchmark can be covered in reasonable time. Another observation is, that the CO_2 does not travel far during the first years and it is enough to model only part of the domain. Furthermore, it is sufficient to only simulate half of the domain as the whole setup is symmetric in x-direction. Figure 9.1 shows the domain that is used for the simulations (note that the domain has a constant tilt of 1%).

The simulations are performed in parallel using 512 processes. A structured grid with $1024 \times 256 \times 16$ cells ($\approx$ 4.2 million cells) for a domain of $20\,000\,\mathrm{m} \times 5000\,\mathrm{m} \times 50\,\mathrm{m}$ is used. The well size is chosen as $312.5\,\mathrm{m} \times 468.75\,\mathrm{m}$ for half the domain, that corresponds to 16×24 cells. The setup is almost the same as for the weak scalability test in Section 9.3. This explains the choice of the size of the well. For the scalability test, twice as many cells in each direction (that is 8 times as many in total) are used, because only a small time span is modeled (see Section 9.3).

As in test cases 2 and 3, water is the wetting component and CO_2 the nonwetting component. The solubility of water in the nonwetting phase is neglected: $x_{\mathrm{n}}^{\mathrm{a}} = 0$. For 20 years 1 megaton CO_2 per year (that corresponds to $31.71\,\mathrm{kg\,s^{-1}}$) is injected from the well (for the simulation only half that amount is injected, because only half the domain is simulated). The top of the domain is located 2500 m under the surface.

The gravity vector is in general given by:

$$g = (0, 0, -9.81)\mathrm{m\,s^{-2}}$$

To include the dip of $\beta = \arctan(1\%) = 0.5729°$ of the domain in the simulation, the

parameter	symbol	value		unit
porosity	ϕ	0.15		
salinity	s_{sal}	0		
temperature	T	345.65		K
Brooks-Corey parameter	$S_{\text{w,res}}$	0.2		
Brooks-Corey parameter	$S_{\text{n,res}}$	0.1		
Brooks-Corey parameter	τ	1		
Brooks-Corey parameter	λ	2		
Brooks-Corey parameter	p_{entry}	2	$\cdot 10^4$	Pa
absolute permeability	K	1	$\cdot 10^{-13}$	m^2
source/sink term	q^κ	0		$\text{kg}\,\text{m}^{-2}\,\text{s}^{-1}$
molecular diffusion coefficient	D_{w}^{b}	1	$\cdot 10^{-9}$	$\text{m}^2\,\text{s}^{-1}$
molecular diffusion coefficient	D_{n}^{a}	1	$\cdot 10^{-9}$	$\text{m}^2\,\text{s}^{-1}$
molar mass	M^{a}	1.8	$\cdot 10^{-2}$	$\text{kg}\,\text{mol}^{-1}$
molar mass	M^{b}	4.4	$\cdot 10^{-2}$	$\text{kg}\,\text{mol}^{-1}$
flux of water at Γ_{in}	w_{in}^{a}	0		$\text{kg}\,\text{m}^{-2}\,\text{s}^{-1}$
flux of CO_2 at Γ_{in}	w_{in}^{b}	15.855		$\text{kg}\,\text{m}^{-2}\,\text{s}^{-1}$

Table 9.1: Parameters for Svalbard benchmark

gravity vector can be adapted as follows:

$$g = \left(-\sin\left(\frac{\beta\pi}{180}\right) \cdot 9.81, 0, -\cos\left(\frac{\beta\pi}{180}\right) \cdot 9.81 \right) \text{m}\,\text{s}^{-2}.$$

The boundary conditions are similar to those in test case 3. No-flux boundary conditions are chosen at the top and bottom of the domain. As only half of the domain is simulated, no-flux conditions are also used at the right side of the domain where the well is located. For the Dirichlet boundaries on the remaining sides, hydrostatic pressure for the water phase and zero pressure for the CO_2 phase (which leads to $x_{\text{w}}^{\text{n}} = 0$) are used:

$$p_{\text{w}}|_{\Gamma_{\text{out}}} = 10^5\,\text{Pa} + (2550 - z)\rho_{\text{mass,w}} \cdot g\,\text{Pa}, \quad p_{\text{n}}|_{\Gamma_{\text{out}}} = 0\,\text{Pa},$$

where z is the z-coordinate in the domain and g the gravity in z-direction. The z-coordinate has to be computed according to the dip of the domain. Again the same values are taken as initial values.

According to the benchmark description (see [24]) the Brooks-Corey model is used (confer Section 3.1) for the water retention curve. The Svalbard benchmark uses a slightly different formula for the computation of the relative permeability of the nonwetting phase:

$$k_{\text{rn}}(S_{\text{n}}) = 0.4\,S_{\text{n}}^2\left(1 - (1 - S_{\text{n}})^2\right).$$

All other parameters used in the simulation are listed in Table 9.1.

9.2 Numerical Results

The simulation is performed in parallel on 512 cores (16 nodes with 32 cores, see also Table 8.1 in Chapter 8) on hermit (CrayXE6) at the HLRS, University of Stuttgart. On hermit there are usage time restrictions (i. e., jobs are only allowed to run for 24 hours). With the large number of cells and the long simulation time of this setup, the simulation is not finished after 24 hours, therefore a restarting mechanism has to be implemented. At certain checkpoints each process writes the current solution on all degrees of freedom into a file. The simulation can then be restarted from the last checkpoint. For the restarting mechanism, every 500th time step is chosen as checkpoint (and later every 250th time step).

For the first 1200 time steps the overall computation time is 14 000 s, so the average computation time for one time step is 11.7 s. The mean time needed for writing the output is 2.82 s. That means that writing output increases the computation time for one time step by 25%. So, in order to reduce computation time, the output is not stored for every time step. The data in each degree of freedom is written out each month (1 month = 2 628 000 s).

Table 9.2 shows the number of time steps simulated in each run (24 hours) and the average time step size (ADT) for the first 10 runs. After a short period in the beginning the average time step size decreases slowly over time. The reason for that behavior is, that the surface of the CO_2 front increases over time. With that also the number of cells that potentially switch from one phase to two phases increases. Due to the kink in the pressure evolution at the one-phase/two-phase interface described in Section 5.5, the time step size cannot grow too big.

Figure 9.2 and 9.3 show a comparison of the spread of the CO_2 plume for different numbers of cells. Each picture shows the CO_2 saturation and the solubility of CO_2 in the water

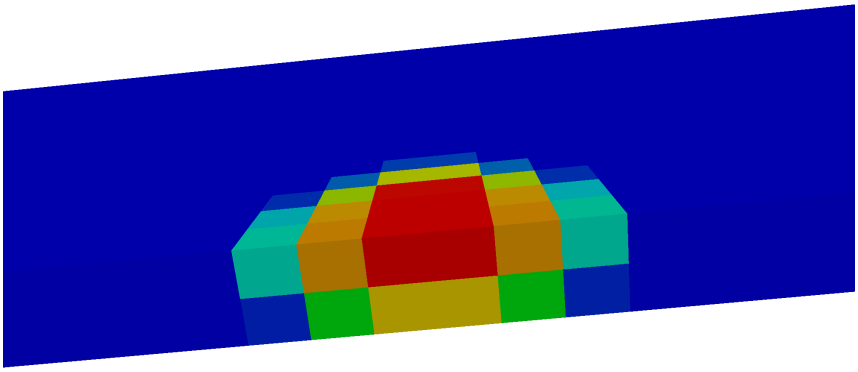
number of run	TS start	TS end	ST[years]	ADT[s]
0	1	5433	4.5	26153
1	5000	10067	5.7	8580
2	10000	15500	6.4	3510
3	15500	21096	6.9	3346
4	21000	26805	7.2	1790
5	26500	32423	7.5	1685
6	32000	37501	7.9	2058
7	37500	43803	8.2	1316
8	43750	49538	8.4	1296
9	49500	55120	8.6	1372
10	55000	60588	8.9	1369

Table 9.2: Number of time steps and average time step size per run of 24 hours

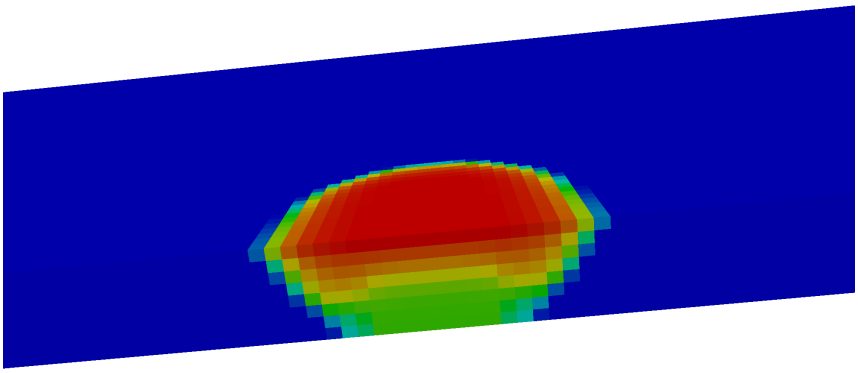
phase. Of the whole simulation domain only a part of 1.75 km is shown and the z-axis is magnified by a factor of 5 for better visibility.

Many benchmark participants use non-uniform or variable-sized grids for their simulation (confer [52]). But even so the reported cell sizes for the cells near the source have a length of at least 100 m compared to 19.5 m for the simulation in this work. Figure 9.2 and 9.3 show that the shape and the extent of the CO_2 plume differs very much for different numbers of cells. This indicates that simulations with too few cells are possibly not sufficient to predict the migration of CO_2.

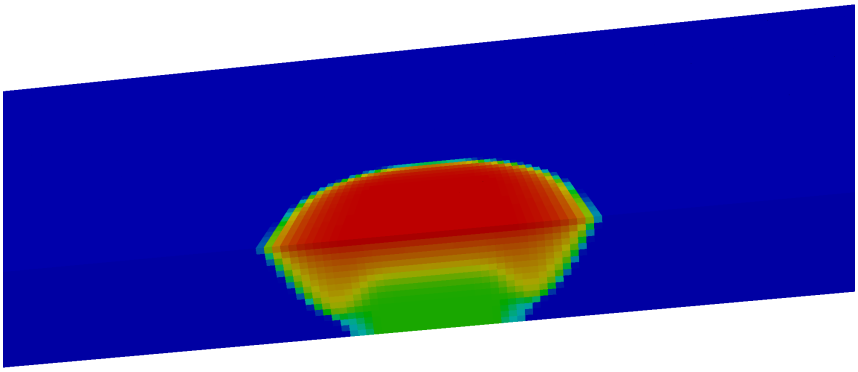
Figure 9.4 shows the results for 512 processes including the solubility of CO_2 in the water phase. As before, only a part of 1.75 km is shown and the z-axis is magnified by a factor of 5.



(a) $\#p = 1$, $N = 2^{13} \approx 8200$ cells

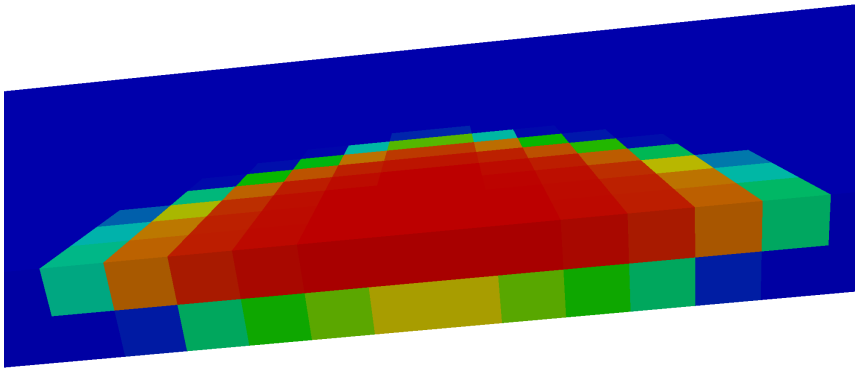


(b) $\#p = 64$, $N = 2^{19} \approx 0.5$ million cells

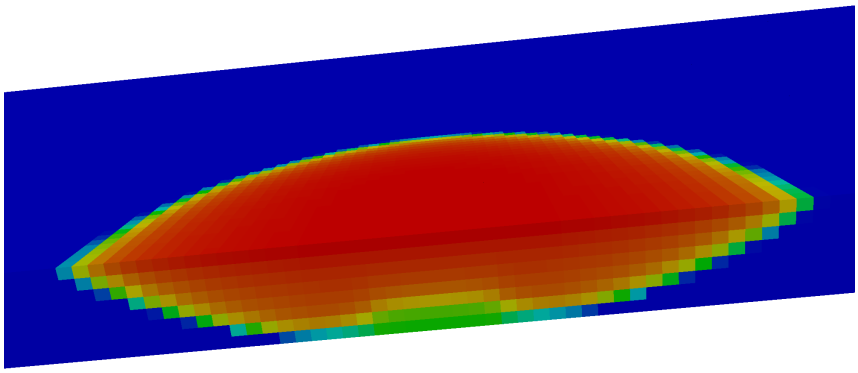


(c) $\#p = 512$, $N = 2^{22} \approx 4.2$ million cells

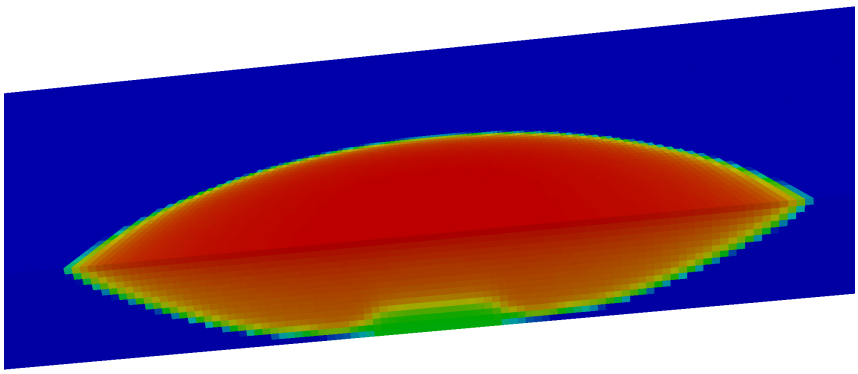
Figure 9.2: CO_2 phase saturation for different processor numbers p for $t = 1$ year, 7 months for test case 4. Only a part of 1.75 km is shown, z-axis scaled by a factor of 5.



(a) $\#p = 1$, $N = 2^{13} \approx 8200$ cells

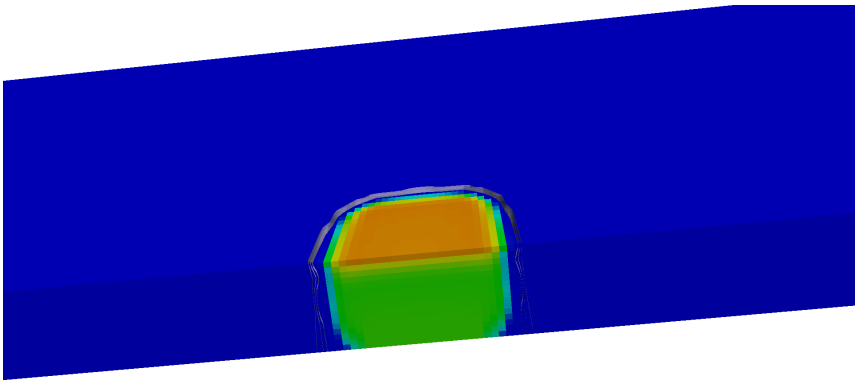


(b) $\#p = 64$, $N = 2^{19} \approx 0.5$ million cells

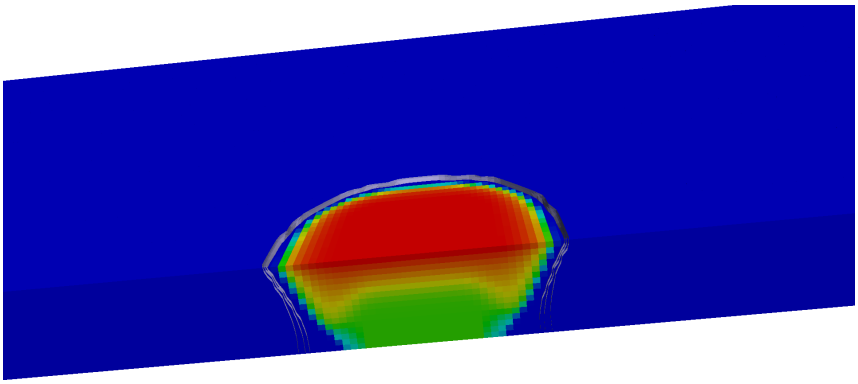


(c) $\#p = 512$, $N = 2^{22} \approx 4.2$ million cells

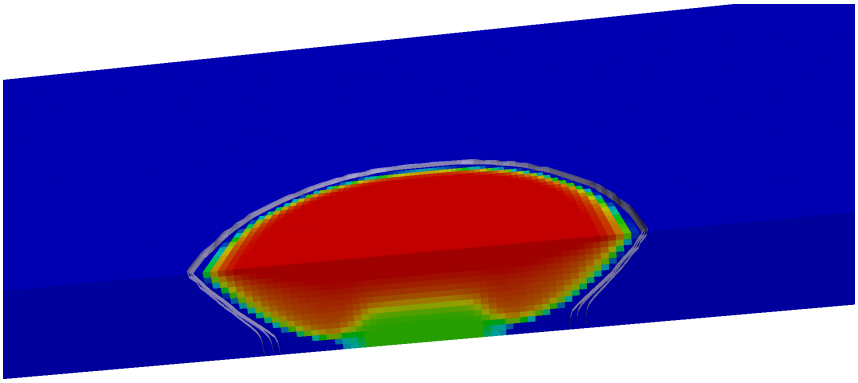
Figure 9.3: CO_2 phase saturation for different processor numbers p for $t = 5$ years, 9 months for test case 4. Only a part of 1.75 km is shown, z-axis scaled by a factor of 5.



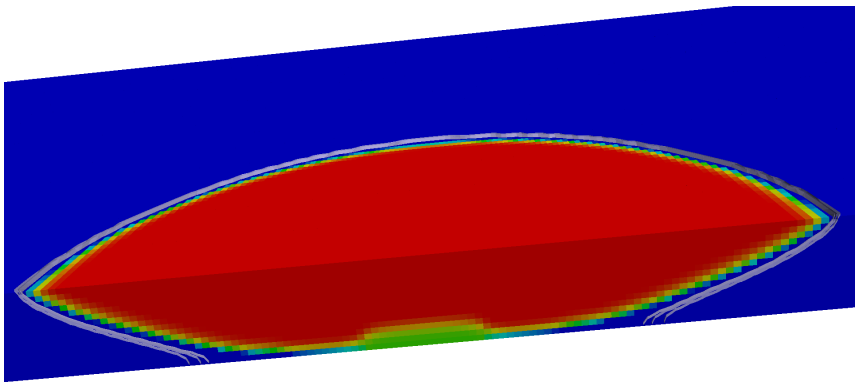
(a) 0 year, 9 months



(b) 1 years, 4 months



(c) 2 year, 7 months



(d) 7 years, 5 months

Figure 9.4: CO_2 phase saturation and molar fraction of dissolved CO_2 in water for test case 4 (contour lines for $x_w^b = 0.0025, 0.0049, 0.0074$). Only a part of 1.75 km is shown, z axis scaled by a factor of 5. Color scale ranges from $S_n = 0$ (blue) to $S_n = 0.579$ (red).

9.3 Weak Scalability

A weak scalability test is performed for the Svalbard benchmark. In contrast to the weak scalability test for test case 3 (see Section 8.3) the domain is very anisotropic. This has to be considered for the distribution of cells to processes.

9.3.1 Choice of Cell Structure and Processor Domains

If the number of unknowns per process is too small, the performance of the linear solver (AMG, confer Subsection 5.6.1) may not be optimal. Therefore, twice as many cells in each direction are used compared to Section 9.1.

This is due to the fact, that the employed linear solver (AMG, confer Subsection 5.6.1) does not perform optimal if the number of unknowns on every process is too small. Therefore, larger grid is chosen to get meaningful results.

The domain size is quite small in z-direction (50 m) compared to the other two dimensions (20 000 m $\times$ 5000 m). At the beginning the main CO_2 movement is in z-direction and there exists a strong coupling between cells in this direction. This is important for the distribution of cells to processors. Therefore, for this scalability test the domain for one process always covers all cells in one column in z-direction.

Table 9.3 displays the number of cells in each direction. A relatively conformable processor domain size is achieved by doubling and quadrupling the processor number in x- and y-direction in turns (see Table 9.4).

$\#p$	n_x	n_y	n_z	N
1	$2^8 = 256$	$2^6 = 64$	$2^2 = 4$	$2^{16} = 65\,536$
8	$2^9 = 512$	$2^7 = 128$	$2^3 = 8$	$2^{19} = 524\,288$
64	$2^{10} = 1024$	$2^8 = 256$	$2^4 = 16$	$2^{22} = 4\,194\,304$
512	$2^{11} = 2048$	$2^9 = 512$	$2^5 = 32$	$2^{25} = 33\,554\,432$
4096	$2^{12} = 4096$	$2^{10} = 1024$	$2^6 = 64$	$2^{28} = 268\,435\,456$

Table 9.3: Amount of cells for different processor numbers for weak scalability test (where n_x is number of cells in x-direction).

$\#p$	p_x	p_y	p_z	n_x^p	n_y^p	n_z^p
1	1	1	1	$2^8 = 256$	$2^6 = 64$	$2^2 = 4$
8	$2^2 = 4$	$2^1 = 2$	1	$2^7 = 128$	$2^6 = 64$	$2^3 = 8$
64	$2^4 = 16$	$2^2 = 4$	1	$2^6 = 64$	$2^6 = 64$	$2^4 = 16$
512	$2^5 = 32$	$2^4 = 16$	1	$2^6 = 64$	$2^5 = 32$	$2^5 = 32$
4096	$2^7 = 128$	$2^5 = 32$	1	$2^5 = 32$	$2^5 = 32$	$2^6 = 64$

Table 9.4: Distribution of domain to processes (where p_x is number of processes in x-direction and n_x^p number of cells of one process domain in x-direction.)

9.3.2 Results

The weak scalability test is performed on hermit (CrayXE6) at the HLRS, University of Stuttgart. The simulations have to be restarted after 24 hours (see Section 9.2) and after 12 hours for a sequential process (i. e., $\#p = 1$).

Table 9.5 shows the results of the weak scalability test for a fixed wall clock time of 12 hours.

The symbols in the table have the following meaning:

N	number of cells
TS	number of time steps
ADT	average time step size
ST	simulation time
ANI	average number of Newton iterations
ALT	average number of linear iterations per time step
TIT	average time for a linear iteration

The average time for a linear iteration (TIT) stays almost constant after the first level. However, the average number of Newton iterations (ANI) drops considerably. As ANI differs very much, instead of the average number of linear iterations per Newton iteration (ALI), average number of linear iterations per time step (ALT) is shown. That is, the average number of linear iterations per time step and not per Newton iteration is given. The table shows a distinct change beginning at 512 processes. Here, ADT drops considerably and additionally ANI is reduced to a much smaller number. The reduced ANI can be

#p	N	TS	ADT[s]	ST[days]	ANI	ALT	TIT[s]
1	$2^{16} = 65\,536$	558	622916	4023	5.96	86.3	0.16
8	$2^{19} = 524\,288$	282	301997	985	5.35	118.8	0.41
64	$2^{22} = 4\,194\,304$	231	72540	193	5.37	164.2	0.47
512	$2^{25} = 33\,554\,432$	321	4325	16	2.96	112.7	0.53
4096	$2^{28} = 268\,435\,456$	233	2107	5	2.46	149.6	0.61

Table 9.5: Weak scalability test on hermit for test case 4 (Svalbard benchmark)

explained by the much smaller ADT. With smaller time steps the nonlinear problem is easier to solve, and a smaller number of Newton steps is sufficient for convergence.

During grid refinement, the volume of each cell is reduced by a factor of 8 while the surface only decreases by a factor of 4. The CO_2 moves mainly in vertical direction after an initial upwards movement. Therefore, at specific cells (e. g., at the front of the CO_2 plume) in each refinement level the CO_2 flux doubles in relation to the cell volume. That means, for small cells (and with that a larger number of processes) a large amount of CO_2 has to be distributed to a relatively small volume. Presumably, this means that the nonlinear problem is harder to solve and can explain the distinct decrease of ADT for 512 processes.

9.4 Convergence Problems after Injection Stop

Previous tests (see Section 5.5) showed that during injection the time step size closely relates to number of new cells switching from one-phase to two-phase and thus the cell size. After injection stop, the time step size should increase because the CO_2 front moves more slowly and less cells switch to two-phase.

Tests in preparation for the Svalbard benchmark showed a very different behavior. During injection the time step stays more or less constant. After injection stop the time step first increases as expected followed by a breakdown of the step size (smaller than during injection).

To simplify the analysis of this problem, a 2D setup of the Svalbard benchmark is chosen: The domain size is $20\,000\,\mathrm{m} \times 50\,\mathrm{m}$. The domain is a 2D section of the 3D setup in Figure 9.1 and matches the rightmost surface in x-z-direction. All other parameters are not changed compared to Section 9.1. The simulation is performed in parallel.

Figure 9.5 shows the time step size evolution for three different simulations. The first two use a different number of cells in x direction, the third one uses a grid with nonuniform rectangles (i. e., the cells are smaller near the well).

All three show the same behavior described above. The time step size is more or less constant until injection stop at approximately $0.6 \cdot 10^9$ s. After an intermittent increase,

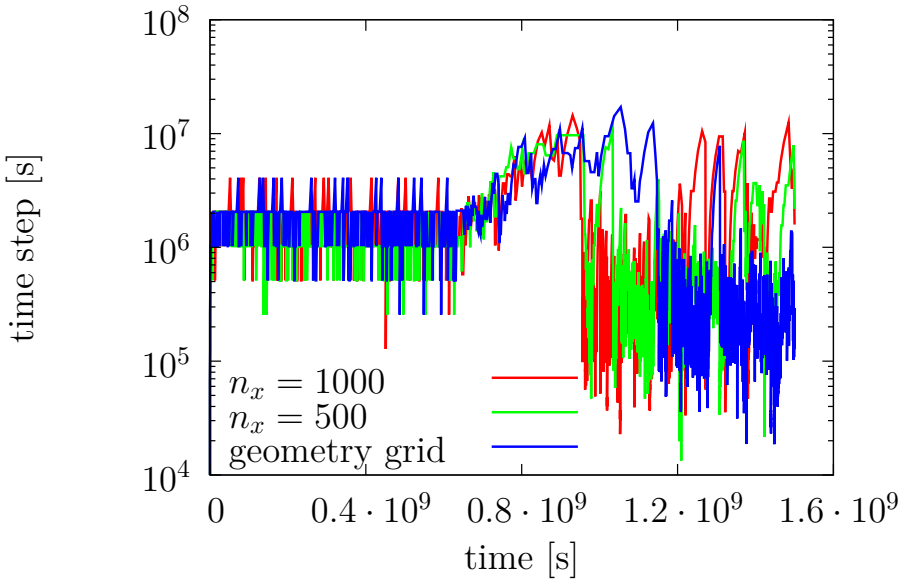


Figure 9.5: Time step evolution for a 2D version of test case 4 for three different grids

the time step size drops down to very small values. These small time step sizes make large scale simulations with fine grids impossible because the computation takes too long.

This behavior can only be observed for at least 2D simulations (more than one cell in z-direction) and for simulations including miscibility. Many participants of the Svalbard benchmark used no miscibility or upscaling (i. e., only one cell in z-direction) and thus did not experience this problem. If some of the remaining groups, that used miscibility in combination with a larger grid did observe a similar behavior is unknown, because the participants did not report information about their average time step size or overall computation time.

Many attempts to solve this problem have been made in the course of this work:

- using different versions of Henry's law (a linear function) for the solubility instead of the complex constitutive relation described in Section 3.2
- using the wetting phase pressure instead of the nonwetting phase pressure as primary variable (p_c/p_w formulation)
- using solubility as primary variable (p_c/x_w^b formulation)
- employing variable switching instead of persistent variables
- applying constant densities
- neglect of upwinding
- using sequential instead of parallel simulations

Nevertheless, for all attempts the time step breaks down after the injection phase.

parameter	symbol	value	unit
initial wetting phase pressure	p_{w}^{i}	$10^5 + z \cdot \rho^a \cdot 9.81$	Pa
injection time	T_{inj}	10	years
flux of water at Γ_{in}	$w_{\mathrm{in}}^{\mathrm{a}}$	0	$\mathrm{kg\,m^{-2}\,s^{-1}}$
absolute permeability	K	$5 \cdot 10^{17}$	m^2

Table 9.6: Parameters for proposed 2D benchmark that differ from definitions in Appendix A

9.5 Proposal for a Simple 2D Benchmark to Investigate Time Step Evolution

To investigate the time step behavior observed in Section 9.4 a new benchmark is proposed: To simplify the analysis of this problem, a 2D version of the benchmark presented in Appendix A is developed. The setup is basically a 2D version of the MoMas benchmark which has the advantage that the simple Henry's law is used for the solubility. Thus, it is easy to implement because no complex constitutive relations (as, e. g., for carbon capture and storage (CCS), see Chapter 3) are needed. An additional simplification is the neglect of diffusion.

The test case is two-dimensional in space ($200\,\mathrm{m} \times 1\,\mathrm{m}$), and the time interval to simulate is $\Sigma = (0, 160)$ years. There are no sources and sinks inside the domain, i.e.,

$$q_\alpha^\kappa = 0. \tag{9.1}$$

Initial conditions are completely water-saturated with hydrostatic pressure:

$$p_{\mathrm{w}}^{\mathrm{i}} = p_0 = 10^5\,\mathrm{Pa} + z \cdot \rho^{\mathrm{a}} \cdot 9.81\,\mathrm{Pa}$$

$$p_{\mathrm{n}}^{\mathrm{i}} = 0\,\mathrm{Pa}$$

at the left boundary. The top of the domain is at 500 m depth. Dirichlet boundary conditions on Γ_{out} are chosen corresponding to the initial conditions. On the top and bottom the domain is closed and no-flux boundary conditions are chosen.

The nonwetting component is injected into the left side of the domain Γ_{in} with a flux of q_0 for 10 years. All parameters that are different from Appendix A are listed in Table 9.6.

For the numerical simulation a grid of 125×2 cells is chosen. The initial time step size is $\mathrm{d}t_{\mathrm{initial}} = 15\,768\,000\,\mathrm{s}$. After successful Newton steps the time step is increased by $\mathrm{d}t_{\mathrm{scale}}^{\mathrm{up}} = 1.5$.

The Newton solver uses a maximum of 15 line search iterations. The relative accuracy is $\epsilon_{\mathrm{nl}} = 1 \cdot 10^{-6}$, and the absolute error is $\epsilon_{\mathrm{nl}}^{\mathrm{abs}} = 0$. During phase appearance the modified Newton solver uses $\epsilon = 1.5$ for the correction of the capillary pressure (confer Section 5.5).

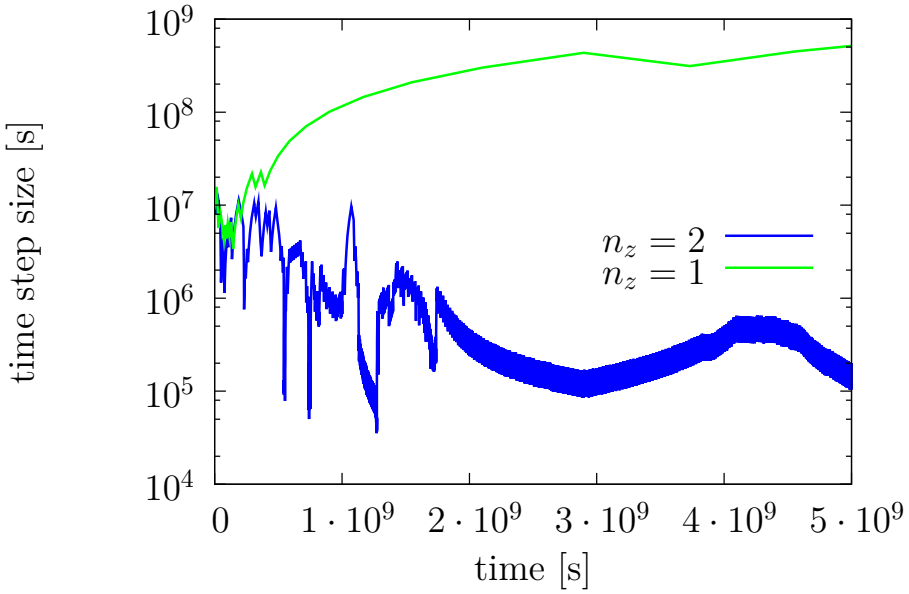


Figure 9.6: Time step size evolution for 2D benchmark

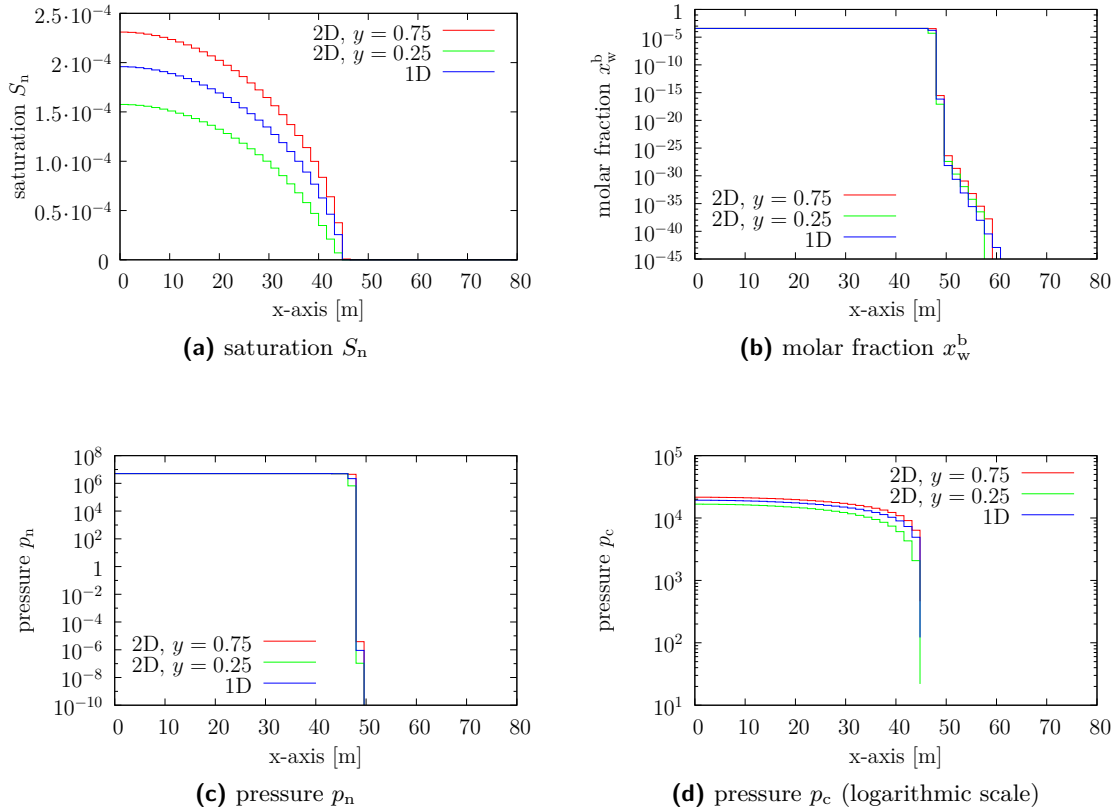
Two simulations are performed. The only difference is the number of cells in z-direction, the 1D simulation uses $n_z = 1$ and the 2D simulation $n_z = 2$. Figure 9.7 displays the distribution of several variables at $4.55 \cdot 10^9$ s (approximately 144 years). The red and green lines display the distribution of the variables for the upper and lower cells of the 2D simulation. The blue line shows the distribution for the 1D simulation. Only a part of the whole simulation domain is shown, because the nonwetting phase only fills about half of the domain. The same amount of nonwetting component is injected for both simulations, so the values for the nonwetting saturation for the 1D simulation lie between the saturations for the upper and lower cells of the 2D simulation.

Figure 9.6 shows a comparison of the time step evolution of the two simulations. For the 2D simulation (using $n_z = 2$), the time step size is more or less constant during the injection (until approximately $3.2 \cdot 10^8$ s). After injection the time step size breaks down.

Using only one cell in z-direction ($n_z = 1$), the time step size behaves as expected. During the injection, the time step size is similar to that of the 2D simulation. After that the time step size increases as expected, because no new nonwetting component is injected. The distribution of CO_2 and water only changes slowly by, e. g., diffusion and dissolution processes.

Results of other groups for this proposed 2D benchmark are highly appreciated. It is an important question, if other approaches for the choice of primary variables (confer Chapter 4) produce better results.

In this chapter, the Svalbard benchmark is used for a simulation with a large number of cells. Numerical simulations with different grid resolutions show that the cell size impacts



(a) saturation S_n

(b) molar fraction x_w^b

(c) pressure p_n

(d) pressure p_c (logarithmic scale)

Figure 9.7: Distribution of important variables at $4.55 \cdot 10^9$ s for 2D benchmark. Only the first 80 m of the domain are shown.

the CO_2 plume distribution. Furthermore, a weak scalability test is performed. After injection stop a distinct decrease of the time step size is observed. A simple 2D benchmark is proposed, which aims to evaluate the time step size evolution during and after injection.

10 Summary and Outlook

The numerical simulation of carbon capture and storage (CCS) poses a challenge for present-days models and computers. During CCS, CO_2 is injected into the subsurface and moves upward until it reaches a structural barrier. Different trapping mechanisms (e. g., residual trapping and solubility trapping) have to be considered for the numerical simulation.

Accurate constitutive relations for water and CO_2 are crucial for meaningful simulation results. The relations for the physical properties, e. g., densities and solubilities are typically developed by fitting a curve to experimentally gathered data. Interpolation tables are successfully used to save computation time for the evaluation of the constitutive relations.

The mathematical model that describes CCS and incorporates solubility trapping is two-phase two-component flow in porous media. This is a system of two coupled partial differential equations with suitable initial and boundary conditions. It describes the interaction between the two phases water and CO_2. Two primary variables have to be chosen for this system of two equations, all other secondary parameters are derived from the primary variables.

One of the challenges in simulating compositional two-phase flow in contrast to immiscible two-phase flow is the fact that there are regions where only one of the phases is present. A standard choice for the set of primary variables is, for example, one phase pressure and the saturation of the other phase. However, this choice is only valid if both phases are present. For two-phase two-component flow a different approach has to be taken.

As an important result in this work, a set of persistent primary variables is developed. Capillary pressure and nonwetting phase pressure are chosen as primary variables. This formulation can be consistently used in the presence or absence of the nonwetting phase. Numerical test show, that this set of primary variables can be successfully used for the challenging simulation of phase appearance in two-phase two-component flow.

All simulations in this work are performed within the DUNE simulation framework. Simulations show that an adapted Newton solver developed in the context of this thesis significantly improves the convergence behavior. Algebraic Multigrid is used as a preconditioner to get an efficient and fast parallel linear solver.

Different test cases show the capability of the implemented code. First, the simple 1D MoMas benchmark is implemented. The results are compared to those of other groups and are in good agreement. A grid convergence study demonstrates, that the solution converges with an optimal order of one during grid refinement.

Using a 2D and 3D domain, CO_2 injection is simulated by employing the constitutive relations for water and CO_2. The spreading of the CO_2 plume in the simulations behaves as expected. Parallel simulations are one possibility for a speed up of simulations of large domain. A strong and weak scalability test demonstrate successfully the parallel capabilities of the simulation.

As last example the 3D Svalbard benchmark is treated. A special challenge of this benchmark is the very large and anisotropic domain. In this setup, CO_2 is only injected for a certain time span. After the injection period, the time step size breaks down contrary to expectations. To investigate this behavior further, a new 2D benchmark similar to the MoMas benchmark is proposed.

The simulation of CCS poses a big challenge, mainly due to the long time spans and the large domains that have to be regarded. Nevertheless, the grids used for the simulation need to be fine enough, to be able to track the position of the CO_2 plume accurately. A good parallel performance of the code, as demonstrated in this work is the key to successful simulations in reasonable time.

A problem in this context is, that in general the time step size for the simulation decreases for smaller grid sizes. Because more cells have to undergo the transition from containing one phase to two phases, the Newton solver does not converge if the time step is too large. Hence, the simulation of long time spans is still computationally expensive.

In this work a set of persistent primary variables is used for all simulations. Using persistent variables is only on possibility to deal with the problem of disappearing phases for compositional two-phase flow. An important question is, how other approaches perform in comparison to the persistent variable approach.

Different benchmarks are regarded in the course of this work. The MoMas benchmark is a simple 1D setup that concentrates on phase appearance for a two-phase two-component flow model. The results of the participating groups are compared and mostly in good correspondence. Nevertheless, a detailed comparison between the numerical advantages and disadvantages and the overall performance of the different models is still missing.

To this end, a new 1D and 2D benchmark are presented in this work. They use the MoMas benchmark with a simple physical setup as a basis. Hopefully, these propositions will inspire groups to contribute with performance results for their approaches.

Appendix A

Compositional Multiphase Benchmark

This benchmark proposition was developed in cooperation with Peter Bastian and Olaf Ippisch.

We consider a generic system consisting of two phases $\alpha \in \{\mathrm{w}, \mathrm{n}\}$ and two components $\kappa \in \{\mathrm{a}, \mathrm{b}\}$. The system is written in terms of molar densities and should be able to accommodate the MoMas test case as well as a CO_2 test case later. For the numerical solution a choice of primary variables needs to be made and a set of reduced equations needs to be derived. Each participant should state this derivation starting from the system given in this section.

The quantities used below are listed in table A.1.

Conservation of each component κ in each phase α:

$$\partial_t(\phi \nu_\alpha^\kappa S_\alpha) + \nabla \cdot (\nu_\alpha^\kappa u_\alpha + j_\alpha^\kappa) = q_\alpha^\kappa + r_\alpha^\kappa. \tag{A.1}$$

Table A.1: Quantities in the generic tow-phase two-component model.

Symbol	Meaning	Unit
ϕ	porosity	-
ν_α^κ	molar density (concentration) of κ in α	$\mathrm{mol\,m^{-3}}$
S_α	saturation of phase α	-
p_α	pressure of phase α	Pa
q_α^κ	source/sink term of κ in α	$\mathrm{mol\,m^{-3}\,s^{-1}}$
r_α^κ	phase exchange of κ in α	$\mathrm{mol\,m^{-3}\,s^{-1}}$
u_α	velocity of phase α	$\mathrm{m\,s^{-1}}$
$k_{r\alpha}$	relative permeability function	-
μ_α	dynamic viscosity of phase α	$\mathrm{Pa\,s^{-1}}$
K	absolute permeability	$\mathrm{m^2}$
ρ_α	mass density of phase α	$\mathrm{kg\,m^{-3}}$
g	$g = (0, 0, -9.81)^T$ gravitational acceleration vector	$\mathrm{m\,s^{-2}}$
j_α^κ	diffusive flux of κ in α	$\mathrm{mol\,m^{-2}\,s^{-1}}$
D_α^κ	effective diffusion coefficient of κ in α	$\mathrm{m^2\,s^{-1}}$
$p_\mathrm{c}(S_\mathrm{w})$	capillary pressure saturation relationship	Pa

Extended Darcy's law:

$$u_\alpha = -\frac{k_{r\alpha}(S_\alpha)}{\mu_\alpha} K \left(\nabla p_\alpha - \rho_\alpha g\right). \tag{A.2}$$

(Note that the source/sink term includes phase exchange).

Diffusion of component κ in phase α:

$$j_\alpha^\kappa = -D_\alpha^\kappa(S_\alpha)\nabla \nu_\alpha^\kappa. \tag{A.3}$$

We assume in addition the condition

$$\forall \alpha : \sum_\kappa j_\alpha^\kappa = 0. \tag{A.4}$$

Finally, we have the capillary pressure saturation relationship

$$p_\mathrm{n} - p_\mathrm{w} = p_\mathrm{c}(S_\mathrm{w}) \tag{A.5}$$

and

$$S_\mathrm{w} + S_\mathrm{n} = 1. \tag{A.6}$$

MoMas Benchmark Test Case 1

This test case has been published in [64, 14]. Parameters have been adapted from the mass-based formulation and are not changed with the exception of the van Genuchten α_VG parameter. A minor difference to the original MoMas Benchmark is a small change in the definition of the water phase density.

Completion of the Model

In this application w is the liquid phase, n is the gas phase, w is the component water and b is the component hydrogen. A first assumption for this system is

$$\nu_\mathrm{n}^\mathrm{a} = 0. \tag{A.7}$$

Molar and mass densities of the gas phase (component) are related to gas phase pressure via the ideal gas law:

$$\nu_\mathrm{n}^\mathrm{b} = \frac{p_\mathrm{n}}{RT}, \qquad \rho_\mathrm{n}^\mathrm{b} = \frac{p_\mathrm{n} M^\mathrm{b}}{RT} \tag{A.8}$$

The water phase is assumed to be an ideal solution with constant molar concentration but varying mass density:

$$\nu_\mathrm{w}^\mathrm{a} + \nu_\mathrm{w}^\mathrm{b} =: \nu_\mathrm{w} = \frac{\rho^\mathrm{a}}{M^\mathrm{a}}, \qquad \rho_\mathrm{w} = \nu_\mathrm{w}^\mathrm{a} M^\mathrm{a} + \nu_\mathrm{w}^\mathrm{b} M^\mathrm{b} \tag{A.9}$$

where M^{a}, M^{b} are the molar masses of water and hydrogen in $\mathrm{kg\,mol^{-1}}$ and ρ^{a} is the mass density of pure water.

Equilibrium phase exchange according to Henry's law is assumed :

$$k_H p_{\mathrm{n}} = \chi^{\mathrm{b}}_{\mathrm{w}} = \frac{\nu^{\mathrm{b}}_{\mathrm{w}}}{\nu^{\mathrm{a}}_{\mathrm{w}} + \nu^{\mathrm{b}}_{\mathrm{w}}} = \frac{\nu^{\mathrm{b}}_{\mathrm{w}}}{\nu^{\mathrm{a}}} \quad \Rightarrow \quad \nu^{\mathrm{b}}_{\mathrm{w}} = \nu^{\mathrm{a}} k_H p_{\mathrm{n}} = H p_{\mathrm{n}} \tag{A.10}$$

where we used (A.9).

For the effective diffusion coefficient of hydrogen (b) in the water phase (w) we take the Ansatz [42]:

$$D^{\mathrm{b}}_{\mathrm{w}}(S_\alpha) = \phi^{4/3} S^2_{\mathrm{w}} D^{\mathrm{b}}_{\mathrm{mol},w} \tag{A.11}$$

Then, according to (A.4) we have $j^{\mathrm{a}}_{\mathrm{w}} = -j^{\mathrm{b}}_{\mathrm{w}}$ and since the gas phase consists only of hydrogen there is no diffusion of hydrogen in the gas phase.

Finally, capillary pressure and relative permeabilities are given by the van Genuchten model:

$$p_c(S_{\mathrm{w}}) = \frac{1}{\alpha_{\mathrm{VG}}} \left(\bar{S}_{\mathrm{w}}^{-1/m} - 1 \right)^{1/n} \tag{A.12}$$

$$k_{rw}(S_{\mathrm{w}}) = \sqrt{\bar{S}_{\mathrm{w}}} \left(1 - \left(1 - \bar{S}_{\mathrm{w}}^{1/m} \right)^m \right)^2 \tag{A.13}$$

$$k_{rn}(S_{\mathrm{n}}) = \sqrt{1 - \bar{S}_{\mathrm{w}}} \left(1 - \bar{S}_{\mathrm{w}}^{1/m} \right)^{2m} \tag{A.14}$$

where

$$\bar{S}_{\mathrm{w}} = \frac{S_{\mathrm{w}} - S_{\mathrm{w,res}}}{1 - S_{\mathrm{w,res}}}, \qquad m = \frac{1}{n}. \tag{A.15}$$

In the first test case the van Genuchten Parameter α_{VG} will be varied. We will choose:

Parameter	Value	Unit	Comment
α_1	$5 \cdot 10^{-7}$	$\mathrm{Pa^{-1}}$	*original value (easy)*
α_2	$5 \cdot 10^{-4}$	$\mathrm{Pa^{-1}}$	*new value (difficult)*

Domain, Boundary and Initial Conditions

The test case is one-dimensional in space with $\Omega = (0, 200\mathrm{m})$ and the time interval to simulate is $\Sigma = (0, 10^6\mathrm{y})$.

There are no sources and sinks inside the domain, i.e.

$$q^\kappa_\alpha = 0. \tag{A.16}$$

Initial conditions are completely water-saturated:

$$p_{\mathrm{w}}(0, x) = p_0, \quad S_{\mathrm{w}}(x, 0) = 1, \quad \nu^{\mathrm{b}}_{\mathrm{w}} = 0 \tag{A.17}$$

Table A.2: Parameter values for the MoMas test case 1.

Parameter	Value	Unit
ρ^{a}	$1 \cdot 10^{3}$	$\mathrm{kg\,m^{-3}}$
M^{a}	$1 \cdot 10^{-2}$	$\mathrm{kg\,mol^{-1}}$
M^{b}	$2 \cdot 10^{-3}$	$\mathrm{kg\,mol^{-1}}$
H	$7.65 \cdot 10^{-6}$	$\mathrm{mol\,m^{-3}\,Pa^{-1}}$
μ_{n}	$9 \cdot 10^{-6}$	$\mathrm{Pa\,s^{-1}}$
μ_{w}	$1 \cdot 10^{-3}$	$\mathrm{Pa\,s^{-1}}$
$D^{\mathrm{b}}_{\mathrm{mol},w}$	$3 \cdot 10^{-9}$	$\mathrm{m^2\,s^{-1}}$
R	8.3144621	$\mathrm{J\,mol^{-1}\,K^{-1}}$
T	303	K
ϕ	0.15	-
$S_{\mathrm{w,res}}$	0.4	-
K	$5 \cdot 10^{-20}$	$\mathrm{m^2}$
n	1.49	-

Table A.3: Initial and boundary condition values for the MoMas test case 1. Note that the time unit is years in this table!

Parameter	Value	Unit
p_0	10^{6}	Pa
T_{inj}	$5 \cdot 10^{5}$	y
q_0	$2.785 \cdot 10^{-3}$	$\mathrm{mol\,m^{-2}\,y^{-1}}$

with pressure p_0 given in table A.3.

Denoting by $w^{\mathrm{a}} = \nu^{\mathrm{a}}_{\mathrm{w}} u_{\mathrm{w}} + j^{\mathrm{a}}_{\mathrm{w}}$ and $w^{\mathrm{b}} = \nu^{\mathrm{b}}_{\mathrm{n}} u_{\mathrm{n}} + \nu^{\mathrm{b}}_{\mathrm{w}} u_{\mathrm{w}} + j^{\mathrm{b}}_{\mathrm{w}}$ the total fluxes of components a (water) and b (hydrogen) we specify the boundary conditions at the left boundary

$$w^{\mathrm{a}}(0,t) = 0, \qquad w^{\mathrm{b}}(0,t) = \begin{cases} q_0 & 0 \le t < T_{\mathrm{inj}} \\ 0 & \text{else} \end{cases} \tag{A.18}$$

and at the right boundary

$$p_{\mathrm{w}}(200,t) = p_0, \qquad S_{\mathrm{w}}(200,0) = 1, \qquad \nu^{\mathrm{b}}_{\mathrm{w}} = 0. \tag{A.19}$$

The values q_0, T_{inj} and p_0 are given in Table A.3.

Appendix B

Implementation of the Constitutive Relations for CO_2 and Water

A short overview over the existing classes and functions for the constitutive relations between CO_2 and water is given here. The implementation is part of the dune-pm module. The class `PhysicalChemistry` is a collection of constants and functions for the computation of physical properties. Only mass units are used instead of molar units. The corresponding molar units can then be computed via the molar masses of CO_2 and water ($M^{\mathrm{a}} = 1.8 \cdot 10^{-2}\,\mathrm{kg\,mol^{-1}}$ and $M^{\mathrm{b}} = 4.4 \cdot 10^{-2}\,\mathrm{kg\,mol^{-1}}$). Table B.1 lists the units for all parameters. The template parameter `T` is the data type used for all computations, in this work `double` is used for all computations.

The two functions `SolubilityWaterInCO2` and `SolubilityCO2InWater` take temperature T, nonwetting phase pressure p_{w} and salinity s_{sal} as arguments and return the mass fractions $X^{\mathrm{b}}_{\mathrm{w}}$ and $X^{\mathrm{a}}_{\mathrm{n}}$ respectively.

```
template<typename T>
T PhysicalChemistry<T>::
  SolubilityWaterInCO2(T temperature, T pressure, T salinity)

template<typename T>
T PhysicalChemistry<T>::
SolubilityCO2InWater(T temperature, T pressure, T salinity)
```

The functions `DensityCO2` and `ViscosityCO2` take temperature T and nonwetting phase pressure p_{w} as arguments and return the mass density $\rho_{\mathrm{mass,n}}$ and viscosity ν_{n}.

parameter	symbol	unit
temperature	T	K
pressure	p	Pa
salinity	s_{sal}	$\mathrm{mol\,kg^{-1}}$
density	ρ_{mass}	$\mathrm{kg\,m^{-3}}$
viscosity	μ	$\mathrm{Pa\,s^{-1}}$
mass fraction	X	$\mathrm{kg\,kg^{-1}}$

Table B.1: Units used for the implementation of constitutive relations between water and CO_2

```
template<typename T>
T PhysicalChemistry<T>::DensityCO2(T temperature, T pressure)

template<typename T>
T PhysicalChemistry<T>::ViscosityCO2(T temperature, T pressure)
```

The function `DensityBrine` returns the mass density $\rho_{\text{mass,w}}$ and takes temperature T, wetting phase pressure p_{w}, salinity s_{sal} and mass fraction X^{b}_{w} as arguments.

```
template<typename T>
T PhysicalChemistry<T>::
  DensityBrine(T temperature, T pressure, T salinity, T massfraction)
```

For the computation of the CO_2 parameters, the class `InterpolateCO2` should be used, because the evaluation of the functions above is expensive (confer Section 5.7). The constructor of the class needs values for temperature and salinity, that are constant in all simulations in this work.

```
template<typename T>
class InterpolateCO2
{ public:
  InterpolateCO2(T temperature, T salinity) };
```

To initialize the interpolation table, the member function `Init` has to be used. The default value for parameter `numPoints` is 10000. For all underlying functions `numPoints` equally distributed values are stored. Afterwards a specific value is approximated with linear interpolation. Table B.2 shows the important member functions of the class. All functions take the pressure p_{n} as argument.

```
void Init(int numPoints = 10000)
```

The CO_2 interpolation class can then be used in the code like this:

```
#include<dune/pm/physics/physical_chemistry.hh>
...
//co2 values interpolation
  Dune::PM::InterpolateCO2<T>* co2values =
    new Dune::PM::InterpolateCO2<T>( temperature, salinity);
  co2values->Init();
  T viscosity = co2values->viscosity( pressure);
```

Member function	Description
`Xlg (pressure)`	returns X^{b}_{w}
`Xgw (pressure)`	returns x^{b}_{w}
`xgw (pressure)`	returns x^{a}_{n}
`density (pressure)`	returns $\rho_{\text{mass,n}}$
`viscosity (pressure)`	returns ν_{n}

Table B.2: Member functions of class `InterpolateCO2`

List of Acronyms

CCS	Carbon capture and storage
EOS	Equation of state
EOC	Experimental order of convergence
TS	Time steps
ADT	Average time step size
MDT	Minimum time step size
ANI	Average number of Newton iterations
ALI	Average number of linear iterations per Newton iteration
ALT	Average number of linear iterations per time step
TIT	Average time for a linear iteration
REV	Representative elementary volume
AMG	Algebraic multigrid
DNAPL	Dense non-aqueous phase liquid

List of Symbols

α	phase α	
α_{VG}	parameter for van Genuchten-Mualem model	$[\mathrm{Pa}^{-1}]$
$\bar{S}_\alpha$	effective saturation of phase α	
ϵ_{lin}	relative accuracy for linear solver in Newton step	
$\epsilon_{\mathrm{nl}}^{\mathrm{abs}}$	absolute error for Newton solver	
ϵ_{nl}	relative accuracy for Newton solver	
κ	component κ	
λ	parameter for Brooks-Corey model	
Γ_{in}	influx boundary	
Γ_{out}	outflow boundary	
$\mathrm{max}_{\mathrm{linesearch}}$	maximum allowed number of linesearch steps for Newton solver	
$\mathrm{max}_{\mathrm{newtoniterations}}$	maximum allowed number of Newton steps	
μ_α	viscosity of phase α	[Pa/s]
b	nonwetting component	
ν_α^κ	molar density concentration of component κ in phase α	$[\mathrm{mol/m}^3]$
n	nonwetting phase	
Ω	simulation domain	
ϕ	porosity	
ρ^{a}	standard water mass density $\rho^{\mathrm{a}} = 1000$	$[\mathrm{kg/m}^3]$
ρ_α	molar density of phase α	$[\mathrm{mol/m}^3]$
$\rho_{\mathrm{mass},\alpha}$	mass density of phase α	$[\mathrm{kg/m}^3]$
τ	parameter for Brooks-Corey model	
a	wetting component	
w	wetting phase	
D_α^κ	molecular diffusion coefficient of component κ in phase α	$[\mathrm{m}^2/\mathrm{s}]$
$D_{\mathrm{pm},\alpha}^\kappa$	diffusion coefficient of component κ in phase α in porous medium	$[\mathrm{m}^2/\mathrm{s}]$
g	gravity vector $g = (0, 0, -9.81)$	$[\mathrm{m/s}^2]$
H	Henry's Law parameter	$[\mathrm{mol/(m}^3\,\mathrm{Pa})]$
j_α^κ	diffusive flux of component κ in phase α	$[\mathrm{mol\,s/m}^2]$
K	absolute permeability	$[\mathrm{m}^2]$
$k_{\mathrm{r}\alpha}$	relative permeability of phase α	

m	parameter for van Genuchten-Mualem model	
M^κ	molar mass of component κ	[kg/mol]
n	parameter for van Genuchten-Mualem model	
p_α^i	initial pressure of phase α	[Pa]
p_w, p_n	wetting and nonwetting phase pressure	[Pa]
p_c	capillary pressure	[Pa]
p_entry	entry pressure	[Pa]
q^κ	source/sink term for component κ	[kg/(m^2 s)]
q_α^κ	source/sink term for component κ in phase α	[kg/(m^2 s)]
r_α^κ	phase exchange of componentκ in phase α	[mol/(m^3 s)]
S_α	saturation of phase α	
$S_{\alpha,\mathrm{res}}$	residual saturation of phase α	
s_sal	salinity of water	
T	temperature	[K]
u_α	velocity of phase α	[m/s]
w_in^κ	flux of component κ at Γ_in	[kg/(m^2 s)]
X_α^κ	mass fraction of component κ in phase α	
x_α^κ	molar fraction of component κ in phase α	
$\mathrm{d}t_\mathrm{initial}$	initial time step size	
$\mathrm{d}t_\mathrm{max}$	maximum time step size	
$\mathrm{d}t_\mathrm{scale}^\mathrm{down}$	scaling factor for time step after failed time step	
$\mathrm{d}t_\mathrm{scale}^\mathrm{up}$	scaling factor for time step after successful time step	

Bibliography

[1] Abadpour, A. and Panfilov, M. Method of Negative Saturations for Modeling Two-phase Compositional Flow with Oversaturated Zones. *Transport in Porous Media*, 79(2):197–214, 2009.

[2] Amaziane, B., Jurak, M., and Keko, A. Modeling Compositional Compressible Two-Phase Flow in Porous Media by the Concept of the Global Pressure. *Computational Geosciences*, 18(3–4):297–309, 2014.

[3] Angelini, O., Chavant, C., Chénier, E., Eymard, R., and Granet, S. Finite volume approximation of a diffusion-dissolution model and application to nuclear waste storage. *Mathematics and Computers in Simulation*, 81:2001–2017, 2011.

[4] Atkins, P. W. *Physikalische Chemie*. VHC Verlagsgesellschaft, 1990.

[5] Bastian, P. *Numerical Computation of Multiphase Flow in Porous Media*. Ph.D. thesis, University of Kiel, 1999.

[6] Bastian, P., Birken, K., Johannsen, K., Lang, S., Neuss, N., Rentz-Reichert, H., and Wieners, C. UG – a flexible Software toolbox for solving partial differential equations. *Computing and Visualization in Science*, 1:27–40, 2014.

[7] Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöfkorn, R., Kornhuber, R., Ohlberger, M., and Sander, O. A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part II: Implementation and Tests in DUNE. *Computing*, 82(2–3):121–138, 2008.

[8] Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöfkorn, R., Ohlberger, M., and Sander, O. A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part I: Abstract Framework. *Computing*, 82(2–3):103–119, 2008.

[9] Bastian, P., Heimann, F., and Marnach, S. Generic implementation of finite element methods in the Distributed and Unified Numerics Environment (DUNE). *Kybernetika*, 46(2):294–315, 2010.

[10] Bear, J. *Dynamics of Fluids in Porous Media*. Elsevier, 1972.

[11] Bielinski, A. *Numerical Simulation of CO2 Sequestration in Geological Formations*. Ph.D. thesis, University of Stuttgart, 2007.

[12] Blatt, M. *A Parallel Algebraic Multigrid Method for Elliptic Problems with Highly Discontinuous Coefficients*. Ph.D. thesis, University of Heidelberg, 2010.

[13] Blatt, M. and Bastian, P. On the Generic Parallelisation of Iterative Solvers for the Finite Element Method. *Int. J. Computational Science and Engineering*, 4(1):56–69, 2008.

[14] Bourgeat, A., Granet, S., and Smaï, F. Compositional Two-Phase Flow in Saturated - Unsaturated Porous Media: Benchmarks for Phase Appearance/Disappearance. *Radon Series on Computational and Applied Mathematics : Simulation of Flow in Porous Media*, pages 81–106, 2013.

[15] Bourgeat, A., Jurak, M., and Smaï, F. Two phase partially miscible flow and transport modeling in porous media ; application to gas migration in a nuclear waste repository. *Computational Geosciences*, 13(1):29–42, 2009.

[16] Bourgeat, A., Jurak, M., and Smaï, F. Modelling and Numerical Simulation of Gas Migration in a Nuclear Waste Repository. *URL: http://arxiv.org/abs/1006.2914*, 2010.

[17] Braess, D. *Finite Elemente*. Springer-Verlag, 1992.

[18] Brooks, R. and Corey, A. Hydraulic Properties of Porous Media. *Colorado State University Hydrology Paper*, 3, 1964.

[19] Burchardt, A. *Effektive Mischung durch dichtegetriebene Strömung*. Ph.D. thesis, University of Heidelberg, 2013.

[20] Butler, M., Barnes, L., Sarma, D. D., and Gelinas, B. Bulldozer: An Approach to Multithreaded Compute Performance. *IEEE Micro*, 31(2):6–15, 2011.

[21] Class, H. *Theorie und numerische Modellierung nichtisothermer Mehrphasenprozesse in NAPL-kontaminierten porösen Medien*. Ph.D. thesis, University of Stuttgart, 2000.

[22] Class, H., Ebigo, A., Helmig, R., Dahle, H. K., J.Nordbotten, Celia, M., Audigane, P., Darcis, M., Ennis-King, J., Fan, Y., et al. A benchmark study on problems related to CO2 storage in geologic formations. *Computational Geosciences*, 13(4):409–434, 2009.

[23] Class, H., Helmig, R., and Bastian, P. Numerical simulation of non-isothermal multiphase multicomponent processes in porous media. *Advances in Water Resources*, 25(5):533–550, 2002.

[24] Dahle, H., Eigestad, G., Nordbotten, J., and Pruess, K. A model-oriented benchmark problem for CO2 storage. *Princeton-Bergen Series on Carbon Storage*, 2009.

[25] Dedner, A., Klöfkorn, R., and Nolte, M. The DUNE-ALUGrid Module. *Mathematical Software*, 2014.

[26] Duan, Z., Moller, N., and Weare, J. H. An equation of state for the CH4-CO2-H2O system: I. Pure systems from 0 to 1000°C and 0 to 8000 bar. *Geochimica et Cosmochimicha Acta*, 56(7):2605–2617, 1992.

[27] Duan, Z. and Sun, R. An improved model calculating CO2 solubility in pure water and aqueous NaCl solutions from 273 to 533 K and from 0 to 2000 bar. *Chemical Geology*, 193:257–271, 2003.

[28] Fenghour, A., Wakeham, W. A., and Vesovic, V. The Viscosity of Carbon Dioxide. *Journal of Physical and Chemical Reference Data*, 27(1):31–44, 1998.

[29] Forsyth, P. A. and Simpson, R. B. A two-phase two-component model for natural convection in a porous medium. *International Journal for Numerical Methods in Fluids*, 12:655–682, 1991.

[30] García, J. E. Density of Aqueous Solutions of CO2. *Lawrence Berkeley National Laboratory*, LBNL-49023, 2001.

[31] Gasda, S. E., Nordbotten, J. M., and Celia, M. A. Vertically-averaged approaches to CO2 injection with solubility trapping. *Water Resources Research*, 47(5), 2011.

[32] van Genuchten, M. T. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal*, 44:892–898, 1980.

[33] Gharbia, I. B. and Jaffré, J. Gas phase appearance and disappearance as a problem with complementarity constraints. *Mathematics & Computers in simulation*, 2013. Accepted.

[34] Hackbusch, W. and Reusken, A. Analysis of a damped nonlinear multilevel method. *Numerische Mathematik*, 55(2):225–246, 1989.

[35] Hassanizadeh, S. and Gray, W. Toward an improved description of the physics of two-phase flow. *Advances in Water Resources*, 16:53–67, 1993.

[36] Hassanizadeh, S. M. and Gray, W. G. General Conservation Equations for Multiphase Systems: 3. Constitutive Theory for Porous Media Flow. *Advances in Water Resources*, 3:25–40, 1980.

[37] Heimsund, B. *Mathematical and Numerical Methods for Reservoir Fluid Flow Simulation.* Ph.D. thesis, University of Bergen, 2005.

[38] IEA Greenhouse Gas R&D Programme (IEA GHG). 1st CO2 Geological Storage Modelling Network Meeting, 2009.

[39] Ippisch, O. *Coupled Transport in Natural Porous Media.* Ph.D. thesis, University of Heidelberg, 2003.

[40] Ippisch, O. *Contributions to the large-scale Simulation of Flow and Transport in Heterogeneous Porous Media* . Ph.D. thesis, University of Heidelberg, 2014.

[41] Jaffré, J. and Sboui, A. Henry's Law and Gas Phase Disappearance. *Transport in Porous Media*, 82:521–526, 2010.

[42] Jin, Y. and Jury, W. A. Characterizing the Dependence of Gas Diffusion Coefficient on Soil Properties. *Soil Science Society of America Journal*, 60:66–71, 1996.

[43] Lauser, A., Hager, C., Helmig, R., and Wohlmuth, B. A new approach for phase transitions in miscible multi-phase flow in porous media. *Water Resources*, 34:957–966, 2011.

[44] Marchand, E. and Knabner, P. Results of the MoMas benchmark for gas phase appereance and disappearance using generalized MHFE. *Advances in Water Resources*, 73:74–96, 2014.

[45] Marchand, E., Müller, T., and Knabner, P. Fully coupled generalized hybrid-mixed finite element approximation of two-phase two-component flow in porous media. Part II: numerical scheme and numerical results. *Computational Geosciences*, 16:691–708, 2012.

[46] Marchand, E., Müller, T., and Knabner, P. Fully coupled generalized hybrid-mixed finite element approximation of two-phase two-component flow in porous media. Part I: formulation and properties of the mathematical model. *Computational Geosciences*, 17:431–442, 2013.

[47] Martens, S., Kempka, T., Liebscher, A., Lüth, S., Möller, F., Myrttinen, A., Norden, B., Schmidt-Hattenberger, C., Zimmer, M., and Kühn, M. Europe's longest-operating on-shore CO2 storage site at Ketzin, Germany: a progress report after three years of injection. *Environmental Earth Sciences*, 2012.

[48] Metz, B., Davidson, O., de Coninck, H., Loos, M., and Meyer, L. IPCC Special Report on Carbon Dioxide Capture and Storage. *Cambridge University Press*, 2005.

[49] Mikyška, J. and Firoozabadi, A. A New Thermodynamic Function for Phase-Splitting at Constant Temperature, Moles and Volume. *AIChE Journal*, 57(7):1897–1904, 2011.

[50] Mikyška, J. and Firoozabadi, A. Investigation of mixture stability at given volume, temperature, and number of moles. *Fluid Phase Equilibria*, 321:1–9, 2012.

[51] Neumann, R., Bastian, P., and Ippisch, O. Modeling and Simulation of Two-Phase Two-Component Flow with Disappearing Nonwetting Phase. *Computational Geosciences*, 17(1):139–149, 2013. doi:10.1007/s10596-012-9321-3.

[52] Nordbotten, J., Flemisch, B., Gasda, S., Nilsen, H., Fan, Y., Pickup, G., Wiese, B., Celia, M., Dahle, H., Eigestad, G., et al. Uncertainties in practical simulation of CO2 storage. *International Journal of Greenhouse Gas Control*, 9:234–242, 2012.

[53] O.Eiken, Ringrose, P., Hermanrud, C., Nazarian, B., Torp, T. A., and Høier, L. Lessons learned from 14 years of {CCS} operations: Sleipner, In Salah and Snøhvit. *Energy Procedia*, 4:5541 – 5548, 2011.

[54] Peaceman, D. W. *Fundamentals of Numerical Reservoir Simulation*. Elsevier, 1977.

[55] Peng, D. Y. and Robinson, D. B. A New Two-Constant Equation of State. *Industrial and Engineering Chemistry Fundamentals*, 15(1):59–64, 1976.

[56] Polívka, O. and Mikyška, J. Compositional Modeling of Two-Phase Flow in Porous Media Using Semi-Implicit Scheme. In preparation.

[57] Polívka, O. and Mikyška, J. Compositional modeling in porous media using constant volume flash and flux computation without the need for phase identification. *Journal of Computational Physics*, 272:149–169, 2014.

[58] Spycher, N. and Pruess, K. CO2-H2O Mixtures in the Geological Sequestration of CO2. II. Partitioning in Chloride Brines at $12-100$°C and up to 600 bar. *Geochimica et Cosmochimicha Acta*, 69(13):3309–3320, 2005.

[59] Spycher, N., Pruess, K., and Ennis-King, J. CO2-H2O Mixtures in the Geological Sequestration of CO2. I. Assessment and calculation of mutual solubilities from 12 to 100°C and up to 600 bar. *Geochimica et Cosmochimicha Acta*, 67(16):3015–3031, 2003.

[60] Walter, L. *Uncertainty Studies and Risk Assessment for CO2 Storage in Geological Formations*. Ph.D. thesis, University of Stuttgart, 2013.

[61] DUNE Distributed and Unified Numerics Environment.
URL: `http://www.dune-project.org`, 2011.

[62] dune-pdelab Howto.
URL: `http://www.dune-project.org/pdelab/pdelab-howto-2.0.0.pdf`, 2014.

[63] Hontomin CCS Pilot Storage Site.
URL: `htpp://http://www.zeroco2.no/hontomin`.

[64] MoMaS Benchmark on Multiphase Flow in Porous Media - Exercise 1.
URL: `http://www.gdrmomas.org/Benchmark/multi phase/multiphasique.html`.