Simplified models for numerical simulation of geological CO2 storage

Odd Andersen

Thesis for the degree of philosophiae doctor (PhD)
at the University of Bergen

2017

Date of defence: February 3rd

List of included publications

I Odd A. Andersen, Sarah E. Gasda and Halvor M. Nilsen, *Vertically Averaged Equations with Variable Density for CO_2 Flow in Porous Media*, Transport in Porous Media **107**, 2015. doi:10.1007/s11242-014-0427-z

II Halvor M. Nilsen, Knut-Andreas Lie, Olav Møyner and Odd A. Andersen, *Spill-Point Analysis and Structural Trapping Capacity in Saline Aquifers Using MRST-co2lab*, Computers & Geosciences, **75**, 2015. doi:10.1016/j.cageo.2014.11.002

III Halvor M. Nilsen, Knut-Andreas Lie and Odd A. Andersen, *Robust Simulation of Sharp-Interface Models for Fast Estimation of CO_2 Trapping Capacity in Large-Scale Aquifer Systems*, Computational Geosciences, **20**, 2016. doi:10.1007/s10596-015-9549-9

IV Halvor M. Nilsen, Knut-Andreas Lie and Odd A. Andersen, *Fully-Implicit Simulation of Vertical-Equilibrium Models with Hysteresis and Capillary Fringe*, Computational Geosciences, **20**, 2016. doi:10.1007/s10596-015-9547-y

V Halvor M. Nilsen, Knut-Andreas Lie and Odd A. Andersen, *Analysis of CO_2 Trapping Capacities and Long-Term Migration for Geological Formations in the Norwegian North Sea using MRST-co2lab*, Computers & Geosciences, **79**, 2015. doi:10.1016/j.cageo.2015.03.001

VI Knut-Andreas Lie, Halvor M. Nilsen, Odd A. Andersen and Olav Møyner, *A Simulation Workflow for Large-Scale CO_2 Storage in the Norwegian North Sea*, Computational Geosciences, **20**, 2016. doi:10.1007/s10596-015-9487-6

VII Odd A. Andersen, Halvor M. Nilsen and Knut-Andreas Lie, *Reexamining CO_2 Storage Capacity and Utilization of the Utsira Formation*, ECMOR XIV - 14th European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8-11 September 2014. doi:10.3997/2214-4609.20141809

VIII Odd A. Andersen, Knut-Andreas Lie and Halvor M. Nilsen, *An Open-Source Toolchain for Simulation and Optimization of Aquifer-Wide CO_2 Storage*, Energy Procedia, **86**, 2016. doi:10.1016/j.egypro.2016.01.033

IX Odd A. Andersen, Halvor M. Nilsen and Sarah E. Gasda, *Modelling Geomechanical Impact of CO_2 Injection Using Precomputed Response Functions*, ECMOR XV - 15th European Conference on the Mathematics of Oil Recovery, Amsterdam, Netherlands, 29 August - 1 September 2016. doi:10.3997/2214-4609.201601760

X Odd A. Andersen, Halvor M. Nilsen and Sarah E. Gasda, *Vertical equilibrium flow models with fully coupled geomechanics for CO_2 storage modeling, using precomputed mechanical response functions*, submitted

Abstract

Carbon capture and storage (CCS) is a proposed strategy to reduce global emissions of greenhouse-gases. The basic principle is to capture CO_2 from power generation or other industrial activities and inject it into deep geological formations for permanent storage. CCS is considered practically indispensable by the Intergovernmental Panel on Climate Change (IPCC) in order to reach internationally agreed climate targets. It can be understood as a bridge technology intended to limit emissions from fossil-fuel based economic activities while working toward the longer-term goal of a sustainable energy system.

Since the purpose of CCS is to permanently prevent large quantities of CO_2 from entering the carbon cycle, the practical storage capacity and long-term safety of candidate storage sites are important questions to address. As for other industrial subsurface operations, numerical simulations based on mathematical models of the involved physics play a key role in helping us understand the processes taking place underground. However, existing industrial simulators for 3D subsurface multi-phase flow are typically developed for the support of hydrocarbon production. Such simulators are by nature limited in their ability to handle problems at the very large spatial and temporal scales that must be taken into account when investigating CO_2 storage issues. To properly address the full range of questions related to CO_2 injection and migration, a variety of mathematical models of different complexity is needed, ranging from detailed multi-physics models describing local conditions around the injection sites, to simplified or mathematically upscaled descriptions capable of modeling developments at much larger spatial scales and timeframes.

This thesis addresses the development, analysis and efficient implementation of simplified mathematical models specially designed to address questions related to long-term CO_2 storage. One class of such models enables computationally efficient long-term simulations based on the assumption of vertical equilibrium (VE). Under this assumption, vertical flow in the storage formation is neglected, which allows for reducing the dimensions of the governing equations and corresponding simulation domain from three to two, while still preserving important 3D effects. A different numerical approach is based on analysis of storage site geometry, and provides a near-instant way of predicting long-term migration pattern and assessing trapping capacity. Together, these methods can be applied as parts of larger workflows set up to address more complex questions related to CO_2 storage.

The work presented in this thesis contributes to the field of mathematical modeling of CO_2 storage with several new developments, including:

- mathematical derivation and inclusion of additional physical effects into the VE modeling framework (compressibility, geomechanics, hysteresis models), and as-

sessment of the impacts;

- efficient algorithms for spill-point analysis of the storage site caprock, and evaluating their applicability for CO_2 migration prediction, trapping capacity estimation and well placement support, based on testing on real/realistic datasets;
- robust implementation of vertical-equilibrium models in a fully-implicit, black-oil framework, combining most of the modeling capabilities that have been published for these types of models (dissolution, residual trapping, capillary pressure, caprock rugosity, and dynamic fluid properties derived from equations of state), followed by testing and validation based on real aquifer models from the North Sea;
- use of the aforementioned approaches in combination with gradient-based nonlinear optimization methods to identify practical injection scenarios that maximize stored CO_2 while minimizing migration out of the target formation.

In the spirit of promoting reproducible computational research [94], most of the computer code underlying the results presented has been made freely available as open software in the form of a separate module, `MRST-co2lab`, to the MATLAB Reservoir Simulation Toolbox (`MRST`), developed, maintained and published by the Computational Geosciences group of SINTEF ICT, Department of Applied Mathematics. The exception is the work on geomechanics, which has not yet been made part of the public code, but is in the pipeline for a future release.

The introduction to this thesis is written as a tutorial that introduces some of the basic theory underlying the papers, and also demonstrates the practical use of the software by gradually outlining a full code example based on a publicly available dataset of the Johansen formation.

Acknowledgements

My PhD position was part of the MatMoRA-II project, funded by the Norwegian Research Council and Statoil ASA. I have been associated with the University of Bergen as a PhD student while at the same time kept my employment as a researcher at the SINTEF ICT, Department of Mathematics in Oslo.

During the thesis work, I have had the fortune of to be surrounded with many supportive people whose assistance, knowledge and encouragement have greatly facilitated my progress. I would like to take the opportunity to thank some of them here.

I would like to start by thanking my formal PhD advisers. At SINTEF, my adviser was Dr. Halvor M. Nilsen, with whom I have collaborated daily during my PhD period. I quickly came to appreciate his insight and deep physical intuition during our many discussions on research matters large and small, as well as his natural ability to draw on the competencies of people around him. He has been an invaluable resource both as an adviser, co-author and colleague. At the University of Bergen, my closest adviser was Dr. Sarah E. Gasda. Despite the geographical distance between Oslo and Bergen, she put in a lot of effort into getting me up to speed from the beginning, and has been a regular point of contact ever since. Thanks to her enthusiasm and knowledge, it was always an inspiration and pleasure to meet and discuss with her, and each of my trips to Bergen were time well spent. Prof. Helge Dahle was my first primary PhD adviser and one of the main architects of the MatMoRA-II project. Without him, the opportunity offered by this PhD position might not have come to pass. Prof. Dahle eventually moved on to become Dean of the Department of Mathematics, and his adviser role was seamlessly picked up by Prof. Inga Berre. I have really appreciated both her positive and friendly attitude, her responsiveness and her professionality in all dealings with me and my PhD work.

I would like to particularly thank my co-author and leader of the Computational Geosciences group at SINTEF ICT, Department of Applied Mathematics, Prof. Knut-Andreas Lie. As the experienced leader of the group, Prof. Lie has offered a large amount of valuable advice and encouragement during my thesis work, both in terms of the scientific work itself, the writing process, and in other aspects of working in the research group. Despite his many responsibilities and tasks, it was never hard to reach him and he always found time to discuss when asked. Together with Prof. Dahle, Prof. Lie was instrumental in making the MatMoRa-II project a reality.

Another co-author (and fellow PhD student) at SINTEF is Olav Møyner, to whom I also wish to express my deep appreciation. Over the years, Olav has proved to be a reliable and supportive colleague who has been a particular resource person for issues related to the inner workings of `MRST`, especially regarding the fully-implicit solvers framework, whose current design and implementation is to a significant extent attributable to him.

More generally, I would like to extend my thanks to all the other members of the Computational Geosciences research group. I have asked and received advice from every single member of the group at some point during the course of my PhD work. As a researcher, is hard to overemphasize the benefit of being surrounded by a team of people that are both deeply knowledgeable, encouraging and willing to help. I consider it a real privilege.

Last but not least I would like to thank my family. I am deeply grateful to my wife Yumi for her continuing support and patience. If my PhD work sometimes required of me a certain level of sacrifice, her support (both in the moral and very practical sense) certainly helped me carry it through. A special thanks goes to my parents-in-law, Hideo and Hiroko Hanashima, for their substantial assistance during the summer of 2016 when most of this dissertation came together. During my family's stay in Tokyo, they took care of all practical details, providing me with weeks of uninterrupted work time that really helped getting the writing done. Finally, I would like to thank my two children Helena and Valdemar, who constantly remind me how life is so much more than a PhD.

Contents

Part I

Background

Chapter 1

Introduction

1.1 CO_2 storage and the need for simplified models

By geological storage of CO_2 we refer to the practice of injecting CO_2 gas captured from industrial activities into deep geological formations for the purpose of permanent sequestration. The motivation behind this concept is to limit further release and accumulation of CO_2 in the atmosphere, which is widely acknowledged as the primary contributor to anthropogenic climate change [56].

Background

The idea of storing CO_2 in the subsurface as an effort to reduce emissions started to circulate in research literature in the early nineties [53, 60, 107]. Only a few years later, in 1996, the first commercial large-scale operation became reality when the Norwegian petroleum company Statoil started to inject CO_2 into the Utsira Sand formation below the Norwegian continental shelf [61]. The injected CO_2 was a by-product from on-site natural gas processing from the nearby Sleipner Vest field. The operation was chosen as an economically superior option to paying emission taxes, and is still ongoing today. It has attracted considerable interest from researchers in the field, both for its pioneer nature and its long duration of operation.

However, 20 years later, according to the Global CCS Institute[1] only three ongoing industrial-scale geological CO_2 storage projects are in operation worldwide: the Norwegian Snøhvit project [68], the Quest project in Canada [3], and the ongoing operations at Sleipner/Utsira. Combined, these projects prevent about 2.7 megatonnes (Mt) of CO_2 annually from being released to the atmosphere, a figure which remains largely insignificant compared to the approximately 32 thousand Mt CO_2 annually resulting from worldwide combustion of fossil fuels [1]. Even if we include large-scale industrial projects where CO_2 is injected into the ground primarily to facilitate oil production[2], the world capacity for injecting and storing CO_2 in the subsurface is currently limited to approximately 28 Mt a year [2].

[1] The Global CCS Institute online Projects Database can be accessed at www.globalccsinstitute.com/projects/large-scale-ccs-projects

[2] the practice of CO_2 injection for enhanced oil recovery is historically much older than the idea of geological CO_2 storage for reducing emissions

It is clear that in order to play a meaningful role for mitigating climate change, carbon capture and storage (CCS) needs to be scaled up several orders of magnitude in the future. Keeping future global warming within a limit of 2°C has become an internationally agreed goal. In the latest 2-degree scenario proposed by the International Energy Agency (IEA), the role of CCS would represent 12 percent of cumulative future emission reductions [4]. Likewise, according to the Intergovernmental Panel on Climate Change (IPCC), most future scenarios that could meet the 2-degree target rely on widespread use of CCS; the absence of which would more than double the cost of reaching the target [55]. Using the 12 percent figure from the IEA scenario and multiplying by annual emissions, 3.8 gigatonnes (Gt) of CO_2 would have to be permanently stored each year through CCS, a figure that is roughly comparable to annual world production of conventional crude oil [31].

There are a few comments to be made here. First, in the IPCC and IEA scenarios above, the long-time mitigation goal remains a transition into a sustainable energy system. The use of CCS can thus be seen as a *bridge technology*, a temporary technological solution that will buy us the time required to make this necessary transition. In addition, the IPCC mitigation scenarios where the 2-degree target is likely to be met typically involve temporary emission overshoots followed by widespread use of Carbon Dioxide Removal (CDR) technologies that produce negative emissions. In particular, this involves afforestation as well as widespread use of *bioenergy with CCS* (BECCS). As such, CCS as a technology will likely remain relevant and important even in the (unlikely) case that fossil fuels could be completely phased out. Another important point to mention is that the use of CO_2 storage as a climate mitigation strategy fundamentally requires us to think in terms of very large scales. Not only do we need to store enormous volumes of CO_2 underground, but we also need to ensure that they remain safely in place for thousands of years.

The need for simplified models

The scientific and technological issues related to geological CO_2 storage are not fundamentally different from well-known principles and solutions used every day in other subsurface engineering activities such as petroleum production, liquid waste disposal, natural gas storage and geothermal energy. The underlying equations describing multiphase flow processes in porous rock are well known and extensively studied, and the technical solutions behind activities such as subsurface exploration, fluid injection and monitoring exist commercially and are being extensively used. Indeed, it is widely agreed that the main factors currently holding back large-scale deployment of CCS are not scientific or technological in nature, but primarily consist of political, economical or public perception issues [22, 106].

Even if the technology exists to get started on larger-scale CCS deployment, that does not mean that the technological challenges are solved. In particular, the upscaling of CO_2 storage operations to levels described above raises significant questions that can only be addressed through a combination of practical experience and computer modeling. As current practical experience is limited to CO_2 injections with an annual rate of no more than 1 Mt, the use of computer modeling will play a fundamental role for addressing questions related to the necessary upscaling of CCS operations. For a given storage formation, these questions are of three fundamental types: (1) how

much CO_2 can the formation store (*storage capacity*); (2) how efficiently can CO_2 be injected into the formation (*injectivity*); and (3) can we ensure the CO_2 will remain in place (*containment*).

The commercially available reservoir simulators today are highly sophisticated and have been developed over many years for the purpose of facilitating the operations of the petroleum industry[24, 96]. These simulators are able to model subsurface flow in 3D involving multiple fluid phases and their evolving compositions, while accounting for a wide range of mechanical, chemical and thermal effects. However, they remain poorly equipped to address the large physical and temporal scales associated with CO_2 injection and long-time migration. Whereas a typical petroleum extraction scenario relates to a subsurface reservoir that may be up to a few kilometers in lateral extent and an extraction operation that may last up to a few decades, a CO_2 storage scenario involves migration processes that may last for millennia, and potentially extend over hundreds of kilometers [52]. On the other hand, smaller scales cannot be ignored. As described in the following section, CO_2 tends to migrate in the subsurface in the form of long, very thin plumes, whose thickness is ultimately measured in centimeters. In order to represent the spatial domain, reservoir simulators employ grids where each discrete grid cell has its own set of properties and unknown variables. The number of grid cells required to represent such thin plumes in three dimensions across the large distances involved quickly renders the simulation problem computationally intractable to conventional simulators [82]. The same challenge is faced by 3D simulators developed within universities and research institutes, such as [39, 83, 89, 111].

The computational problem is compounded by a general lack of information describing the storage formation in terms of shape, rock properties and rock-fluid interactions. Available information will typically be sparse and come with significant uncertainties. This is generally true for conventional reservoir simulation as well, but the situation is exacerbated by the large distances involved in CO_2 storage modeling. Moreover, as there has so far been much less economic interest in CO_2 storage than in petroleum extraction, geological formations suitable for large-scale CO_2 storage have generally been subject to less detailed surveying and mapping.

For computer simulations to provide real value in the face of data scarcity and large uncertainty, it will often be necessary to investigate a wide range of possibilities, running simulations with different plausible assumptions on the input data. In the aggregate, the simulated outcomes can thus provide insight into possibilities and likelihoods of future developments, and provide bounds on expected behavior. As new monitoring data becomes available during operation, the update of reservoir models to account for actually observed behavior (history matching) will typically also require a number of simulations to be run in the search of a best fit.

In short, the computational requirements for simulation-based investigation of large-scale CO_2 storage scenarios quickly become prohibitive using conventional 3D simulators. Instead, what is needed are methods that reduce the complexity of the computational problem by introducing simplifying assumptions, removing processes with limited impact on the final results, and that are built on a computational framework that is well suited to handle the different balances of forces associated with injection and migration processes, as presented in the upcoming section. Such methods can provide practical means for rapid exploration of possible long-term outcomes, assessing impact of parameter variations, optimizing injection scenarios and continuously updating sim-

ulation models as new data becomes available. Over recent years, the modeling of CO_2 storage has thus developed into a separate field of research, combining and adapting elements from both reservoir modeling and primary migration studies of hydrocarbons in basin modeling. It is in this context that the work of this dissertation has been carried out.

1.2 CO_2 storage - a qualitative description

This section presents a very simplified overview of the main processes taking place and the factors to consider as CO_2 is injected and stored in a geological formation. The focus is on the qualitative, physical description; the corresponding mathematical description is presented in Chapter 2.

The storage formation

Sedimentary rock is formed through long-time deposition of sediments (sand, crushed sea shells, precipitates, etc.) followed by hardening through lithification processes. At the microscopic level, sedimentary rock consists of individual rock grains with void spaces in between, referred to as *pore space*. Under natural conditions, this pore space is initially filled with fluid. For a saline aquifer, this fluid is *brine*, i.e., water with dissolved minerals. For a petroleum reservoir, the fluid can also be some combination of hydrocarbons. The degree to which individual pores are connected determines how easily fluids can flow through the rock. This is a quantifiable property of the rock called *permeability*, and the higher its value, the less resistance is presented to fluid flow within rock.

By forcing out and replacing resident fluid, sedimentary rock can be used to store CO_2. For this to be practical, a geological formation needs to consist of rock with sufficiently high permeability to allow fluid replacement to take place, and sufficiently high pore volume to provide enough space for storage. Many types of sandstone are good candidates in this regard.

A candidate formation must also be deep enough. At surface temperature and pressure, CO_2 is a gas with a density somewhat higher than air. On the other hand, at pressure and temperature conditions reached at depths of approximately 800 m or deeper [10], CO_2 takes the form of a dense phase, which allows considerably larger amounts of mass to be stored per given volume. A storage formation should thus be at least 800 m deep (and preferably deeper) to avoid storing CO_2 in gas phase, which would be vastly inefficient.

In any case, CO_2 will in virtually all relevant circumstances remain less dense than resident brine. The difference in density creates a buoyancy force that will drive CO_2 back up towards the surface if not kept in place by other means. It is therefore necessary that the storage formation is overlaid by a *caprock*, a layer of rock that hinders further upwards migration of CO_2. Fluid flow through the caprock is hindered by its very low permeability, but in addition a phenomenon called *capillary exclusion* can completely prevent CO_2 from entering caprock unless a significant overpressure, called *entry pressure*, is reached. This phenomenon is caused by capillary forces, and is further discussed in Chapter 2. A good caprock typically consists of a (preferably thick)

layer of shale, clay or mudstones.

The presence of a caprock is what has for millions of years prevented hydrocarbons in petroleum reservoirs from migrating to the surface. If we use a depleted petroleum reservoir to store CO_2, we can therefore be relatively confident in its sealing properties. For a saline aquifer, the quality of the caprock needs to be affirmed by extensive examination and evaluation prior to eventual use. In any case, on-going monitoring activities need to be in place both during injection operations and in the post-injection phase, to detect any unexpected development.

Finally a word about geometry. Sedimentary rocks tend to be deposited in a sequence of layers, or *strata*, each of which may have quite different rock properties. The target formation for CO_2 storage will therefore typically take the form of a large, thin sheet of sandstone or other permeable rock, overlain by one or more sheets of low-permeability caprock. The thickness of a storage formation can vary from a few meters to a few hundred meters, while a lateral extent that is generally much larger and can reach up to several hundred kilometers [51]. To illustrate this point, Figure 1.1 shows a simulation model of the Sleipner CO_2 storage site in the Norwegian North Sea, represented using the real aspect ratio. The thin, almost flat nature of the aquifer is clearly visible on the lower plot.

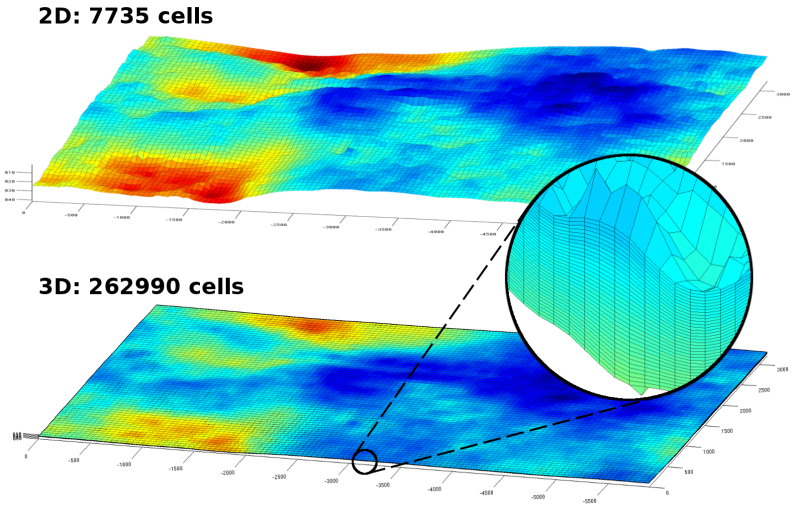


Figure 1.1: Simulation model of Layer 9 of the Sleipner CO_2 storage site located in the Utsira formation situated in the Norwegian North Sea. The lower plot shows the full 3D model, drawn using the correct aspect ratio, with a local zoom to make visible its 3D structure. Color represents depth. The upper plot is a 2D grid representing the top surface of the aquifer. Variation in depth has been exaggerated in order to visualize the topography.

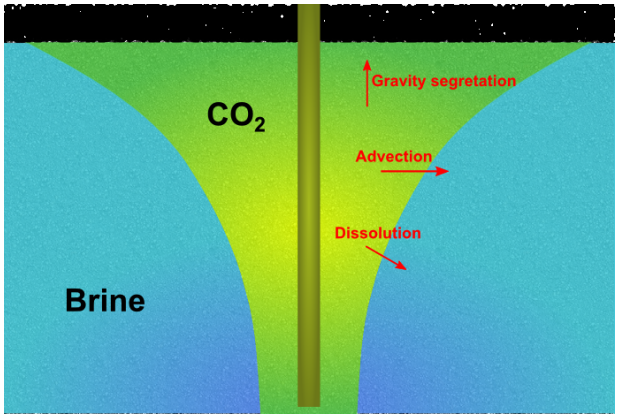


Figure 1.2: Illustration of CO_2 injection into an aquifer though a vertical well under a sealing caprock.

The injection process

We now consider the injection of CO_2 from a vertical well[3] into a saline aquifer overlain by an impermeable caprock.

Figure 1.2 presents a simplified drawing of a cross-section of the aquifer, centered on the wellbore. When injection starts, high pressure forces CO_2 out from the well and into the pore space of the surrounding rock, pushing away the brine in what is known as a *drainage* process. The system can be described in terms of two-phase flow, with distinct CO_2 and brine phases each occupying its own fraction of the local pore space. The fraction of pore space occupied by a specific phase in the immediate neighborhood of a given point is referred to as its phase *saturation*. Some limited exchange of mass occurs between the two phases, as a small amount of CO_2 will dissolve into the brine, and an even smaller amount of brine will evaporate into the CO_2, but the impact of this mixing remains small at this stage. Fluid flow is mainly advective, where the dominant driver of flow is the pressure gradient induced by the high pressure in the wellbore. However, gravity also plays a role in directing the flow. Since CO_2 is significantly less dense than brine, it will tend to flow upwards until further vertical movement is blocked by the overlying caprock. As a result, the shape of the region containing CO_2 will develop into that of an inverted cone, as illustrated in Figure 1.2. Over time, the upper part of this cone will spread outwards and become gradually thinner. (In reality, the shape might not always be as neat as illustrated, as the presence of rock heterogeneities or thin, low-permeability baffles may significantly complicate the picture (see e.g., [103]).

The invading CO_2 cannot completely drive out all water in the pore space. A residual amount of water will remain trapped in place, as a result of the water phase losing spatial continuity and individual droplets becoming surrounded with CO_2. The remaining amount of water is referred to as irreducible water saturation, and is determined by properties of the rock-fluid system. Near the well, the amount of trapped brine will fur-

[3]Note that a CO_2 injection well would not necessarily be vertical. A horizontal well along the bottom of the formation would often be a more beneficial configuration.

ther decrease over time by gradually evaporating into the constant supply of injected, dry CO_2. As a result, salt or other minerals previously dissolved in the brine may precipitate as a solid matter within the pores, which can under some conditions clog them up and significantly reduce injectivity [71].

There are two main mechanisms in play to allow for the extra room needed to store the injected CO_2 in the formation. One is the gradual migration of brine, either across lateral boundaries or through the caprock, into neighboring formations. Although CO_2 is ideally prevented from entering the caprock through the mechanism of capillary exclusion, this does not apply to brine, which is able to pass through the low-permeable rock, albeit very slowly. The second mechanism is through *compressibility*. As pressure rises in the formation due to the injection process, the densities of CO_2 and brine slightly increase, thus allowing a larger amount of fluid to occupy the formation. Moreover, the elevated fluid pressure causes the rock to slightly expand, increasing the overall volume of available pore space. The expansion of rock is often considered to be directly proportional to local pressure increase. However, in the near-well area, large spatial and temporal variations in the pressure field may lead to more complex rock deformations such as arching effects. Moreover, the elevated pressure field causes changes in the global stress field of the rock matrix, which may ultimately lead to rock fracturing or fault reactivation (movement along formerly inactive faults) if pressure buildup is not controlled and kept within safe limits. Another risk associated with elevated pressures is the possibility that CO_2 pressure becomes high enough to overcome capillary exclusion, in which case CO_2 will enter the caprock. Taking pressure buildup into account is therefore an important concern for any CO_2 injection operation.

Finally, the effect of temperature should be mentioned. The temperature of injected CO_2 may differ significantly from the temperature of the formation [109]. Moreover, as pressure drops away from the borehole, CO_2 expands, which causes local cooling due to the Joule-Thomson effect. This has the potential to induce thermal stresses in the rock, but on the other hand the large heat capacity of the rock generally prevents dramatic temperature changes.

The migration process

After injection operation ceases, the well is shut off and the pressure gradient it induced gradually dissipates. As a result, advection quickly ceases to be a dominant driver of flow, and the fluid phases start to rearrange themselves in the aquifer, driven by gravitational and capillary forces. Over the course of the injection, the CO_2 gradually collects in a zone increasingly separate from the brine. This zone surrounds the well and extends outwards as a thin plume below the caprock. The sudden drop in pressure when injection ceases will cause the CO_2 in this zone to expand relative to the brine, inducing a secondary pressure effect that causes this zone to keep swelling for a while. The upward, buoyancy-driven movement of CO_2 also continues, driving the CO_2 plume to further thin and spread out. As a consequence, regions that were earlier saturated with CO_2, in particular around the well area, become filled with brine once again. In what is called an imbibition process, water re-invades the pore space as CO_2 evacuates. In a way similar to what happened to brine during the previous drainage process, a fraction of the CO_2 is left behind, completely trapped by capillary forces. This is referred to as *residual trapping*, and is one of four major trapping mechanisms identified in the

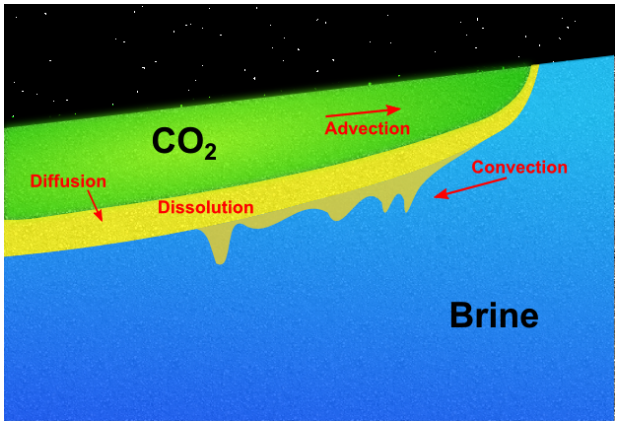


Figure 1.3: Gravity-driven migration of CO_2 below a sealing caprock.

context of geological CO_2 storage [62].

Once CO_2 reaches the top of the aquifer, the caprock prevents further direct upwards migration. The CO_2 plume is still laterally mobile, and will slowly migrate in the upslope direction below the caprock, as illustrated on Figure 1.3. The CO_2 is considered to be *hydrodynamically trapped* - it it still mobile but prevented from direct upward movement. During this gradual upslope migration, quantities of CO_2 may become permanently trapped as it collects and fills up in local pockets, anticlines, sealed fault blocks or other topographical features encountered along the way, in which case we talk about *structural trapping*. The same phenomenon can be caused by unconformities, pinch-outs or changes in rock type within the storage layer itself, in which case it is referred to as *stratigraphic trapping*. Structural and stratigraphic trapping represent a second major trapping mechanism in geological CO_2 storage.

Capillary forces are also at play. Capillary pressure refers to the difference in pressure between the two fluid phases. At equilibrium, there is a direct relationship between capillary pressure and fluid saturation, caused by differences in the adhesive forces between each of the liquids and the walls of the rock matrix, in combination with the surface tension at the CO_2-brine interface and the pore size distribution of the rock. As vertical equilibrium between gravitational and capillary forces is established, a transition zone will develop between the CO_2 plume on top and the brine below. In this transition zone, called the *capillary fringe*, there is a smooth transition in fluid saturation from mostly CO_2 (within the plume) to all brine (outside the plume). The fringe is indicated in yellow in Figure 1.3. Depending on the particular rock and other conditions in the aquifer, the capillary fringe can range from very thin to several meters thick, and can have a significant effect on the plume shape and migration speed [81].

So far, residual trapping and structural/stratigraphic trapping have been mentioned as two of four major trapping mechanisms. The remaining mechanisms are *dissolution trapping* and *mineral trapping*. It was mentioned above that small amounts of CO_2 would dissolve in brine wherever the two phases were in contact. During injection, the effect of this dissolution is of limited importance as it affects only the part of the phases that are in the immediate neighborhood of the CO_2-brine interface. In this neighborhood, each phase will quickly become saturated with the other, thus bring-

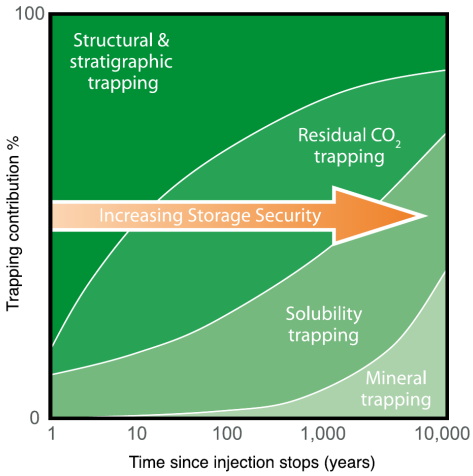


Figure 1.4: Relative importance of different trapping mechanisms over time. Diagram originally presented by IPCC [56].

ing the mixing process to a halt. The transport of dissolved CO_2 in brine away from the interface is driven by diffusion, which for this system is a very slow process that scales with the square-root of time. However, during the migration stage, dissolution becomes more important. The spreading-out and flattening of the plume and the gradual development of a capillary fringe significantly increases the amounts of brine and CO_2 that come in contact with each other. Moreover, CO_2-saturated brine is slightly denser than unsaturated brine. This will lead to gravitational instabilities within the vertical brine column, as CO_2-rich, 'heavy' brine below the CO_2 plume sits on top of a zone with lighter, unsaturated brine. Eventually, this may trigger the formation of convection cells within the brine phase, where the heavier saturated brine sinks downwards whereas the unsaturated brine rises upwards from below [38]. The onset of this process is highly dependent on rock permeability, porosity and other reservoir conditions, and may range from a few days to thousands of years [54]. As soon as this process starts, the transport of dissolved CO_2 away from the interface becomes a much more efficient process which can be approximately described by a constant rate. Thus, the dissolution of CO_2 into brine becomes linear in time and may potentially end up dominating all other trapping mechanisms, constantly eroding both the residually trapped CO_2 and CO_2 in structural traps, the capillary fringe and the moving plume itself.

The final trapping mechanism, mineral trapping, refers to geochemical reactions where dissolved CO_2 may precipitate as carbonate minerals, depending on the chemical composition of the pore water and rock as well as temperature and pressure in the storage aquifer [90]. This mechanism is often overlooked in the study of CO_2 storage, as the processes involved are generally assumed to be so slow to make this form of trapping relevant only on very long timescales [56, 112]. However, there have also been published studies suggesting that mineral trapping may be relevant on much shorter timescales [102, 113] in some situations.

Although the specifics may vary greatly between storage sites, a conceptual diagram showing how the role of different trapping mechanisms develop over time if presented

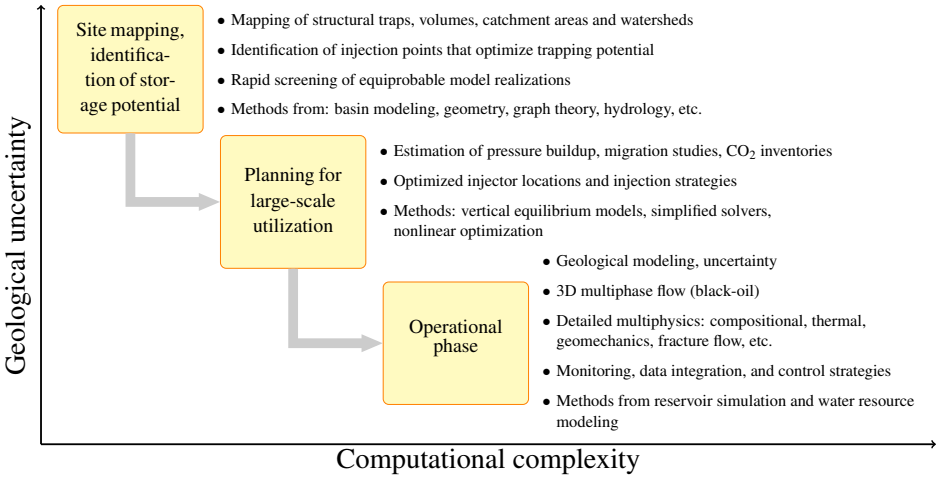


Figure 1.5: Different stages of planning and executing a CO_2 storage operation: related issues and modeling approaches.

in Figure 1.4. The diagram can be understood as relating to a 'typical' storage site. Through simulation it is possible to construct such diagrams for specific sites and storage scenarios. In this dissertation, such diagrams are referred to as *trapping inventories*.

1.3 Simplified modeling of CO_2 storage using `MRST-co2lab`

The presentation of background theory and simplified models in the upcoming chapters follows a practical approach where concepts and methods are explained and illustrated using concrete code examples. As argued in the related papers in Part II, part of the novelty of the models discussed in this thesis lies in their implementation within a robust and extendable software framework built on formulations and discretizations standard within the reservoir engineering community, and capable of working with industry-standard grids. In combining the presentation of background theory, models and software implementation, the emphasis is placed on the practical aspect of CO_2 storage modeling, as well as facilitating for the interested reader to understand and further explore the software tools provided.

Most of the models presented and discussed in this thesis have been implemented and made publicly available as open-source software package. This package is presented in the form of a separate module, called `MRST-co2lab`, of the MATLAB Reservoir Simulator Toolkit (`MRST`), developed and maintained by the Computational Geosciences group within the Department of Applied Mathematics at SINTEF ICT [72, 100]. While the work on `MRST-co2lab` is an ongoing group effort, the work and results presented in this thesis significantly contributed to its development into its current form.

The main focus of `MRST-co2lab` is on simplified models particularly useful for

studying long-time developments. There are two main groups of tools provided. The first group is related to spill-point analysis (Section 3.2), which computes the path followed upwards in the formation by a trickle of CO_2 injected at an infinitesimal rate. This analysis also includes identification of structural traps below the sealing caprock. The second group of tools consists of simulation models based on a depth-integrated (or vertically-averaged) physical description with the additional assumption of vertical equilibrium (Section 3.1).

A central point of this dissertation is that no single computational method is able to adequately address all aspects of modeling CO_2 storage, but that the modeling approach needs to be adapted to the level of analysis, the physical characteristics of the storage site under study, and the economic and regulatory conditions influencing particular usage scenarios. The availability of a range of computational tools that can work together on common datasets and interoperate with common algorithms allows the construction of flexible workflows, which makes it possible to tailor the study to the particular problem at hand. This point is further elaborated and illustrated in Section 3.4. In Figure 1.5, different tasks and methods relevant to CO_2 storage have been roughly sorted according to project planning stage. The stages are further lined up according to level of geological uncertainty and computational complexity. Current modeling capabilities of `MRST-co2lab` are mostly associated with the two stages preceding the operational phase, where the focus is on mapping, capacity estimation, large-scale planning and assessment of long-term migration. The issues related to the operational phase, on the other hand, are similar to those encountered in reservoir management for oil and gas production, and can to a large extent be covered by existing reservoir simulation tools for that purpose. Much of this functionality, including fully-implicit 3D multiphase black-oil type solvers, is also covered by other parts of `MRST`. A more comprehensive introduction to `MRST` and its functionality can be found in [64].

All the code examples presented in the following sections are completely spelled out, so that they can be directly entered and run from a MATLAB shell[4]. The only prerequisite is that `MRST` with `MRST-co2lab` is installed and the public datasets downloaded. A practical guide for how to set things up is provided in Appendix C.

[4]The example code is also available as scripts found in the `/examples/papers/DISSERTATION2017` folder within the `MRST-co2lab` module directory.

Chapter 2

A presentation of reservoir modeling

Before discussing simplified modeling methods, it is important to have a general understanding of the topic of numerical reservoir simulation, which is the purpose of the present chapter.

The following sections present the fundamental definitions, physical descriptions and mathematical equations and relationships used in the modeling of multiphase flow in a subsurface reservoir or aquifer. The focus is on reservoir modeling in general; simplified models are discussed in Chapter 3. Throughout this and the next chapter, concepts and ideas are illustrated by applying functionality from `MRST-co2lab` on a publicly available simulation dataset of a sector of the Johansen Formation [36]. This formation has been considered a candidate for large-scale CO_2 storage, and the dataset is one of the few publicly available 3D models of such sites.

Section 2.1 introduces the physical description of aquifer geometry, geology and the fluids involved. Moreover, the representation of aquifer state and the operation of wells is discussed.

Section 2.2 presents the mathematical framework, including the fundamental conservation equations, constitutive relations and equations of state, as well as how these can be solved numerically. The discussion leads up to a complete example of a CO_2 storage simulation of the Johansen Formation, based on `MRST`.

When presenting dimensional quantities in this section, M refers to mass, L to length and T to time.

2.1 The data model

2.1.1 Physical model of an aquifer or a reservoir

For the purposes of reservoir simulation, the physical simulation domain must be subdivided into a set of discrete, non-overlapping cells. This volumetric description is referred to as a *simulation grid*. Each cell in such a grid will be assigned its own distinct values for various properties and states needed in the simulation. There are many different types of simulation grids, depending on how the domain is subdivided into cells. These range from simple rectangular Cartesian grids, where all cells have identical shape and size, to completely unstructured grids with arbitrary topology and cell shapes. An important type of grid structure is the *corner-point grid*, where cells are

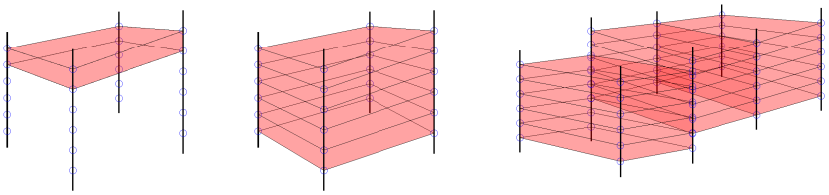


Figure 2.1: Structuring of cells in a corner-point grid. Pillars are drawn as thick black lines. Cell corners, constrained to lie on the pillars, are indicated with blue circles. *Left:* One cell with surrounding pillars. *Center:* A full vertical stack of cells. *Right:* Several neighboring stacks of cells. Cells of two adjacent stacks do not have to share the same corner points.

quasi-polyhedra[1] organized into vertical or inclined stacks, and whose corners are constrained to lie along a specified set of rectilinear "pillars" (for details, refer to [64]). This grid format has proved to be particularly convenient for describing the layered nature of sedimentary rock formations, and it can easily accommodate common features such as pinch-out or faults. It is therefore extensively used in practice and can be considered the predominant industry standard [96].

The aforementioned model of the Johansen Formation comprises a simulation grid provided on the corner-point format. A utility script provided with the bundled `MRST-co2lab` examples has been specially prepared to read and convert the most relevant part of this grid to the `MRST` grid format. Provided everything has been set up as described in Appendix C, the grid can be loaded by giving the following command to the MATLAB interpreter:

```
[G, rock, bcIx] = makeJohansenVEgrid;
```

The simulation grid is here represented by the structure `G`. To display its fields, we enter '`G`'and hit return:

```
G =

        nodes: [1x1 struct]
        faces: [1x1 struct]
        cells: [1x1 struct]
    cartDims: [100 100 11]
         type: {1x4 cell}
      griddim: 3
```

In order to accommodate the many different types of grid structures that can be encountered in practice, grids in `MRST` are represented using a general, unstructured format. As we see above, the grid structure contains a set of cells, a set of faces and a set of nodes. Each cell is defined as a volume bounded by a set of faces, and each face is defined as a surface bounded by a specified set of 3D nodes. A detailed description of the storage format can be found in [64], or by giving the command '`help`

[1] The term "quasi" is used here in recognition of the fact that cells in this format have top and bottom surfaces that may be bilinear in shape.

`grid_structure`' to the interpreter.

The grid can be visualized using `plotGrid`. The resulting graphic is shown in Figure 2.2.

```
plotGrid(G);                % plot the grid
view(-35,15); axis tight; % orient the grid for better viewing
```

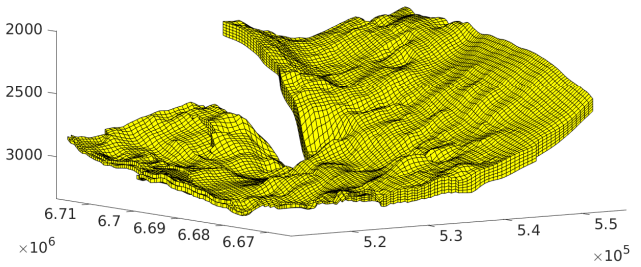


Figure 2.2: The Johansen simulation grid.

The full grid of the Johansen model contains additional layers of cells, but the script only extracts the parts that represent good-quality sandstone. The over- and underlying layers of low-permeability shale are stripped away. A predominant feature of this model is the large fault that cuts half way through the grid. We also note that the vertical resolution is limited to five cells at most. This resolution is less than ideal for CO_2 storage simulation, due to the spatial scale issues discussed in Section 1.1 (see also [36]). However, it will be adequate for our illustrative purposes.

The loading script also returned a set of petrophysical data, contained in the structure `rock`:

```
rock =

    perm: [29128x3 double]
    poro: [29128x1 double]
```

There are two fields in this structure: `perm` and `poro`, which specify porosity and permeability values for each cell in the grid. Porosity and permeability are macroscopic parameters that ultimately derive from the amount, shape and connectedness of the pore network between individual grains in the rock. As such, they cannot be specified pointwise in a strict sense, but must be defined by averaging over a small volume, called *representative elementary volume* (REV). A REV must be large enough to prevent spatial fluctuations in parameter values caused by the stochastic nature of individual rock pores, yet small enough not to interfere with the description of larger-scale structures (fractures, sedimentary layers, etc.) A detailed explanation can be found in [11].

Porosity is a dimensionless number defined as the ratio of accessible pore volume to bulk volume for a given REV. The porosity value attributed to a cell in a simulation grid represents the average porosity within that cell. It is an important parameter for reservoir simulation, as it specifies how much fluid can be contained within a given volume of rock. It is usually denoted by the symbol ϕ.

We can visualize the porosity field of the Johansen Formation by plotting each cell with a color value corresponding to its porosity as follows (result shown in Figure 2.3):

```
plotCellData(G, rock.poro); % plot grid with porosity values
view(−35,15); axis tight;   % orient the grid for better viewing
colorbar;                   % display a legend
```

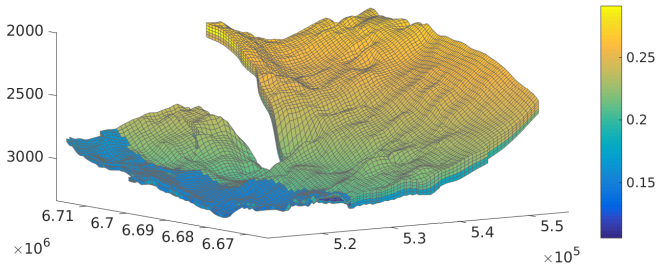


Figure 2.3: Porosity field for the Johansen simulation grid.

From this figure, we see that porosity values range from approximately 0.12 to 0.28. Since porosity in this case varies with spatial position, it is referred to as *heterogeneous*. On the other hand, a spatially constant porosity field would have been called *homogeneous*.

Permeability describes the conductivity of a porous medium to fluid flow. Sometimes it is referred to as *intrinsic permeability* to distinguish it from the related concept of *relative permeability*, which will be presented in the discussion of two-phase flow in Section 2.2.2. A porous medium has a well-defined permeability if a linear relation exists between fluid flow through the medium and the gradient of a pressure potential defined as fluid pressure less hydrostatic pressure. Permeability represents the corresponding proportionality constant, in a sense that will be made mathematically precise when Darcy's law is introduced in Section 2.2.1. Since both fluid flow and pressure gradient are vector quantities, permeability is in general a tensor quantity and represented by a square matrix. This matrix has to be symmetric and positive definite according to Onsager's principle [84]. When permeability can be described using only a scalar quantity, spatial direction plays no role and it is referred to as *isotropic*. If spatial direction matters and a matrix thus is needed, it is referred to as *anisotropic*. A permeability field can be homogeneous or heterogeneous, depending on whether it varies with spatial position. Permeability is measured in Darcy, and has the dimension of a squared length (L^2). It is typically denoted using k or $\mathbf{K}$. It is important to mention that since permeability describes a linear relation between pressure gradient and flow, one cannot apply its value over a REV to its value over an entire grid cell by simple spatial averaging. Instead, more advanced upscaling techniques are required.

In our example, we note that `rock.perm` contains three values per grid cell. These represent the diagonal elements of the corresponding permeability matrix (off-diagonal elements are here considered to be zero). The three diagonal elements express the permeability components along the three local principal axes of each grid cell. Let us visualize a lateral and a vertical component (in this simulation model, both lateral

components are equal):

```
plotCellData(G, rock.perm(:,1)); % permeability in x-direction
view(-35,15); axis tight;    % orient the grid for better viewing
colorbar;                    % display a legend
figure;                      % new figure window
plotCellData(G, rock.perm(:,3)); % permeability in z-direction
view(-35,15); axis tight;    % orient the grid for better viewing
colorbar;                    % display a legend
```

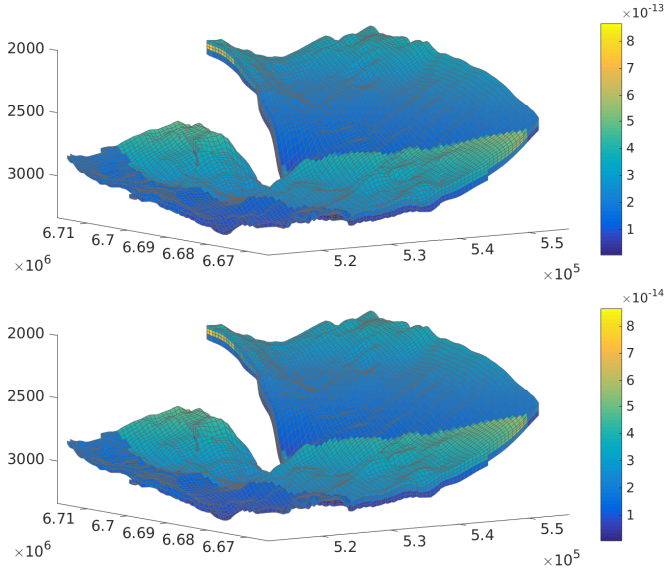


Figure 2.4: Lateral and vertical component of the permeability field for the Johansen simulation grid.

The result is shown in Figure 2.4, where we note that the vertical permeability component is much lower than the horizontal one. This is a typical feature of permeability fields in geological formations, caused by the layered structure of the rock.

2.1.2 Representation of evolving state

When using numerical models to simulate a physical system, it is assumed that there is a certain set of variables that are sufficient to completely describe the state of the modeled system at any given time. In other words, if the value of these variables are known at time t, all other aspects of the system at time t can be derived. These are here referred to as the *state variables* of a modeled system[2]. The term *independent variables* is also commonly used.

[2]In the field of thermodynamics, the term 'state variable' has a more narrow meaning and refers to thermodynamic state functions. However, we will here use it to refer to all variables specifying the state of our model, not just thermodynamic state functions.

The number and choice of state variables depends on the complexity of the modeled system. When modeling fluid flow in a reservoir, *fluid pressure* is almost always used as a state variable. When the system involves more than one fluid phase, *phase saturations* must typically also be included. In this context, a phase refers to a physically distinct fluid or gas. The saturation of a phase within a given reference volume simply means the fraction of the pore space occupied by that phase. Since it is natural to assume that all pore space is occupied by *some* phase (i.e., no vacuum exists), phase saturations will always sum to one. As a consequence, a system with N distinct phases only needs to keep explicit track of $N-1$ saturation values.

Since the complete state of the modeled system at a given time can be known from its state variables, the latter is usually referred to when talking about system *state*. The value of independent variables evolve over time, governed by a set of mathematical equations. In a numerical simulation, time is divided up into a number of discrete intervals numbered $t_0, t_1, ..., t_n$, where t_0 represents the initial state of the system, and the state at time t_i is completely determined from values of previous states. If the state at t_i cannot be completely determined from the state at t_{i-1}, but also requires information about earlier states, the system is referred to as *history dependent*, or that it exhibits *hysteretic* behavior. To avoid the need for explicitly involving multiple past timesteps when updating the state variables, it is often practical to include one or more *hysteresis variables* in the description of a state. Such variables are used to carry over sufficient information about past states to allow computing the updated values of the state variables. Hysteretic variables thus differ from other state variables in that they describe aspects about the system's past, rather than its present. The need to introduce hysteretic variables often arises when the macroscopic description of a system in terms of state variables is insufficient to describe relevant phenomena going on at smaller scales. For example, a saturation value describes the fraction of pore space occupied by a certain fluid phase, but does not describe its microscopic distribution within that pore space, which is the origin of phenomena such as *residual saturation* and *hysteretic relative permeability*, discussed in the next section.

For our Johansen example, we set up a simple, two-phase system, where brine and CO_2 represent the two separate, immiscible phases. Only two independent variables are needed, namely the pressure and saturation of brine. From these, CO_2 pressure and saturation can be immediately known using the constraint that saturations add up to one, and using a capillary pressure relation described in Section 2.1.3. In addition, we will include one hysteretic variable, which tracks the maximum CO_2 saturation over time at any given point in the aquifer. The state at each timestep is described in `MRST` as a structure array whose fields represent the different state variables. Before simulation starts, we need to describe the initial state of the system, so we create an initial state structure as follows:

```
gravity on;                    % tell MRST to turn on gravity
g = gravity();                 % get the gravity vector
rhow = 1000 kilogram/meter^3; % density of brine
initState.pressure = rhow * g(3) * G.cells.centroids(:,3);
initState.s = repmat([1, 0], G.cells.num, 1);
initState.sGmax = initState.s(:,2);
```

```
  initState =
```

```
          s: [29128x2 double]
   pressure: [29128x1 double]
      sGmax: [29128x1 double]
```

In the above listing, each cell in the simulation grid is assigned hydrostatic pressure. In other words, if z represents the depth below ground surface (assumed to be the water level), p_0 the surface pressure, ρ_w the density of brine and g the gravitational constant, then the pressure p is given by:

$$p(z) = p_0 + \int_0^z \rho_w g \mathrm{dz} \, .$$

In general, brine density varies with depth, making the integral above nontrivial to compute. However, in our simple model, a constant brine density, `rhow`, of 1000 kg/m^3 is assumed, so that pressure simply becomes: $p(z) = \rho_w g z$ (neglecting surface pressure). This is computed by the fourth line in the listing above, where the z value of each cell obtained from the third component of its centroid. As initial saturation, we set brine saturation to 1 and CO_2 saturation to 0 everywhere. (Strictly speaking it is only necessary to keep track of one of the saturations, but `MRST` stores both in its state objects). Finally, `sGmax` is the hysteretic variable which we set to zero for all cells.

Our Johansen example is about as simple as a two-phase system can be. More advanced models will need additional independent variables. For instance, a model that includes conservation of energy needs to have temperature or enthalpy among its independent variables. Likewise, for systems that include phase mixing or chemical reactions, additional independent variables will be needed to describe the evolving phase compositions.

2.1.3 Fluids and their properties

The next point to be discussed is the description of the fluids themselves. In the present context, a "fluid" refers to a physical substance that may be present in liquid, gas or supercritical form. For our example, the fluids will be brine and CO_2.

For the purpose of flow simulations, two very important fluid properties are *density* and *dynamic viscosity*. Density is defined as mass per volume (M/L^3). It plays a key role in the definition of the mass conservation equations presented in Section 2.2.1, as these equations require volume to be associated with fluid mass. Dynamic viscosity, from now on referred to simply as viscosity, is a measure of fluid resistance to shear deformation. It is also an important fluid property for simulation, since it represents the fluid-specific part of the linear relationship between pressure and fluid flow. In the International System of Units, viscosity is measured in Poiseuille (P), which has the dimension $M/(L \cdot T)$. In the context of reservoir simulation, a more commonly used unit is the Poise, which represents 0.1P.

Fluid density and viscosity are not fixed quantities. Density is a thermodynamic property, which means it can be tied to other thermodynamic properties such as pressure and temperature by an *equation of state*. Generally speaking, equations of state express mathematical relationships between thermodynamical quantities of a substance or mix of substances. They can be quite complex for models that take phase mixing or

chemical activity into account. However, for simpler systems such as ours, the practical consequence is that density will be described as a function of temperature and pressure, In reservoir software, equations of state are often modeled using cubic equations with fitted parameters [7], although highly precise models based on interpolation of a large number of experimental results also exist [101]. Viscosity is not a thermodynamic property, but referred to as a *transport property*, for which there also exist experimentally based fluid-specific correlations that associate it with thermodynamic properties.

In `MRST-co2lab`, fluid density and viscosity can be modeled as functions of pressure and temperature, based on densely sampled tables computed using the open-source CoolProp software library [12].

Simulation models that assume constant temperature are referred to as *isothermal*. For such models (which includes our Johansen example), density and viscosity can be modeled as functions of pressure only. In practice, a simpler model for pressure dependence is often employed, where constant *compressibility* is assumed. The compressibility c_f of a fluid is defined as:

$$c_f = \frac{1}{\rho}\frac{\mathrm{d}\rho}{\mathrm{d}p} \,. \tag{2.1}$$

By assuming that c_f can be approximated as constant over the relevant pressure range, starting from a reference pressure p_0 and a corresponding, known reference density ρ_0, the density as a function of pressure can be expressed as:

$$\rho(p) = \rho_0 e^{c_f(p-p_0)} \,. \tag{2.2}$$

Incidentally, a similar linear relationship is also often used to model changes in pore volume as a function of pressure. An increase in fluid pressure expands the rock matrix and the individual pores, thus increasing the storage volume available. Recalling that porosity is denoted by ϕ, one can write:

$$\phi(p) = \phi_0 e^{c_\phi(p-p_0)} \,, \tag{2.3}$$

where ϕ_0 corresponds to porosity at pressure p_0, and $c_\phi = \frac{1}{\phi}\mathrm{d}\phi/\mathrm{d}p$ is the *pore volume compressibility*.

We will now set up a structure containing the required fluid information for simulation of our Johansen example. We will model CO_2 density using relation (2.1). We keep brine density constant in order to remain consistent with our hydrostatic pressure field set up previously. Fluid viscosities will also be kept constant, for simplicity.

```
co2     = CO2props(); % load sampled tables of co2 fluid properties
p_ref   = 30 * mega * Pascal; % choose reference pressure
t_ref   = 94 + 273.15; % choose reference temperature, in Kelvin
rhoc    = co2.rho(p_ref, t_ref); % co2 density at ref. press/temp
cf_co2  = co2.rhoDP(p_ref, t_ref) / rhoc; % co2 compressibility
cf_wat  = 0; % brine compressibility (zero)
cf_rock = 4.35e-5 / barsa; % rock compressibility
muw     = 8e-4 * Pascal * second; % brine viscosity
muco2   = co2.mu(p_ref, t_ref) * Pascal * second; % co2 viscosity

mrstModule add ad-props; % load necessary module

% Use function 'initSimpleADIFluid' to make a simple fluid object
fluid = initSimpleADIFluid('phases', 'WG'           , ...
                           'mu'    , [muw, muco2]   , ...
                           'rho'   , [rhow, rhoc]   , ...
                           'pRef', p_ref            , ...
                           'c'     , [cf_wat, cf_co2] , ...
                           'cR'    , cf_rock        , ...
                           'n'     , [2 2]);
```

The above code snippet assumes `rhow` has already been defined from the script in Section 2.1.2. The first code line gives us access to sampled tables that provide us with fairly accurate, interpolated values for CO_2 density and viscosity. We define reference pressure and temperature on the following two lines, and then compute the corresponding values for CO_2 density (line 4) and the compressibility factor (line 5) as defined by (2.1). Then, we specify the brine and pore volume compressibilities[3], and brine and CO_2 viscosities (the latter interpolated from the sampled tables). We finally call the `MRST` function `initSimpleADIFluid` to create the fluid object, passing it all our specified information. The last parameter pair, "`'n', [2 2]`", is used in the definition of relative permeability curves, which will be explained in the discussion of multiphase flow in Section 2.2.2.

2.1.4 Wells and boundary conditions

Simply described, wells are thin, round holes that are drilled down into the subsurface to allow for the extraction or injection of fluids. Wells may be vertical, inclined or follow a more complex path that may extend horizontally along an aquifer for kilometers. In general, wells are characterized by extreme aspect ratios, with cross-section diameters measured in centimeters whereas most wells are at least several hundred meters long. The presence of an impermeable well casing and tubing prevents fluids from arbitrarily flowing in and out of the wellbore along its path. Fluid communication between the well and its surroundings is thus limited to perforations in the casing at chosen locations.

For reservoir simulations, a well model provides the information necessary to determine the flow of fluids in and out of the wellbore at each perforation, as well as the internal flows and pressures of the well. This includes in particular the location and flow properties of each perforation, the value of well control parameters, and (in

[3] By a quirk of `MRST` history, pore volume compressibility is contained within the fluid object, although it might well be considered a property of the rock instead.

the case of injection wells) the fluids being injected. Well operation can be controlled either by prescribing a specific pressure at the bottom bole, or by prescribing a total rate of fluid flow. Since these parameters are linked, they cannot both be prescribed at the same time, so it makes sense to distinguish between pressure-controlled and rate-controlled wells. Wells that inject fluids into the subsurface are referred to as *injectors*, whereas wells that produce fluids are called *producers*.

The Johansen model dataset suggests the location of one vertical injection well. To avoid going into detail, a pre-identified set of grid cells has been prepared, corresponding to the perforation locations of this well. We can visualize them using the following code, thus confirming that they indeed form a vertical stack:

```
% Cell indices for well perforations
wc = [3715, 10210, 16022, 21396, 26770];

% Construct a small grid only consisting of the well cells
wellCellGrid = extractSubgrid(G, wc);

% Plot wireframe of global grid
plotGrid(G, 'facecolor', 'none', 'edgealpha', 0.1);

% Plot grid representing well cells in red
plotGrid(wellCellGrid, 'facecolor', 'red');
```

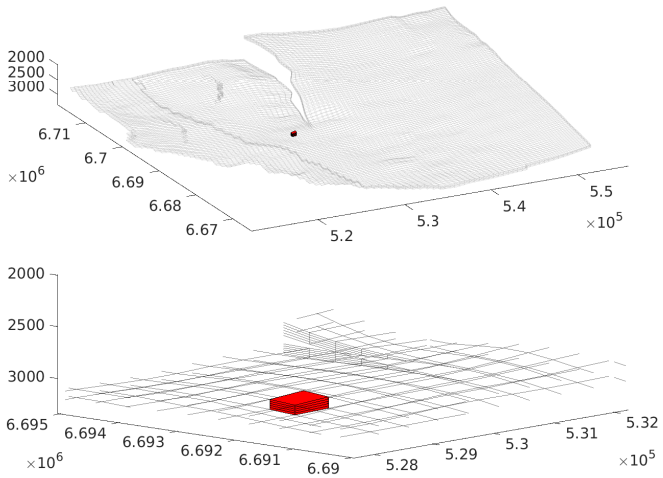


Figure 2.5: Grid cells perforated by vertical well in Johansen example. Top figure shows the global view; the bottom figure gives a zoom of the local region around the well.

The following code creates the data structure representing the well object in `MRST`.

```
% Calculate the injection rate
inj_rate = 3.5 * mega * 1e3 / year / fluid.rhoGS;

% Start with empty set of wells
W = [];

% Add a well to the set
W = addWell(W, G, rock, wc, ...
            'type', 'rate', ...  % inject at constant rate
            'val', inj_rate, ... % volumetric injection rate
            'comp_i', [0 1]);    % inject CO2, not water
```

We specify the well to be rate-controlled by setting its type to `'rate'`. The annual injection rate is chosen to be 3.5 megatonnes of CO_2. Note that rate should be given in volumetric terms, which means we have to divide by reference CO_2 density on the first code line. Reference CO_2 density is provided by the field `rhoGS` in the fluid object.

Lateral boundary conditions must also be specified. This is generally a difficult issue for CO_2 storage modeling, since flow behavior and even geology outside aquifer boundaries may be poorly known, and since the choice of boundary conditions will often have a significant impact on the simulation outcome.

No-flow conditions are natural to assign to lateral parts of the boundary that correspond to pinch-outs with vanishing thickness of the permeable layer. Elsewhere, some extent of fluid flow across boundaries must generally be assumed. Open boundaries could be modeled by prescribing constant pressure on boundary faces. However, imposing constant pressure on lateral boundaries will not account for fluid flow resistance and gradual pressure build-up outside the simulated domain. Attempts to capture such effects include endowing boundary cells with very large, compressible volumes, adding virtual production wells, increasing flow resistance across boundaries by locally lowering permeability, as well as coupling with some analytical or numerical model representing outside aquifers. See [36] for examples of boundary modeling choices applied on the Johansen model.

When the `makeJohansenVEgrid` function was called in Section 2.1.1 above, a vector with indices to non-closed lateral boundary faces, `bcIx`, was returned. We will here use these indices to set up a `MRST` structure that describes hydrostatic pressure conditions on these faces, by calling the `addBC` function:

```
% Start with an empty set of boundary faces
bc = [];

% Computing hydrostatic pressure for boundary cells
p_bc = G.faces.centroids(bcIx, 3) * rhow * g(3);

% Add hydrostatic pressure conditions to open boundary faces
bc = addBC(bc, bcIx, 'pressure', p_bc);
```

To visualize the location of the open faces on the simulation grid, we can use the `plotFaces` command:

```
plotGrid(G, 'facecolor', 'none', 'edgealpha', 0.2);
plotFaces(G, bcIx);
axis tight; view(-40, 40)
```

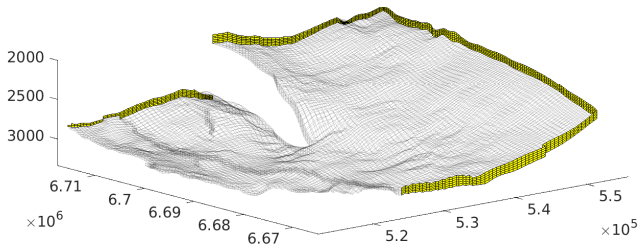


Figure 2.6: Open boundaries of the Johansen simulation model.

As is clear from the figure, the faces along the sealing fault are not considered open, nor are the faces along the western border, where aquifer thickness vanishes.

2.1.5 Schedules

A *schedule* provides the necessary information at each simulation timestep about the operational state of each well, as well as eventual changes to boundary conditions. During operation, wells may be switched on or off, operational parameters may be changed over time, and different fluids may be injected at different timesteps (alternating injections).

We will finish the discussion of the data model by specifying a schedule for our Johansen example. We aim to simulate an injection scenario where CO_2 is injected for 100 years, followed by 1000 years of migration. The well thus has two states: active and turned off. Boundary conditions will remain constant at all times.

To construct our schedule, we create two copies of the well model and set the well rate to zero in the second copy, representing the well after the injection period has ceased. Our schedule will contain two *controls*, which each specifies a particular choice of well state and boundary conditions. Next, we define the number and length of timesteps and specify which of the two controls is active at each. Control 1 will be in use for the first hundred timesteps, representing the injection period, and control 2 will be used for second hundred timesteps, representing the migration period.

The complete code to set up the schedule is:

```
% Setting up two copies of the well and boundary specifications.
% Modifying the well in the second copy to have a zero flow rate.
schedule.control    = struct('W', W, 'bc', bc);
schedule.control(2) = struct('W', W, 'bc', bc);
schedule.control(2).W.val = 0;

% Specifying length and number of simulation timesteps
schedule.step.val = [repmat( 1 * year, 100, 1); ...
                     repmat(10 * year, 100, 1)];

% Specifying which control to use for each timestep.
% The first 100 timesteps will use control 1, the last 100
% timesteps will use control 2.
schedule.step.control = [ones(100, 1); ...
                         ones(100, 1) * 2];
```

We have now specified all the data we need to simulate a CO_2 injection scenario into the Johansen formation. Next, we need to identify the mathematical equations that govern the evolution of our system over time.

2.2 The mathematical model

Having presented all the data going into the physical description of the model, attention is now turned to the differential equations and mathematical relationships that define the evolution of the system, and how these are practically solved by a numerical simulator.

It is important to note that a numerical simulation model is specified in terms of discrete data (partitioning of space into cells, state variables for a finite number of timesteps). On the other hand, the equations describing the evolution of the system are derived and formulated for the continuous setting. This means that some form of discretization must be applied on the equations before they can be used as a basis for numerical simulation.

The mathematical equations that describe two-phase (or general multi-phase) flow are obtained by extending the equations for single-phase flow, which will be presented first. After having discussed two-phase flow, the role of geomechanics and how it can be modeled in a poroelastic framework is explained. The section concludes with a discussion on how a CO_2-brine system that includes dissolution of CO_2 into brine can be formulated, discretized and practically solved within a "black-oil" framework.

In what follows, characters in boldface are used to describe vector or tensor quantities.

2.2.1 Mass conservation, Darcy's law and single-phase flow

Local conservation of mass is the fundamental principle underlying the description of single-phase fluid flow through a porous medium. This principle dictates that the change in mass within any given volume and over any time interval equals the flow of mass across the volume boundaries during that same time interval. By shrinking the considered volume and time interval to their infinitesimal limits, the principle of mass conservation can be stated in the form of a continuity equation:

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = q. \tag{2.4}$$

Here, ρ represents fluid density, ϕ porosity, $\mathbf{v}$ the *superficial velocity* (volumetric fluid flow rate per unit surface of porous medium) and q a source term[4]. The divergence operator $\nabla\cdot$ is applied on the mass flux vector $\rho\mathbf{v}$. The first term on the left of this equation is commonly referred to as the *accumulation term*, the second as the *flux term*, and the right hand side as the *source term*. As informally explained in Section 2.1.1, fluid flow $\mathbf{v}$ is associated with the pressure gradient ∇p through an empirical, linear relation referred to as Darcy's law, which involves the permeability tensor $\mathbf{K}$, the fluid viscosity μ and the gravity vector $\mathbf{g}$:

$$\mathbf{v} = -\frac{\mathbf{K}}{\mu}(\nabla p - \rho \mathbf{g}). \tag{2.5}$$

By combining (2.4) and (2.5) above and assuming that ρ and ϕ are functions of pressure only, we obtain an equation where the only unknown is the pressure field p:

$$\frac{d}{dp}(\rho\phi)\frac{\partial p}{\partial t} - \nabla \cdot (\rho \frac{\mathbf{K}}{\mu}(\nabla p - \rho \mathbf{g})) = q. \tag{2.6}$$

The simulation of a simple, isothermal single-phase system thus requires just a single state variable, namely pressure. In general, (2.6) is a parabolic equation. However, when considering an incompressible system where ρ and ϕ remain constant, the first term vanishes and (2.6) reduces to a Poisson (elliptic) equation with variable coefficients. The system ceases to be time-dependent.

2.2.2 Two-phase immiscible flow

The fundamental difference between a one-phase and a multi-phase flow system is that the latter must keep track of mass conservation of several distinct phases. Therefore, phase saturations must be introduced as new unknowns to the set of state variables describing the system. This will also entail further complexities in the form of new constitutive relationships depending on these new unknowns.

To begin, we will limit ourselves to two-phase immiscible flow. The system considered thus consists of two phases that remain completely separate at all times (no dissolution or evaporation of one phase into the other). When this assumption holds, independent mass conservation equations can be formulated for each phase α:

$$\frac{\partial(s_\alpha \rho_\alpha \phi)}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{v}_\alpha) = q_\alpha. \tag{2.7}$$

The subscript α indicates a variable associated with the specific phase α, which for CO_2 storage scenarios will normally be brine or CO_2. We see that the accumulation term now includes the phase saturation, which as we recall from Section 2.1.2 represents the fraction of local pore space occupied by that phase.

[4]Given that mass can be neither created nor destroyed, the presence of a source term in this continuity equation may seem surprising. However, the source term is typically used to represent mass fluxes that remain unaccounted for in the practical definition of the volume boundary. In particular, for reservoir simulation it is used to represent injection from and extraction to wells.

Phase velocities are linked to phase pressures using the multiphase extension of Darcy's law:

$$\mathbf{v}_\alpha = -\frac{\mathbf{K}k_{r,\alpha}}{\mu_\alpha}(\nabla p_\alpha - \rho_\alpha \mathbf{g}). \tag{2.8}$$

Compared with the single-phase version of Darcy's law, the multi-phase formulation introduces two new complexities associated with multiphase flow: the relative permeabilities $k_{r,\alpha}$ and the presence of separate phase pressures p_α.

From the perspective of a given phase α, the presence of other phases in the pore space will affect its flow properties. In almost all cases, this results in a reduction in the medium's effective permeability as experienced by the phase. To a large extent, this reduction is a result of the reduction in pore space available for phase α to flow, but is also linked to effects arising from phase-phase and phase-rock interactions at the molecular level, such as surface tension and wettability. The relative permeability $k_{r,\alpha}$ of a phase is defined as the (dimensionless) ratio of the effective over the absolute permeability $\mathbf{K}$ of the porous rock as experienced during flow. Generally speaking, this is a tensorial quantity that depends on several factors including saturation, pressure, temperature and phase composition [13]. In practice however, relative permeabilities are usually modeled as scalar functions of saturation. The graph corresponding to a relative permeability function is most often a convex curve, monotonously increasing from zero (for $s_\alpha = 0$) to one (for $s_\alpha = 1$). A simple mathematical model for this type of behavior is to use a power law, i.e., to posit:

$$k_{r,\alpha}(s_\alpha) = s_\alpha^{N_\alpha}. \tag{2.9}$$

for $N_\alpha \geq 1$. This type of model is often referred to as a modified Brooks-Corey relation, due to its similarity with a specific model for oil-gas systems first described by Corey in [26].

When we constructed our fluid object in Section 2.1.3, we passed along a parameter pair, "`'n', [2 2]`", that was not fully explained at the time. These represented the exponents of automatically generated relative permeability functions on the form of (2.9). We can plot these two functions, `krW` and `krG`, in terms of brine saturation with the following code lines:

```
sw = linspace(0, 1, 200);
figure; hold on;
plot(sw, fluid.krW(sw),   'b', 'linewidth', 1.5); % brine rel. perm. curve
plot(sw, fluid.krG(1-sw), 'r', 'linewidth', 1.5); % CO2 rel. perm. curve
xlabel('water saturation'); ylabel('relative permeability')
```

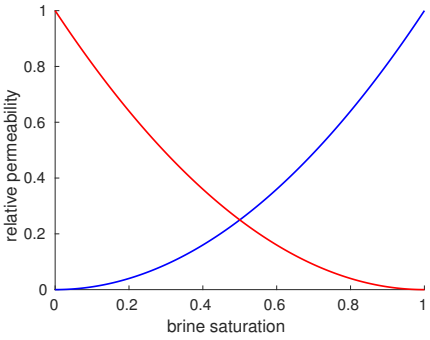


Figure 2.7: Simple model of a two-phase relative permeability system. In our example, the blue curve represents brine and the red CO_2.

The resulting plot is shown in Figure 2.7. We note that the figure is completely symmetric, a feature of our very simple model. In the real world, things are more complex. Relative permeability functions can be experimentally obtained in the laboratory for specific rock/fluid systems. An example is shown in the left plot of Figure 2.8, which shows curves measured for a brine/CO_2 system on a sample of Cardium sandstone from Alberta, Canada [13]. The figure presents two sets of relative permeability curves: one set corresponding to *drainage* (where CO_2 is in the process of displacing brine) and another to *imbibition* (where brine is displacing CO_2). The differences between drainage and imbibition curves are caused by hysteretic behavior in the two-phase fluid flow system, originating from differences in how one fluid replaces the other at the pore scale [62].

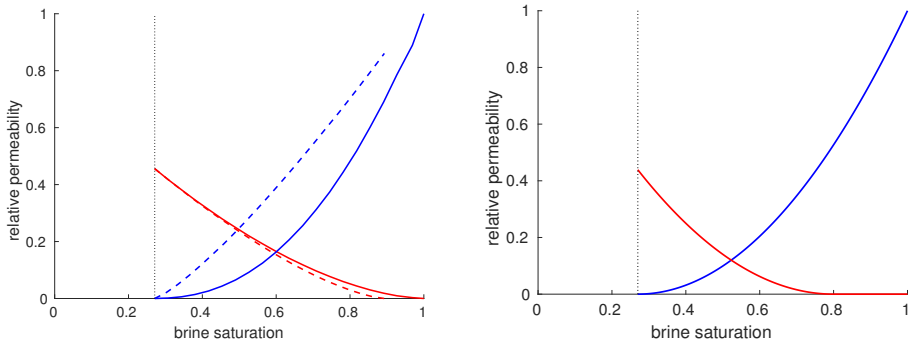


Figure 2.8: *Left:* Measured CO_2 (red) and brine (blue) relative permeabilities for Cardium sandstone. Solid curves represent drainage, dashed curves imbibition. The vertical line highlights the place where residual brine saturation is reached. *Right:* Curves resulting from a rescaling of our simple quadratic model.

Another noteworthy feature of the figure is that the relative permeability of brine reaches zero at a saturation well above zero. As a consequence, brine will stop flowing altogether once this saturation is reached during a drainage process. The remaining

brine is stuck by capillary forces as disjoint droplets and films in the pore space. This is referred to as *residual brine*, and its fractional amount of the pore space is called *residual brine saturation*, denoted $s_{w,r}$. This means that CO_2 can never reach a higher saturation than $1 - s_{w,r}$, and its curve is thus truncated on the plot. A similar phenomenon can be seen in the imbibition curves for CO_2, where in the course of an imbibition process the relative permeability reaches zero before brine saturation has reached one. The remaining amount of CO_2 is referred to as *residual CO_2*, and its saturation value as *residual CO_2 saturation*, denoted $s_{n,r}$.

In our simple example, we avoid the additional complexity of introducing hysteresis, and will use the same set of relative permeability curves to describe flow regardless of whether drainage or imbibition is taking place. However, we do want to capture the effect of residual saturation, which represents one of the main trapping mechanisms of CO_2 as we already discussed in Section 1.2. One easy way to obtain this is by rescaling the argument to our relative permeability functions, so that they reach zero when the corresponding saturation reaches its residual value $s_{r,\alpha}$, rather than zero. In other words, we define the relative permeabilities functions by applying the simple power law on a rescaled saturation s^*_α defined as:

$$s^*_\alpha = \frac{s_\alpha - s_{r,\alpha}}{1 - s_{r,\alpha}} \tag{2.10}$$

In our example code, we can therefore redefine our existing functions by passing them modified parameters:

```
srw = 0.27; % residual brine saturation
src = 0.20; % residual CO2 saturation
fluid.krW = @(s) fluid.krW(max((s-srw)./(1-srw), 0)); % redefine krW
fluid.krG = @(s) fluid.krG(max((s-src)./(1-src), 0)); % redefine krG
```

The residual brine and CO_2 saturation values used above are reasonably within the range of measured values as reported in literature [19]. If we plot the newly obtained curves using the previous plotting code, we obtain the right graph of Figure 2.8, which are qualitatively much more similar to the experimentally obtained curves than those from our original model.

To round off the discussion of relative permeabilities, it should be mentioned that the residual saturation of a given phase is generally not a fixed value, but will depend on previous saturation history, in particular the maximal previously reached phase saturation. To include this dependence in a numerical simulation requires more advanced models for residual saturation and relative permeability than those sketched here [58, 62].

When presenting the multiphase extension of Darcy's law in (2.8), the presence of separate phase pressures was pointed out. In fact, when two or more phases are present in a porous medium like rock, capillary forces will drive flow towards an equilibrium where the pressure for each phase will differ. The physical mechanism behind capillary forces is related to the presence of interface surface tensions and different phase 'wetting' properties (how strongly the molecules in the phase adheres to the solid medium) [62]. The difference in phase pressures is called *capillary pressure*, and is denoted p_c. It is defined as $p_c = p_n - p_w$ were p_n denotes the pressure in the *non-wetting* phase

(usually CO_2 for a CO_2-brine system), whereas p_w denotes the pressure in the *wetting* phase (usually brine for a CO_2-brine system).

When a porous medium contains pores with a distribution of different sizes (like most rocks do), it is possible to express capillary pressure as a function of phase saturation. We can for instance write $p_c = p_c(s_w)$, where s_w is the wetting (brine) saturation. For a particular fluid/rock system, this relationship can be experimentally measured. By plotting pressure against saturation, we obtain curves that typically qualitatively look like the left plot of Figure 2.9.

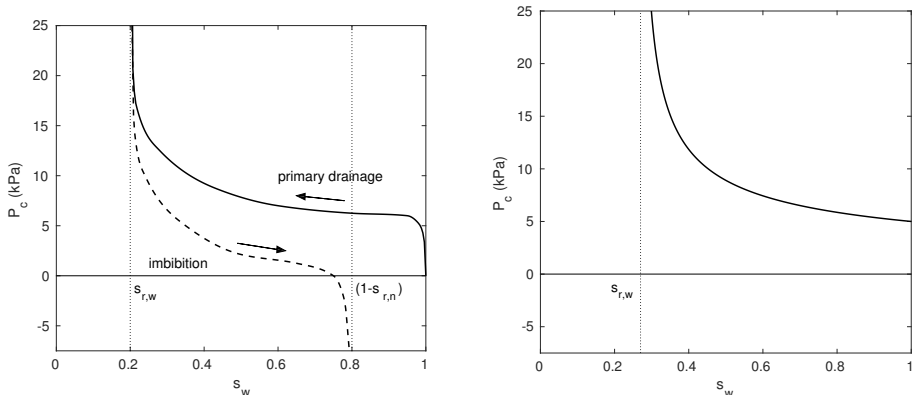


Figure 2.9: *Left:* Example of drainage (solid line) and imbibition (dashed line) capillary pressure curve pair. Here, $s_{r,w}$ and $s_{r,n}$ denote the residual saturations of the wetting and the nonwetting phases, respectively. *Right:* Simple capillary pressure model for the Johansen example.

The two curves in this plot represent the capillary pressure as a function of saturation during initial drainage (when CO_2 displaces brine for the first time), and imbibition (when brine re-invades pore space occupied by CO_2). Again, we have a system that exhibits hysteresis. From the primary drainage curve, we can note two things. First, a finite increase in capillary pressure is required in order for CO_2 to enter a fully saturated brine system (the abrupt jump at $s_w = 1$). For pressure differentials lower than this value, CO_2 will not be able to enter the porous medium at all. This is referred to as *capillary exclusion* and is the mechanism that in general prevents CO_2 from entering the confining caprock layer in a geological storage scenario.

The second thing to notice is that capillary pressure goes to infinity when the wetting saturation approaches the value of the residual saturation. In other words, no amount of additional CO_2 pressure will be able to drive the remaining brine out of the rock. A similar behavior can be seen in the imbibition curve, where capillary pressure tends to negative infinity when CO_2 saturation approaches its residual threshold value.

By default, the fluid object we obtained from `initSimpleADIFluid` has not been assigned any capillary pressure function, which is interpreted by the simulation code as a capillary pressure of zero (phase pressures always equal). However, we can choose to include a nonzero capillary pressure by constructing a new function and adding it to the fluid object. The function must be named `pcWG`, and depend on CO_2 saturation only. The following code provides an example:

```
pe = 5 * kilo * Pascal; % define a reasonable entry pressure
pcWG = @(sw) pe * sw.^(-1/2); % define function without residual sat.

% Rescale input argument to account for nonzero residual saturation, and
% change argument from brine to water saturation (in accordance with
% what is expected by the simulator engine).  Avoid singularity at
% zero by capping to a negligible nonzero saturation value, here 1e-5.
fluid.pcWG = @(sg) pcWG(max((1-sg-srw)./(1-srw), 1e-5));
```

For the example above, we have used a very simple model for the capillary pressure function, which is obtained by inverting the following relation between saturation and capillary pressure, proposed by Brooks and Corey [18]:

$$s_w = \begin{cases} (p_c/p_e)^{-n_b}, & \text{if } p_c > p_e \\ 1 & p_c \leq p_e. \end{cases} \tag{2.11}$$

The parameter p_e here represents entry pressure, which we have here set to 5 kPa. We have chosen 1/2 as the value of the parameter n_b, which is related to pore size distribution. We show the resulting function on the right plot of Figure 2.9. Needless to say, many other, more advanced models are available in literature.

At this point, our equation system contains four unknowns (phase pressures and saturations), and four equations: two differential equations (the two mass conservation equations, combined with the multiphase extension of Darcy's law), and two direct functional relationships (phase saturations must sum to one; capillary pressure determined from saturation). Since the direct functional relationships provide immediate links between variables, there is no need to explicitly keep track of more than two independent state variables. A common (but not the only possible) choice is to use one phase pressure and one phase saturation as independent unknowns. There is thus no need to introduce additional state variables to our Johansen simulation example.

2.2.3 Miscible flow, dissolution and black-oil formulation

For miscible flow, each fluid phase may consist of multiple components, and phases may exchange components with each other. Below, the general structure of the conservation equations describing multi-component flow is first briefly outlined. Then, the attention will turn to the particular example of a miscible two-phase system, with two fully developed formulations for modeling a brine-CO_2 system that includes CO_2 dissolution into brine.

When modeling multi-component flow, a set of distinct material *components* and a set of distinct fluid phases is considered. A simple example would be a brine-CO_2 system with CO_2, H_2O and salt as separate components, and with the distinct CO_2 and brine phases that have already been discussed. Each phase may consist of more than one component. For example, some H_2O may have evaporated from the brine phase into the CO_2 phase, and reciprocally some CO_2 may have dissolved into the brine phase, which also contains a certain amount of dissolved salt.

The quantitative composition of a phase is described by its *mass fractions*. For a given phase α, the mass fraction of component i is denoted m_α^i, and refers to the ratio

of mass of the component to the total mass of the fluid phase. It follows from this definition that the mass fractions for a phase must sum to one, i.e., for each phase α we have:

$$\sum_i m_\alpha^i = 1. \tag{2.12}$$

For multi-component systems, the standard approach is to apply the conservation equations on each individual component instead of each phase. Since each component may be found in more than one phase, a summation across phases must be applied. When considering a system with N components and M phases, the conservation of component i is expressed:

$$\frac{\partial}{\partial t}\left(\phi \sum_{\alpha=1}^{M} m_\alpha^i \rho_\alpha s_\alpha\right) + \nabla \cdot \left(\sum_{\alpha=1}^{M} m_\alpha^i \rho_\alpha \mathbf{v}_\alpha + j_\alpha^i\right) = \sum_{\alpha=1}^{M} m_\alpha^i \rho_\alpha q_\alpha. \tag{2.13}$$

In addition to the summation of a component across phases, another qualitative difference between the above equation and Equation (2.7) is the introduction of the terms j_α^i, which represent non-advective flux of a given component within a given phase, driven by diffusion or dispersion. A simple model for this process is linear Fickian diffusion, in which case we write:

$$j_\alpha^i = -\rho_\alpha s_\alpha D_\alpha^i \nabla m_\alpha^i, \tag{2.14}$$

where ∇ is the gradient operator and D_α^i is the *diffusion tensor* for component i in phase α.

The number of variables needed to describe a multi-component system can become quite high. With N components distributed over M phases, the number of individual mass fractions m_α^i is $N \times M$. In addition, we need to keep track of M saturations and M phase pressure values. The number of relations discussed thus far (N continuity equations combined with the multiphase extension of Darcy's law; M relations on the form (2.12); up to $M-1$ capillary pressure relations; and the requirement that phase saturations must sum to one) will not suffice to close the system except in trivial cases. Additional equations are generally needed to describe how components transfer between or distribute across phases. One approach is to assume that components instantly partition across phases according to a set of system-specific equilibrium relations. This is often referred to as *flash calculations*. When instant equilibrium cannot be assumed, the time-dependent transfer of individual components from one phase to another will need to be modeled. This usually requires introducing mass balance equations for individual components *within* individual phases, and let component transfer be modeled in the form of source or sink term.

Another complication is that phase densities ρ_α and viscosities μ_α will now depend not only on pressure and temperature, but also on phase composition. To account for this, more advanced equations of state and transport property models have to be employed.

To illustrate the above, one may consider a very simple example of a two-component, two-phase system with CO_2 and water. Each component and phase is here denoted with c for CO_2 and w for water. As such, this system presents a total of 8 unknowns: 4 mass fractions m_c^c, m_c^w, m_w^c and m_w^w, 2 phase saturations s_c and s_w, and 2 phase pressures p_c and p_w.

A matching number of equations is needed to close the system. Equations already discussed above include: 2 continuity equations (one per component), 1 capillary pressure relation, 2 requirements that mass fractions sum to one (one per component), and 1 requirement that phase saturations sum to one. Two more equations need to be specified. If evaporation of water into CO_2 is considered to be negligible, one has $m_c^w = 0$, which provides one (trivial) equation. It remains to identify a relation modeling the amount of CO_2 that will dissolve into water.

The equations (2.13) for the simple CO_2-water system under consideration can be reformulated on a form known as the *black-oil equations*. This form, originally used to represent three-phase miscible flow involving oil, gas and water, is widely adopted in reservoir simulator software. A system that can be presented on this form is thus easy to include in existing simulator frameworks.

To formulate the conservation equations of the CO_2-water system in the black-oil framework, one specifies a reference pressure and temperature, associated with "standard conditions", and introduces the *formation volume factors*, defined as:

$$B_i = \frac{V_i}{V_i^S} \quad \left(= \frac{\rho_i^S}{\rho_i} \right). \tag{2.15}$$

Here, V_i and V_i^S here denote the volume occupied by a bulk of component i at reservoir and standard conditions, respectively. Likewise, ρ_i and ρ_i^S refer to component density at reservoir and standard conditions. For isothermal models, B_i are functions of pressure only[5], and for ease of notation in what follows we also define $b_i = 1/B_i$. The variable r_c is used to denote the amount of dissolved CO_2 per volume of water, measured as a CO_2/water volumetric fraction at standard conditions, and $r_{c,s}$ represents the value of r corresponding to fully CO_2-saturated water.

Neglecting the terms j_α^i and assuming that water volume does not appreciably change when CO_2 is dissolved, the equations (2.13) for the simple CO_2-water system can be rewritten on the following black-oil form:

$$\frac{\partial}{\partial t}(\phi b_w s_w) + \nabla \cdot (b_w \mathbf{v}_w) = b_w q_w \tag{2.16}$$

$$\frac{\partial}{\partial t}(\phi (b_c s_c + r_c b_w s_w)) + \nabla \cdot (b_c \mathbf{v}_c + r_c b_w \mathbf{v}_w) = b_c q_c. \tag{2.17}$$

A relation is still needed to describe the amount of CO_2 dissolved into water at any given time. One approach is to assume that CO_2 will instantly dissolve in brine up to saturation $r_{c,s}$ is reached, or until local CO_2 in gas form is depleted, whatever comes first. This can be considered a (very simple) example of a flash calculation. The additional relation will then be:

$$r_c = r_{c,s} \text{ for } s_c > 0 \text{ and } s_c = 0 \text{ for } r_c < r_{c,s}. \tag{2.18}$$

In other words, if the amount of CO_2 within a specified volume can be fully dissolved in water, it will, and r_c takes on the corresponding value. Otherwise, water will be

[5] Strictly speaking, B_i are also functions of phase composition, i.e., the amount of one phase that is evaporated/dissolved in the other.

fully saturated ($r_c = r_{c,s}$) and any excess CO_2 is present as a separate gas phase. In the included Paper III, this model is referred to as the *instantaneous dissolution model*.

If instantaneous dissolution cannot be assumed, a rate-driven model must be considered instead. For the vertical equilibrium models introduced in Section 3.1, this may be a more reasonable choice, as these models describe fluid phases that to a large extent occupy different regions of space, and where the effective dissolution rate is governed by convective mixing process, which may act over long time scales [38]. In such a case, Equation (2.18) is replaced by a separate conservation equation for dissolved CO_2:

$$\frac{\partial}{\partial t}(\phi r_c b_w s_w) + \nabla \cdot (r_c b_w \mathbf{v}_w) = q_{c,\text{rate}}. \tag{2.19}$$

Here, $q_{c,\text{rate}}$ describes the rate of dissolution, and can only be positive when $s_c > 0$ and $r_c < r_{c,s}$. In Paper III, this is referred to as the *rate-dependent dissolution model*.

2.2.4 Solving the multi-phase flow system

The equations we have presented so far describe the temporal evolution of a multi-phase, multi-component flow system in a continuous setting. A continuous description corresponds with physical reality at the scales considered, but cannot be used directly to describe the evolution in a simulation model where space is subdivided into a set of discrete cells and time is represented as a finite number of separate steps. In order for the equations to be applicable in this setting, they first have to be *discretized*, i.e., expressed in approximate form as relations between the sets of discrete values representing the system's state variables. These equations can be solved numerically to evolve the system from one timestep to the next. Provided that a suitable discretization has been used, the solution thus obtained will approximate and converge towards the continuous solution as the spatial and temporal resolutions are increased.

The choice of a proper discretization for a given set of differential equations is not trivial. Different methods have different trade-offs in terms of truncation error, convergence behavior, computational efficiency and adherence to absolute physical requirements involving mass conservation, continuity, monotonicity and entropy. A wrong choice of discretization can lead to solutions that are unstable, converge to the wrong solution, require extremely short timesteps, or introduce nonphysical oscillations or high levels of numerical smearing.

A detailed discussion on discretization is outside the scope of this text. Instead, the focus will be on an approach frequently encountered in reservoir simulation software and implemented in `MRST`, where the conservation equations are discretized using a *finite-volume* approach in space and implicit first-order discretization in time. This is followed by a demonstration where a fully implicit solver from `MRST` is used to run the simulation of our example scenario on the Johansen model.

The finite-volume discretization is based on the strict enforcement of the mass conservation principle at the grid cell level. In other words, for any given time interval, the change in the amount of a conserved quantity within each gridcell should exactly equal the flux of that quantity across cell faces during that interval, plus the eventual source term. By associating an unique flux with each face in the grid, we also ensure flux continuity, since this entails that any quantity flowing out of cell A across a shared face with cell B will equal the quantity flowing into cell B across that same face.

Using superscript n to denote timesteps and subscripts i and f to respectively refer to grid cell and grid face indices, the time-implicit finite volume discretization of Equation (2.7) for grid cell i can be written:

$$\frac{\phi_i^{n+1}\rho_{\alpha,i}^{n+1}s_{\alpha,i}^{n+1} - \phi_i^n\rho_{\alpha,i}^n s_{\alpha,i}^n}{t^{n+1}-t^n} + \sum_{f\in\mathcal{F}_i}\rho_{\alpha,f}^{n+1}v_{\alpha,f}^{n+1} - q_{\alpha,i}^{n+1} = 0. \tag{2.20}$$

Here, t refers to time, $\mathcal{F}_i$ to the set of grid faces bounding cell i, and $v_{\alpha,f}$ to the net volumetric flux of phase α across face f, using a sign convention where outflow is positive. Assuming isothermal conditions, $\rho_{\alpha,i}^n$ is used as a shorthand for $\rho_\alpha(p_i^n)$, and likewise $\phi_i^n = \phi_i(p_i^n)$. The first term in the discretized equation refers to the change in mass of phase α within cell i over the time interval $t^{n+1}-t^n$. The second term refers to the mass phase flux across the cell boundary, and the third term (which has been moved to the left-hand side compared with Equation (2.7)) is the source term.

The face fluxes $v_{\alpha,f}^{n+1}$ are obtained from a discretized version of the multiphase extension of Darcy's law (Equation (2.8)). A simple but frequently employed way of doing this is to use a scheme referred to as *two-point flux approximation* (TPFA). Assuming that face f represents the common interface between cells k and l, the flux from cell l to cell k across f can be approximated using TPFA as:

$$v_{\alpha,f} = -T_f\lambda_{\alpha,f}^{n+1}\left[p_{\alpha,k}^{n+1} - p_{\alpha,l}^{n+1} - \rho_{\alpha,f}^{n+1}g(z_k - z_l)\right]. \tag{2.21}$$

In this expression, $\lambda_{\alpha,f}^{n+1} = \lambda_{\alpha,f}(s_f^{n+1}, p_f^{n+1})$ denotes phase *mobilities*, which are shorthand for relative permeability divided by viscosity, i.e.: $\lambda_\alpha(s,p) = k_{r,\alpha,f}(s_\alpha)/\mu_\alpha(p_\alpha)$. Moreover, z_k and z_l represent the depth value of the respective cell centroids, and T_f the *transmissibility* associated with face f. The face transmissibility is a value based on the associated cell geometries and permeability values. It can be described as the harmonic average of the associated half-face transmissibilities:

$$T_f = \left(T_{k,f}^{-1} + T_{l,f}^{-1}\right)^{-1}, \tag{2.22}$$

where the half-face transmissibility associated with cell k and face f is given by:

$$T_{k,f} = \frac{\mathbf{c}_{k,f}^T\mathbf{K}_k\mathbf{N}_{k,f}}{||\mathbf{c}_{k,f}||^2}. \tag{2.23}$$

In this expression, $\mathbf{K}_k$ represents the permeability tensor of cell k, $\mathbf{c}_{k,f}$ the vector pointing from the cell centroid of cell k to the face centroid of face f, and $N_{k,f}$ the area-scaled normal of face f, pointing out of cell k.

The TPFA scheme is computationally lightweight and easy to implement, but only consistent with Equation (2.8) for grids where $\mathbf{K}_k\mathbf{N}_{k,f} \mid\mid c_{k,f}$ (so-called K-orthogonal grids). On grids that do not satisfy K-orthogonality, artifacts or errors related to the orientation of the grid will be present in the solution. Despite these grid orientation effects, the TPFA scheme is widely used also on more general grids. More advanced methods exist that remain consistent on general grids, including multipoint flux approximation schemes [6], mixed finite elements [17], and the mimetic method [28].

Finally, the discretized system is closed using the remaining relations from the continuous case: $\sum_\alpha s_{\alpha,i} = 1$ and $p_{n,i} - p_{w,i} = p_c(s_i)$. In the rest of the present discussion,

we will simply use $\mathbf{s}$ and $\mathbf{p}$ (no subscript α) to refer to the vectors of chosen unknowns associated with a specific phase, knowing that the saturation and pressure values for the opposite phase can be immediately obtained using these relations.

In Equations (2.20) and (2.21), fluxes $v_{\alpha,f}$ are associated with grid faces. However, the equations also involve other quantities projected on faces, namely $\rho_{\alpha,f}$ and $\lambda_{\alpha,f}$. These quantities depend on pressure and saturations, whose discrete representation is associated with cells rather than faces. Since each interior face is connected to two cells, the question arises on how to define the saturation and pressure values for which to compute $\rho_{\alpha,f}$ and $\lambda_{\alpha,f}$. A simple average of cell values does not work well in practice. Instead, *upwind weighting* is employed, where the saturation and pressure values used to compute $\rho_{\alpha,f}$ and $\lambda_{\alpha,f}$ are taken from the upstream cell, i.e., the cell where the flow is directed outwards across the face. This has proven to be robust in practice, is supported by physical considerations (the properties that influence flow should be those associated with the fluid actually crossing the interface), and is in line with the upstream differencing widely employed in computational fluid dynamics. In the case of one-dimensional two-phase finite-volume discretization with an imposed flux, the use of upwind weighting has been proven to converge to the physically correct solution [16].

The discretization shown in (2.20) and (2.21) is an example of an *implicit* scheme, since the unknown quantities $\mathbf{s}^{n+1}$ and $\mathbf{p}^{n+1}$ cannot be explicitly computed from the previous values $\mathbf{s}^n$ and $\mathbf{p}^n$ but must be arrived at by solving an equation system. Moreover, this system is clearly nonlinear as it involves multiplication of quantities that depend on the unknown variables, as well as (potentially) nonlinear functions λ_α and ρ_α. In order to compute the new values of $\mathbf{s}$ and $\mathbf{p}$, we must therefore use an iterative approach. Denoting the vector of unknowns $\mathbf{x}^{n+1} = [\mathbf{s}^{n+1}, \mathbf{p}^{n+1}]$, the equation system defined by (2.20) and (2.21) (combined for both phases) can be expressed as:

$$\mathbf{G}(\mathbf{x}^{n+1}, \mathbf{x}^n, \mathbf{u}^{n+1}) = \mathbf{0}, \tag{2.24}$$

where $\mathbf{u}^{n+1}$ represents externally imposed controls (typically well rates or pressures regulating the source terms $\mathbf{q}_\alpha^{n+1}$). This system can be solved iteratively using the Newton-Raphson method. Starting from an initial guess $\mathbf{x}_0^{n+1}$, we define the following sequence $\{\mathbf{x}_1^{n+1}, ..., \mathbf{x}_k^{n+1}, ...\}$:

$$\mathbf{x}_{k+1}^{n+1} = \mathbf{x}_k^{n+1} - J_{\mathbf{G}}^{-1}(\mathbf{x}_k^{n+1})\mathbf{G}(\mathbf{x}_k^{n+1}, \mathbf{x}_k^n, \mathbf{u}^{n+1}), \tag{2.25}$$

where $J_{\mathbf{G}}(\mathbf{x}_k^{n+1})$ denotes the Jacobian of $\mathbf{G}$ with respect to $\mathbf{x}^{n+1}$. For a sufficiently good initial guess (typically chosen as the solution at the previous timestep), this sequence will converge to the correct root $\mathbf{x}$ of $\mathbf{G}$. If convergence is not achieved, the timestep will have to be shortened. It should be mentioned that for a real simulation, equations in (2.20) would be modified to account for boundary conditions, and supplemented with additional equations to link well pressures and rates to reservoir flow before constructing $\mathbf{G}$.

The use of the Newton-Raphson method is conceptually simple and has quadratic convergence rate. However, a prerequisite for using the method is the ability to compute the Jacobian $J_{\mathbf{G}} = (\frac{\partial \mathbf{G}_i}{\partial x_j})_{ij}$. For our simple example, the Jacobian of $\mathbf{G}$ could be implemented directly as long as the derivatives of ρ_α and λ_α were available, but for general

G this process quickly becomes complicated and bug-prone. One way to obtain the Jacobian of any function **G** that can be defined in terms of simple algebraic operations and simpler functions whose derivatives are known, is to use automatic differentiation (AD) when implementing the function. AD is a programming technique that automatically computes numerical derivatives to machine precision of any expression involving the use of a special class of variables that include derivative information. A description of the technique and how it is implemented in `MRST` is provided in Appendix A.

Formulas (2.20)-(2.22) describe a fully implicit discretization of the two-phase flow equations, with pressure and saturations solved as a coupled system. The use of such formulations is prevalent in simulation software used for practical reservoir engineering [5, 63, 89, 96]. There are however other established alternatives. One approach widely used in the research community is based on a reformulation of the mass conservation equations (2.7) into separate equations for pressure evolution and phase transport. This formulation is referred to as the *fractional flow formulation* [85]. It has the advantage that it separates the system into two subcomponents that have inherently different behavior, and can thus be efficiently solved in a sequential manner by using numerical techniques adapted for each component. The pressure equation will have a strongly *elliptic* character, where changes in boundary conditions or well pressure propagate almost instantly throughout the simulation domain. On the other hand, the equation describing evolution in saturations ("transport equation") will have a strong *hyperbolic* character, where information is steadily propagated along identified directions ("characteristics"). This approach is also widely known, and also supported by `MRST`.

Everything is now ready to demonstrate how to use `MRST` to employ the fully implicit solution strategy to simulate the scenario previously defined for our Johansen example. So far, the grid `G`, the rock properties `rock`, the initial state `initState` and the operating schedule `schedule` (which includes information about wells and boundary conditions) have been defined. The next step is to assemble the grid, fluid and rock into a specific simulation model object as follows:

```
% Constructing simulation model object
mrstModule add ad-core; % load necessary module
model = twoPhaseGasWaterModel(G, rock, fluid, 0, 0);
```

The construction of a simulation model assigns a set of discretized equations (the mathematical model) with the specified data model. Several simulation models are implemented in `MRST`. In our case, we use `twoPhaseGasWaterModel` to construct a two-phase immiscible model consisting of gas and water phases, based on Equations (2.7) and (2.8). The two last arguments to the call above are related to the vertical thermal profile. Since the properties of our fluid object do not depend on temperature (isothermal conditions assumed), these arguments remain unused and we arbitrarily set them to zero.

We then invoke the solver by calling `simulateScheduleAD`, which in addition to the model takes the initial conditions and the operating schedule:

```
% Running simulation
[wellSol, states] = simulateScheduleAD(initState, model, schedule);
```

Running this command will start computing the solution for each simulation

timestep.

```
Solving timestep 001/200:                          -> 1 Year
Solving timestep 002/200: 1 Year                   -> 2 Years
Solving timestep 003/200: 2 Years                  -> 3 Years
...
Solving timestep 200/200: 1090 Years               -> 1100 Years
```

Upon completion, the simulation results are returned in the form of two data structures, `wellSol` and `states`, which respectively provide, for each timestep, well information (pressure and rates) and the values of state variables. On way to visually inspect the result is to use `plotCellData`, which we here employ to show the CO_2 distribution in the aquifer after 100 years (end of injection) and after 1100 years (end of the simulated migration).

```
% Plot CO2 saturation at timestep 100
plotCellData(G, states{100}.s(:,2)); view(-63, 68); colorbar;
% Plot CO2 saturation at timestep 200
plotCellData(G, states{200}.s(:,2)); view(-63, 68); colorbar;
```

The resulting plots are shown in Figure 2.10. We note that just after the end of injection, during which flow has been mainly driven by pressure, the plume outline remains relatively regular and centered on the injection point. Thousand years after injection, however, most of the plume has migrated upwards and left a clearly visible trail of residual CO_2 saturation in its wake. The areas where high concentrations of CO_2 still remain are mainly confined to structural traps in the caprock.

2.2.5 Geomechanics and poroelasticity

So far, the modeling focus has been on fluid flow, with little discussion of the impact of pressure changes in the aquifer on the solid medium itself. In recent years, increasing attention has been paid to the geomechanical aspects of geological CO_2 storage, i.e., the study of how rock stresses and strains are impacted by the elevated pore pressure caused by the injection of large amounts of CO_2 underground. Geomechanical side effects may include ground uplift (already observed at the In Salah injection site [88, 93]), rock fracturing that may compromise storage integrity, and fault reactivation, which may both compromise storage integrity and also lead to notable seismic events [91]. In this context, geomechanical studies become important from a safety perspective, both to estimate safe pressure levels [9, 92] and to interpret observed behavior during operation [14].

Even in the absence of fractures or fault issues, geomechanical effects influence reservoir fluid flow, as rock porosity and permeability may be significantly affected by changes to the rock strain and stress fields [29]. In Section 2.1.3, we introduced a simple relation to represent how pore volume expands with increasing pressure, based on the concept of pore volume compressibility. As argued in Paper IX in Part II of the thesis, this simple model is adequate in many practical cases for simulating flow, and is extensively used in practice. However, it does not accurately represent the physics involved and may therefore give unsatisfactory results in situations where strong variations in

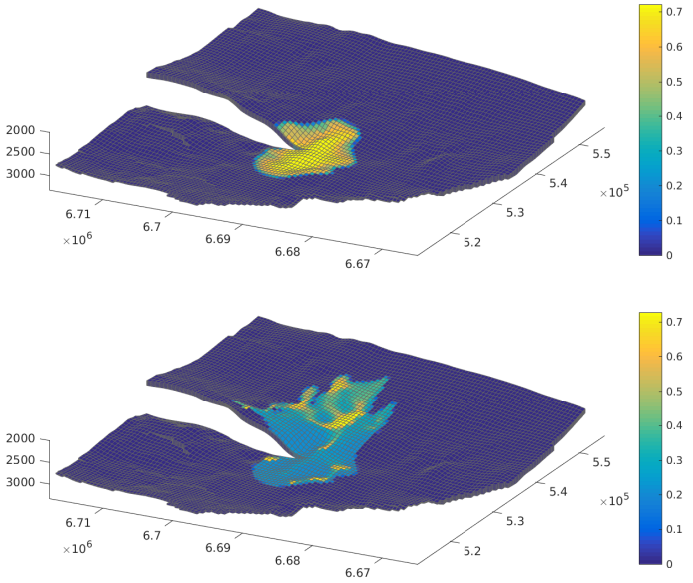


Figure 2.10: CO_2 saturation in the Johansen injection scenario. *Top:* After 100 years (end of injection). *Bottom:* After 1100 years (end of simulation).

the pressure fields cause significant rock deformation, typically occurring around injection wells. In reality, the change in pore space for a given control volume of rock can be separated into changes due to the compression of individual rock grains, and changes caused by expansion or contraction of the control volume itself[6]. Whereas compression of rock grains depends on local pore pressure, changes to the control volume itself is described by rock strain, which depends on pore pressure in a global sense and can only be computed by coupling geomechanical equations with flow equations in the simulation model.

The inclusion of geomechanics in a flow simulation incurs significant computational cost. In addition to the flow unknowns (pressure, saturations, etc.), geomechanics adds a new set of unknowns, e.g., in terms of 3D displacements. Moreover, it is not enough to model the aquifer in isolation; typically the full overburden up to the surface has to be included, as well as a large part of the underlying geology, significantly extending the computational domain. If geomechanics is included to evaluate the potential for fault slip or fracturing, it may be sufficient to solve the flow problem in isolation and compute mechanical stresses and strain as a post-processing step. On the other hand, if the influence of geomechanical deformation on flow is relevant, the fully coupled system must be addressed. Over the past decade, there has been a significant research focus on fully-coupled geomechanical and flow models and how the resulting equation system can be efficiently solved [30, 47, 59, 66, 70, 97, 98]. Paper IX proposes an

[6]We here assume a control volume in the Lagrangian sense, i.e., a volume tied to a specific amount of rock material. Expansion/contraction of the rock thus translates to expansion/contraction of the associated control volume.

alternative approach to address this computational issue.

As a fluid-saturated porous medium, the subsurface can be understood as a *poromechanical* system. In cases where mechanical displacements remain small, the mechanical behavior of this system can be modeled in terms of the equations of linear poroelasticity [110]. These equations arise by coupling the equations of linear elasticity with the conservation equation for fluid flow through fluid pressure. For the solid medium, the equations describe a state of mechanical equilibrium, i.e., balance of forces. At every point of the simulation domain, we associate a displacement vector:

$$\mathbf{u} = [u_x, u_y, u_z]^T. \tag{2.26}$$

Assuming displacements to be small, we define the infinitesimal *strain tensor* $\boldsymbol{\varepsilon}$ to be the symmetric part of the gradient of the displacement field:

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right). \tag{2.27}$$

This tensor represents infinitesimal changes in length associated with the displacement field. Internal forces are described by the *stress tensor* $\boldsymbol{\sigma}$, a space-dependent rank-two tensor that associates a directed force to each possible plane through a given point in the medium. The stress tensor can be described by a 3x3 matrix, and rotational equilibrium requires that it be symmetrical:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T. \tag{2.28}$$

Moreover, translational equilibrium requires that the divergence of the stress tensor counterbalances body forces $\mathbf{F}$ (usually gravity in practice):

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{F} = 0. \tag{2.29}$$

If tensile stresses ("stretching") are taken to be positive, the stress tensor can be expressed in terms of *effective stress* $\boldsymbol{\sigma}'$ and *pore pressure* p as follows:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}' - \alpha p\mathbf{I}. \tag{2.30}$$

Here, $\mathbf{I}$ represents the identity tensor and α the *Biot-Willis coefficient*, a scalar value between ϕ (porosity) and 1 that relates to the ratio between rock grain compressibility and compressibility of the porous medium itself. Hooke's law links effective stress with the strain tensor, using a fourth-order elasticity tensor $\mathbf{C}$:

$$\boldsymbol{\sigma}' = \mathbf{C}\boldsymbol{\varepsilon}. \tag{2.31}$$

For an isotropic material, C can be expressed in terms of two separate elastic moduli, which are properties of the porous medium. There are several equally valid choices for elastic moduli to describe the relation. If we choose *bulk modulus* K (resistance to compression) and *shear modulus* G (resistance to shear), relation (2.31) can be spelled out as:

$$\boldsymbol{\sigma}' = 2G\boldsymbol{\varepsilon} + (K - \frac{2}{3}G)\mathrm{tr}(\boldsymbol{\varepsilon})\mathbf{I}. \tag{2.32}$$

Combining Equations (2.29)-(2.32), and inserting the gravitational force $-\rho_b\mathbf{g}$ for $\mathbf{F}$ (ρ_b denoting bulk density), the displacement formulation of the force balance equation can be written:

$$\nabla \cdot (G\nabla\mathbf{u}) + \nabla\left((K + \frac{1}{3}G)\nabla \cdot \mathbf{u}\right) - \nabla(\alpha p) = \rho_b\mathbf{g}. \tag{2.33}$$

Pore pressure p in this equation is linked to the flow equation. For simplicity, we here stick to the equation for single phase flow, which we discussed in Section 2.2. In the development of Equation (2.6), the accumulation term was expressed in terms of density and porosity, assumed to depend on pressure only. In poroelastic literature, it is more common to work with the *increment of fluid content* ζ, whose time derivative expresses the volumetric rate of fluid flow in/out of a control volume. The relation with the accumulation term in Equation (2.6) can be understood as:

$$\rho \frac{\mathrm{d}}{\mathrm{d}t}\zeta = \frac{\mathrm{d}}{\mathrm{dt}}(\phi\rho). \tag{2.34}$$

Increment of fluid content is considered to depend on pressure and volumetric strain $\varepsilon = \mathrm{tr}(\boldsymbol{\varepsilon}) = \nabla \cdot \mathbf{u}$, so that its time derivative can be expressed it as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\zeta = \frac{\partial \zeta}{\partial p}\frac{\mathrm{d}p}{\mathrm{d}t} + \frac{\partial \zeta}{\partial \varepsilon}\frac{\mathrm{d}\varepsilon}{\mathrm{d}t}. \tag{2.35}$$

It can be shown (c.f. Appendix B) that $\frac{\partial \zeta}{\partial \varepsilon} = \alpha$, the Biot-Willis coefficient. Moreover, $\frac{\partial \zeta}{\partial p}$ is referred to as the *specific storage coefficient at constant strain* and denoted S_ε. If linear compressibility is assumed for the fluid, the following expression can be derived for S_ε [110]:

$$S_\varepsilon = \frac{1}{K}(1-\alpha)(\alpha-\phi) + \frac{\phi}{K_f}, \tag{2.36}$$

where K is the bulk modulus of the medium and $\frac{1}{K_f}$ is the compressibility of the fluid.

Dividing Equation (2.6) by ρ (neglecting the term in $\nabla\rho$) and replacing the accumulation term in the resulting expression by (2.35), we obtain the following formula for the flow equation ($q_{\mathrm{vol}} = \frac{q}{\rho}$):

$$\alpha \frac{\mathrm{d}}{\mathrm{d}t}(\nabla \cdot \mathbf{u}) + S_\varepsilon \frac{\mathrm{d}}{\mathrm{d}t}p - \nabla \cdot \left(\frac{\mathbf{K}}{\mu}(\nabla p - \rho \mathbf{g})\right) = q_{\mathrm{vol}}. \tag{2.37}$$

Together, equations (2.33) and (2.37) describe a linear elastic poromechanical system on a domain Ω, where the unknowns are pore pressure p and displacements $\mathbf{u}$. In addition, boundary conditions need to be specified. Boundary conditions for flow are typically specified in terms of boundary pressures or flow rates, whereas mechanical boundary conditions are specified in terms of forces or displacements at $\partial\Omega$.

2.2.6 Thermal models

For the sake of completeness, a simple set of equations for describing conservation and transfer of energy, needed when describing non-isothermal processes, is presented below.

The *specific internal energy* (energy per unit mass) of a fluid phase α depends on temperature τ and pressure p, denoted $e_\alpha(p,\tau)$. Likewise, we use $e_{\mathrm{rock}}(p,\tau)$ to refer to the *volumetric* internal energy of the rock. The related quantity *specific enthalpy* h is defined as:

$$\begin{aligned} h_\alpha(p,\tau) &= e_\alpha(p,\tau) + p v_\alpha(p,\tau) \\ &= e_\alpha(p,\tau) + \frac{p}{\rho_\alpha(p,\tau)}, \end{aligned}$$

where v_α is the specific volume, i.e., volume per unit mass.

Energy can enter or exit a control volume either by thermal conduction (transfer through matter) or advection (transport by a fluid phase). The energy flux by thermal conduction $\mathbf{q}_{\text{cond}}$ can be described using Fourier's law:

$$\mathbf{q}_{\text{cond}} = -\kappa \nabla \tau. \tag{2.38}$$

The heat conductivity κ here represents that of the complete solid-fluid system, and will therefore depend on the phase saturations s_α. If we assume kinetic energy, gravity work, thermal dispersion and other sources of energy (e.g., radiation) to be negligible, the conservation equation for energy can be written:

$$\frac{\partial}{\partial t}\left(\phi \sum_\alpha s_\alpha \rho_\alpha e_\alpha + (1-\phi) e_{\text{rock}}\right) + \nabla \cdot \left(\sum_\alpha \rho_\alpha h_\alpha \mathbf{v}_\alpha\right) - \nabla \cdot \kappa \nabla \tau = q_e. \tag{2.39}$$

This equation consists of an accumulation term, a transport term, a diffusive term and (on the right hand side) a source term. The fluid phase velocities $\mathbf{v}_\alpha$ can be linked to pressure using the multiphase extension of Darcy's law shown in (2.8). The use of enthalpy rather than internal energy in the transport term implicitly accounts for the pressure work involved to move fluids around.

Chapter 3

Simplified simulation models

As argued in Chapter 1 and illustrated in Chapter 2, the modeling of geological CO_2 storage is in general based on the same concepts and equations as reservoir modeling. In principle, contemporary 3D simulation tools used in reservoir engineering could be employed to investigate issues related to CO_2 injection and migration in a saline aquifer. For studies limited to the local region during a CO_2 injection process, this is a reasonable approach. However, for the study of long-term CO_2 migration processes, a number of practical concerns make the use of "traditional" 3D simulation tools highly problematic in practice:

- Due to the density difference with brine, migration of CO_2 is primarily in the form of a thin plume below the caprock. To accurately resolve the plume shape and capture its behavior, a very high *vertical* resolution is needed.
- Long-term CO_2 migration is primarily gravity driven, involving a thin plume drifting upslope beneath a confining caprock. The process is therefore highly sensitive to caprock geometry. To correctly capture the geometry, a high *lateral* grid resolution is needed.
- The spatial and temporal scales can be very large (hundreds of kilometers of lateral extent, and thousands of years of migration).
- Due to the large geographical areas involved, data is expected to be scarce. As a consequence, parameters will be poorly constrained and a large number of simulations might have to be carried out to investigate different possibilities.

In summary, the use of standard 3D simulation tools to investigate long-term CO_2 migration quickly becomes prohibitively expensive from a computational point of view. This chapter introduces some alternative CO_2 storage modeling tools that require significantly less computational resources while still capturing the main behavior of the system. Section 3.1 presents a simulation approach based on the assumption of *vertical equilibrium* (VE) in combination with depth integration, which reduces the dimensionality of the problem while retaining most of the 3D behavior. Section 3.2 presents an even simpler approach based on simple geometrical analysis of caprock shape, related to methods used for studying primary migration in the context of basin modeling. The combined use of these tools to search for an optimal injection scenario is presented in Section 3.4, which also discusses more generally how simplified modeling approaches

can be applied together as components to form larger workflows. A newly proposed approach for rapid computation of coupled geomechanical effects is presented in Section 3.3.

As in Chapter 2, the use of the introduced methods is demonstrated on the Johansen data model, using concrete code examples that the reader can try out him/herself using implementations found in `MRST-co2lab`. Moreover, in Appendix C some interactive demonstration tools available in the same software package are presented, allowing the user to further explore the tools presented in this chapter by applying them on real datasets.

3.1 Vertical equilibrium models

Motivation and justification

Strictly speaking, in the context of CO_2 modeling,"vertical equilibrium models" could more precisely be called "transverse equilibrium models". The key assumption underlying such models is the existence of equilibrium in the transverse direction, i.e., the direction perpendicular to the local aquifer plane. For general aquifers, the transverse direction is not necessarily co-linear with the true vertical direction, but may make a small, nonzero angle to it. In the discussion below, the word "vertical" is nevertheless used to refer to the transverse direction, with the implicit understanding that this entails that the gravity vector must be considered to have separate "vertical" and "lateral" components.

The thin, laterally extensive nature of a typical aquifer means that vertical flow will constitute a very small component of overall flow (c.f. Figure 1.1). Moreover, the significant density difference between brine and CO_2[1] means that the fluids will tend to segregate relatively quickly due to gravity forces. This separation process can be considered complete when vertical equilibrium has been established between capillary and gravity forces. The end state is characterized by separate CO_2 and brine zones and no vertical flow, i.e., a vertical pressure field in fluid-static equilibrium. If capillary forces are negligible compared to gravity, the CO_2 and brine zones will be separated by a sharp interface; otherwise a transition zone of finite thickness will be present. If the time it takes for vertical equilibrium to develop is short compared with the time scale considered for the simulation, the corresponding flow can be practically neglected and the system assumed to be in vertical fluid-static equilibrium at all times. As such, the vertical pressure and fluid distribution profiles can always be reconstructed from knowledge of a set of *upscaled* variables that only depend on the lateral coordinates. By assuming VE and a known saturation distribution, upscaled variables and parameters can be obtained from the original conservation equations by vertical integration. By doing so, the vertical direction is eliminated from the equation system, reducing the dimensionality of the model from three to two. The practical significance is that the resulting equations become much less expensive to solve from a computational standpoint. Although the system obtained is two-dimensional, much of the behavior of the three-dimensional system is still implicitly captured by the definition of the upscaled variables.

[1] A significant density difference between brine and CO_2 can be assumed in practically all relevant scenarios.

A look at a aquifer cross-section plot of our Johansen model simulation confirms the assumption of fluid segregation. Figure 3.1 shows the computed CO_2 saturation at the end of the injection period and at the end of the simulation period. At the end of the injection period, the contour of a CO_2 plume that is spreading under the caprock can be clearly distinguished. The plume only reaches the bottom of the aquifer around the injection well, and its CO_2 saturation is relatively constant. On the right plot one can see how the plume has migrated upslope just below the caprock, leaving a trail of residual CO_2 in its wake. One can also discern how it has collected in two separate structural traps along the way. The plot moreover suggest that the vertical resolution might be inadequate to properly capture the plume shape, in particular its advancing tip.

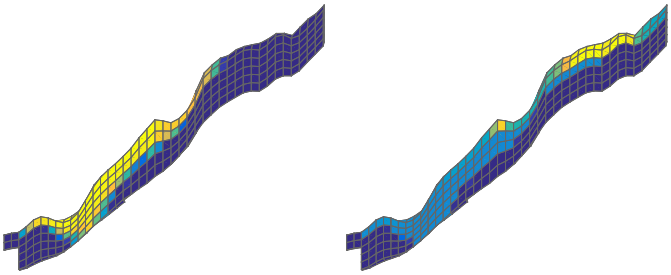


Figure 3.1: Vertical section of the Johansen aquifer, showing the saturation of CO_2 in the local region around the injection site (at the left side of the plots). *Left:* CO_2 saturation after 100 years of injection. *Right:* CO_2 saturation after another 1000 years of migration.

Models based on the VE assumption are not new. In hydrology, unconfined water flow is often treated using the Dupuit approximation, where hydrostatic pressure distribution is assumed and flow is considered essentially horizontal, thus reducing the number of spatial dimension by one [11]. The same principle was introduced for two-phase flow models used in the oil industry in the 50's and 60's [25, 32, 69], when computing resources were scarce. On the other hand, the development and use of such models in the context of CO_2 storage is comparatively recent [79, 80], and is a response to the aforementioned computational challenges inherent in the modeling of geological CO_2 storage.

When can the VE assumption be expected to hold? A frequently cited analytical study of the two-dimensional two-phase flow equations [114] defines the number

$$R_L = \frac{L}{H}\sqrt{\frac{k_V}{k_H}},$$

where L is a characteristic length scale, H a characteristic height scale, and k_V and k_H represent vertical and lateral aquifer permeability. An asymptotic analysis demonstrates that the vertical pressure profile in the domain will be increasingly well described by vertical pressure equilibrium with increasing R_L, and cites numerical results to argue that equilibrium is satisfied quite well for $R_L \geq 10$. By itself, the study provides a basis for arguing that the high aspect ratio of a typical aquifer supports the VE assumption. However, the time scale at which equilibrium can be assumed for a given

value of R_L remains elusive, for two reasons. First, the analysis, which is limited to two dimensions, is predicated on a specific injection velocity q, in the absence of which the formulations fall apart. Moreover, both R_L and the characteristic time used scale linearly with L, signifying that an increasing value of R_L caused by an increased value of L does not lead to a more rapid establishment of VE, although this *would* be the case if R_L increased due to a change in H, k_V or k_H.

A different argument [81], based on estimating the time it takes for CO_2 to rise to the top of a vertical column, provides the following formula for a dimensionless time t_z^* associated with vertical segregation:

$$t_z^* = \frac{t\lambda_n^\dagger k_z \Delta\rho g \cos\theta}{\phi H}. \tag{3.1}$$

Here, t represents time, $\Delta\rho$ the density difference, g gravity, θ aquifer dip angle, ϕ porosity, H aquifer thickness and $\lambda_n^\dagger$ an assumed characteristic mobility of the CO_2 phase. It is argued that VE cannot be assumed for $t_z^* < 1$. A later study [27] uses practical simulations with realistic parameters to show that the applicability of VE models is also limited by the time scale of *brine drainage*, and presents a formula very similar to (3.1), with the difference that the characteristic mobility $\lambda_w^\dagger$ is that of brine, not of CO_2.

Several studies exist where results from simulations based on the VE assumption are compared to full 3D models. Simulations of a scenario based on the Johansen Formation [65] suggest that, in addition to the computational advantage, results from using a VE model can be superior to those obtained from 3D simulations, due to the errors caused by limited vertical grid resolution in the latter. A similar conclusion can be reached from comparing simulations of CO_2 injection at the Sleipner site [76], where two VE models are compared with the 3D results using the commercial ECLIPSE simulation software. The VESA VE-model [41] has been included in several comprehensive benchmark and comparison studies [23, 73, 82]. In [23], the results from using VESA were generally within the spread of results from the 3D-based models. In [82], the results from using a VE model also compared well with those obtained from 3D simulations. It was argued that some differences observed (higher plume tip speed, more pronounced dissolution) could be closer to the true problem, as the vertical resolution remains a limitation of the 3D models.

Derivation

The definition of a VE model for a CO_2-brine system requires a description of the assumed vertical saturation distribution and phase pressure profiles. In what follows, a VE formulation is derived for a system with incompressible fluids, where the flowing CO_2 plume and the underlying brine zone are separated by a sharp interface. In other words, the transition zone ("capillary fringe") between brine and CO_2 regions is considered to be thin enough to be neglected. In terms of the underlying two-phase flow physics, this situation arises when capillary pressure at the pore scale varies only weakly as a function of saturation. The assumption of a sharp interface leads to a VE model whose formulation is relatively simple to derive, but it is not strictly necessary. Paper IV of this thesis presents some general VE formulations where capillary pressure plays a significant role and the resulting capillary fringe has a significant impact

on the resulting flow. A formal VE formulation for systems with compressible fluids is derived in Paper I.

If residual saturation is included in the model, the vertical distribution of phases in an aquifer can be described as illustrated in Figure 3.2. A vertical column through the aquifer can thus be divided in three zones of zero or larger thickness. Denoting the residual saturations for CO_2 and brine respectively $s_{r,n}$ and $s_{r,w}$, the zones can be described from the top to the bottom as:

1. a zone with free-flowing CO_2 (plume), of thickness h. CO_2 saturation here is 1-$s_{r,w}$ and brine saturation $s_{r,w}$;
2. a zone that used to be part of the plume, but where brine has re-invaded pore-space so that only residual CO_2 remains. Saturations are thus $s_{r,n}$ for CO_2 and $1 - s_{r,n}$ for brine. The thickness of this zone is $h_{\max} - h$, where $h_{\max}$ is the historically largest value of h up to present simulation time;
3. a zone of thickness $H - h_{\max}$ consisting of brine only, where H is the local aquifer thickness.

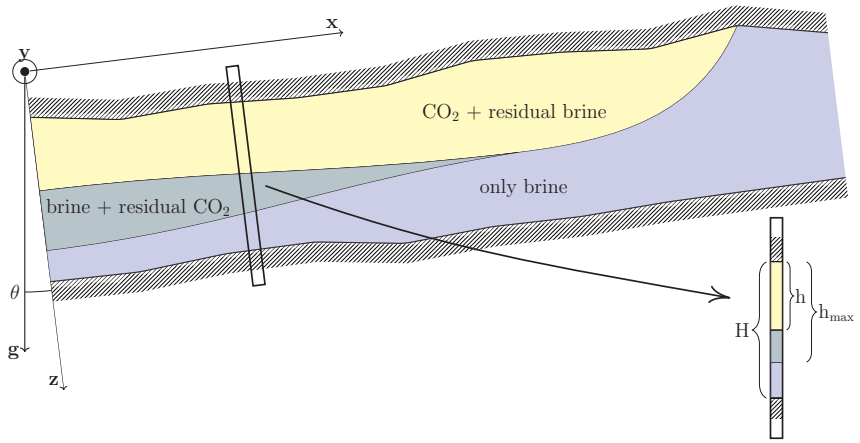


Figure 3.2: Assumed phase distribution in a cross-section of an aquifer for a sharp-interface VE model with residual trapping. The definitions of H, h and $h_{\max}$ are indicated for a vertical column. Note that on this figure, the coordinate system is tilted so that the x-axis is aligned with the aquifer slope θ.

The fluid-static pressure profiles corresponding to this saturation distribution is illustrated in Figure 3.3. The fluid pressure of phase $\alpha = \{n, w\}$ has a vertical gradient of $\rho_\alpha g \cos\theta$, where g is the gravitational constant and θ the inclination of the coordinate system (see Figure 3.2). The different fluid densities thus lead to pressure profiles with different steepness. At the level of the CO_2-brine interface (depth h), the difference in phase pressure is equal to the CO_2 entry pressure $p_{c,i}$, which is here shown as a finite nonzero value, but is often neglected in practice (considered zero).

P_n and P_w are used to denote phase pressures at some predefined reference level, which is here chosen to be the caprock level[2]. The difference $P_c = P_n - P_w$ is referred

[2]This is a somewhat arbitrary choice; other choices of reference level lead to equally valid VE formulations.

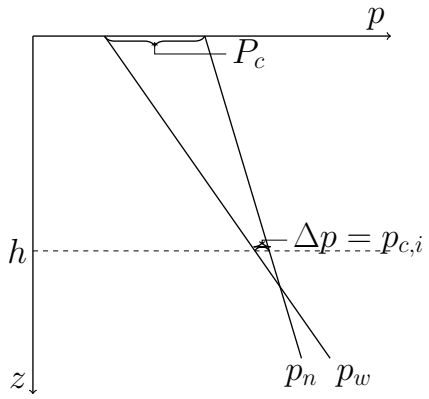


Figure 3.3: Hydrostatic pressure profiles for CO_2 (p_n) and brine (p_w) phases. At the interface between the CO_2 and brine phases (depth h), the pressure difference equals the CO_2 entry pressure $p_{c,i}$. Upscaled capillary pressure is indicated as P_c.

to as *upscaled* (or *pseudo*) *capillary pressure*, and can be computed from h as:

$$P_c(h) = p_{c,i} + gh(\rho_w - \rho_n). \tag{3.2}$$

In the upscaled VE equations, P_c plays a role analogue to that of fine-scale capillary pressure p_c in the original flow equations. Sticking to the inclined coordinate system of Figure 3.2, we have that h, P_n and P_w are functions of the lateral coordinates x and y.

Finally, the z-coordinates of the top and bottom surfaces of the aquifer can be expressed as functions of lateral coordinates: $\zeta_T(x,y)$ represents the caprock level and $\zeta_B(x,y)$ the aquifer bottom. The function describing the full, three-dimensional pressure field can then be expressed:

$$p_\alpha(x,y,z) = P_\alpha(x,y) + \rho_\alpha g[z - \zeta_T(x,y)]\cos\theta. \tag{3.3}$$

Everything is now in place for deriving the sharp-interface VE formulation. Starting from the phase mass-conservation equations (2.7) combined with the multiphase extension of Darcy's law (2.8), the VE formulation is obtained by integrating from aquifer top to bottom along the vertical dimension:

$$\frac{\partial}{\partial t}\int_{\zeta_T}^{\zeta_B} s_\alpha \phi_\alpha \mathrm{d}z - \int_{\zeta_T}^{\zeta_B} \nabla\cdot\left(\frac{k_{r,\alpha}}{\mu_\alpha}\mathbf{K}(\nabla p_\alpha - \rho_\alpha \mathbf{g})\right)\mathrm{d}z = \int_{\zeta_T}^{\zeta_B}\frac{q_\alpha}{\rho_\alpha}. \tag{3.4}$$

As fluids are here considered to be incompressible, the Equation (2.7) has been divided by the constant phase density ρ_α. The gravity vector $\mathbf{g}$ can be decomposed into a lateral and a transversal component, where the lateral component $\mathbf{g}_{||}$ lies in the aquifer plane, and the transversal component follows the z-direction with magnitude $g\cos\theta$. By using $\nabla_{||}$ to denote the lateral component of the del operator, i.e., $\nabla_{||} = [\partial_x, \partial_y]$, and assuming zero flow across top and bottom boundaries, Leibniz' rule can be applied on the second integral to obtain:

$$\frac{\partial}{\partial t}\int_{\zeta_T}^{\zeta_B} s_\alpha \phi_\alpha \mathrm{d}z - \nabla_{||}\cdot\int_{\zeta_T}^{\zeta_B}\left(\frac{k_{r,\alpha}}{\mu_\alpha}\mathbf{K}(\nabla p_\alpha - \rho_\alpha \mathbf{g})\right)\mathrm{d}z = \int_{\zeta_T}^{\zeta_B}\frac{q_\alpha}{\rho_\alpha}. \tag{3.5}$$

By assuming a tensor $\mathbf{K}$ that can be decomposed into separate, independent lateral ($\mathbf{K}_{||}$) and transversal ($\mathbf{K}_{\perp}$) components, lateral flow will only depend on the lateral pressure gradient. Inserting (3.3) for fine-scale pressure p_α, (3.5) can be further developed:

$$\frac{\partial}{\partial t}\int_{\zeta_T}^{\zeta_B} s_\alpha \phi_\alpha \mathrm{d}z - \nabla_{||}\cdot\left[\int_{\zeta_T}^{\zeta_B}\frac{k_{r,\alpha}}{\mu_\alpha}\mathbf{K}_{||}\mathrm{d}z\right]\left(\nabla_{||}(P_\alpha-\rho_\alpha g\cos\theta\zeta_T)-\rho_\alpha \mathbf{g}_{||}\right)=\int_{\zeta_T}^{\zeta_B}\frac{q_\alpha}{\rho_\alpha}. \tag{3.6}$$

By introduction of a set of upscaled variables, Equation (3.6) can be rewritten as:

$$\frac{\partial}{\partial t}(\Phi S_\alpha)-\nabla_{||}\cdot\left[\boldsymbol{\Lambda_\alpha}\boldsymbol{\kappa}\left(\nabla_{||}(P_\alpha-\rho_\alpha g\cos\theta\zeta_T)-\rho_\alpha\mathbf{g}_{||}\right)\right]=Q_\alpha. \tag{3.7}$$

The definitions of the upscaled variables are:

$$\begin{aligned}
\Phi &= \int_{\zeta_T}^{\zeta_B}\phi\,\mathrm{d}z\\
\boldsymbol{\kappa} &= \int_{\zeta_T}^{\zeta_B}\mathbf{K}_{||}\mathrm{d}z\\
S_\alpha &= \Phi^{-1}\int_{\zeta_T}^{\zeta_B}\phi s_\alpha\mathrm{d}z\\
\boldsymbol{\Lambda_\alpha} &= \left(\int_{\zeta_T}^{\zeta_B}\frac{k_{r,\alpha}}{\mu_\alpha}\mathbf{K}_{||}\mathrm{d}z\right)\boldsymbol{\kappa}^{-1}\\
Q_\alpha &= \int_{\zeta_T}^{\zeta_B}\frac{q_\alpha}{\rho_\alpha}\mathrm{d}z.
\end{aligned}$$

The upscaled saturations S_α and mobilities $\boldsymbol{\Lambda_\alpha}$ are defined for each phase, and can be explicitly expressed in terms of h and $h_{\max}$ based on the sharp interface assumption as follows:

$$S_n(h,h_{\max}) = \Phi^{-1}\left[(1-s_{r,w})\int_{\zeta_T}^{\zeta_T+h}\phi\,\mathrm{d}z + s_{r,n}\int_{\zeta_T+h}^{\zeta_T+h_{\max}}\phi\,\mathrm{d}z\right] \tag{3.8}$$

$$S_w(h,h_{\max}) = 1-S_n(h,h_{\max}) \tag{3.9}$$

$$\boldsymbol{\Lambda_n}(h) = \left(\lambda_{n,e}\int_{\zeta_T}^{\zeta_T+h}\mathbf{K}_{||}\mathrm{d}z\right)\boldsymbol{\kappa}^{-1} \tag{3.10}$$

$$\boldsymbol{\Lambda_w}(h,h_{\max}) = \left(\frac{1}{\mu_w}\int_{\zeta_T+h_{\max}}^{\zeta_B}\mathbf{K}_{||}\mathrm{d}z+\lambda_{w,e}\int_{\zeta_T+h}^{\zeta_t+h_{\max}}\mathbf{K}_{||}\mathrm{d}z\right)\boldsymbol{\kappa}^{-1}. \tag{3.11}$$

In the above formulations, the symbols $\lambda_{n,e}$ and $\lambda_{w,e}$ are used to represent endpoint mobilities at the fine scale, i.e., $\lambda_{n,e}=\frac{k_{r,n}(1-s_{r,w})}{\mu_n}$ and $\lambda_{w,e}=\frac{k_{r,w}(1-s_{r,n})}{\mu_w}$. It is worth pointing out that if in addition the rock properties ϕ and $\mathbf{K}$ are assumed to be constant in the vertical direction, the above formulas for upscaled variables simplify considerably,

becoming:

$$\Phi = H\phi \tag{3.12}$$

$$\boldsymbol{\kappa} = H\mathbf{K}_{||} \tag{3.13}$$

$$S_n(h, h_{\max}) = \left[(1 - s_{r,w})h + s_{r,n}(h_{\max} - h)\right]/H \tag{3.14}$$

$$S_w(h, h_{\max}) = 1 - S_n(h, h_{\max}) \tag{3.15}$$

$$\boldsymbol{\Lambda_n}(h) = \lambda_{n,e}h/H \tag{3.16}$$

$$\boldsymbol{\Lambda_w}(h) = \left(\frac{1}{\mu_w}(H - h_{\max}) + \lambda_{w,e}(h_{\max} - h)\right)/H. \tag{3.17}$$

The equations in (3.7) are two-dimensional in nature, and expressed in terms of the four unknowns S_α and P_α. Similar to the fine-scale equation system, the number of unknowns is reduced to two by including the upscaled capillary pressure function P_c and the requirement that the upscaled phase saturations must sum to 1. As such, one may choose one upscaled saturation (S_n or S_w) and one reference phase pressure (P_n or P_w) as the independent unknowns when solving the equation system. The use of upscaled saturations when expressing and solving the VE equations is referred to as the *s-formulation*. This approach has the advantage of remaining very similar to the original 3D equations, allowing them to be solved using existing 3D simulator code with only modest modifications.

On the other hand, the expressions for upscaled capillary pressure (3.2) and mobilities (3.10, 3.11) are computed in terms of h and $h_{\max}$, not in terms of upscaled saturations. This requires the introduction of functions that allow the inverse computation of h and $h_{\max}$ from S_n and $S_{n,\max}$[3], which, depending on the complexity of the VE model used, can be relatively simple or quite complex. Details are discussed in Paper IV of this thesis. An alternative is to use h directly as an independent unknown instead of S_n or S_w. This approach, known as the *h-formulation*, allows for easy computation of upscaled mobilities and capillary pressures. In that respect, the choice of h as an independent unknown could be considered a "natural" choice for a VE system. The h-formulation constitutes a larger departure from the fine-scale equation system (2.7), as h has no direct analogue at the fine scale[4]. For fully implicit VE simulations, `MRST-co2lab` employs the s-formulation of the equations, expressed on a black-oil form similar to (2.16-2.17), which also models CO_2 dissolution into brine as detailed in Paper III. For systems solved using operator-splitting techniques (rather than fully-implicit), the h-formulation is also supported.

The computational domain of the VE equations (3.7) is a 2D grid representing the top surface (or more generally a specified reference surface) of the modeled aquifer. However, essential 3D information remains implicitly present in the simulation. Aquifer thickness and impacts of vertical rock heterogeneities are accounted for in the definition of upscaled variables. Moreover, the impact of varying caprock topography is preserved in the term $\nabla_{||}(\rho_\alpha g \cos\theta \zeta_T)$ of (3.7), and aquifer slope by the term $\rho_\alpha \mathbf{g}_{||}$. In practice, for real aquifer models, a global aquifer slope may not exist,

[3]Here, $S_{n,\max}$ is used to denote the hysteretic variable representing the historically maximum value for S_n per vertical column.

[4]By multiplying with $(\rho_w - \rho_n)g$ and neglecting entry pressure $p_{c,i}$, one can however think of h as corresponding to the fine-scale capillary pressure at the caprock level, c.f. Equation (3.2) and Figure 3.3.

and the introduction of local slope will require the introduction of a spatially varying coordinate system with associated computational complexity. When a global slope is not present, a practical choice is therefore to consider θ to be zero globally, and let the impact of local slope be entirely captured by the term $\nabla_{||}(\rho_\alpha g \zeta_T)$. The error associated with this approximation is discussed in the appendix of Paper III. As `MRST-co2lab` focuses on real aquifer models, nonzero slope is currently not supported.

Practical example using `MRST-co2lab`

To demonstrate the use of VE models, we revisit our Johansen example and run the scenario again, this time applying the VE functionality provided by `MRST-co2lab`[5].

The following code will assume that the Johansen grid has been loaded and all variables previously defined in Chapter 2 are present. The following code snippet ensures this to be the case:

```
% Ensuring previously defined information is present
gravity on;
g       = gravity;
rhow    = 1000;               % water density
co2     = CO2props();         % CO2 property object
p_ref   = 30 *mega*Pascal;    % reference pressure
t_ref   = 94+273.15;          % reservoir temperature
co2_rho = co2.rho(p_ref, t_ref); % CO2 density at ref. conditions
co2_c   = co2.rhoDP(p_ref, t_ref) / co2_rho; % CO2 compressibility
wat_c   = 0;                  % water compressibility
c_rock  = 4.35e-5 / barsa;    % rock compressibility
srw     = 0.27;               % residual water saturation
src     = 0.20;               % residual CO2 saturation
pe      = 5 * kilo * Pascal; % capillary entry pressure
muw     = 8e-4 * Pascal * second; % brine viscosity
muco2   = co2.mu(p_ref, t_ref) * Pascal * second; % co2 viscosity
mrstModule add ad-props;  % necessary module
mrstModule add ad-core;   % necessary module

% Load Johansen model
[G, rock, bcIx, ~, ~, bcIxVE] = makeJohansenVEgrid();

% Specify well information
wc = [3715, 10210, 16022, 21396, 26770]';
inj_rate = 3.5 * mega * 1e3 / year / co2_rho;
W = addWell([], G, rock, wc, ...
            'type', 'rate', ...  % inject at constant rate
            'val', inj_rate, ... % volumetric injection rate
            'comp_i', [0 1]);    % inject CO2, not water
```

Once the 3D grid and rock objects are available, we construct a corresponding grid and rock object of a type that can be used for VE simulation:

[5]For an interactive exploration of VE simulation applied on real datasets, the reader is referred to the tool `exploreSimulation` which is included with `MRST-co2lab` and introduced in Appendix C.

```
% Make top surface grid used for VE simulation
[Gt, G, transMult] = topSurfaceGrid(G);

% Computing vertically averaged rock object
rock2D = averageRock(rock, Gt);
```

The VE grid is returned in the variable `Gt`. It consists of a 2D grid representing the top surface of the aquifer, with additional embedded information about column heights and mappings back to the original 3D grid. Each cell in `Gt` thus represents a whole column in `G`, as illustrated in Figure 3.4. The vector `transmult` contains multipliers that modify certain face transmissibilities of `Gt` (c.f. Section 2.2.4) in the presence of partially overlapping faults in the 3D model. The return value from the call to `averageRock` is a vertically averaged version of the 3D rock object `rock`, i.e., column-wise averaged permeability and porosity values.

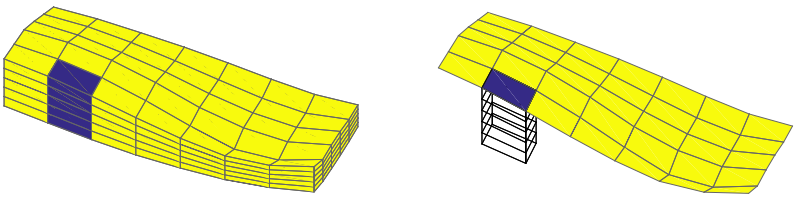


Figure 3.4: A 3D simulation grid (left) and the corresponding 2D grid generated by `topSurfaceGrid` (right). Each cell in the 2D grid corresponds to an entire column of cells in the 3D grid, as indicated in blue for one example column. This mapping is explicitly stored in the 2D grid, as indicated in wireframe on the right plot.

We can visualize the 2D top surface grid by plotting it on top of the original 3D grid. To ensure visual separation of the grids, we introduce a shift of 100 meters in the vertical coordinate of (a copy of) the 3D grid:

```
% Shift G 100 meters down, and plot both grids for comparison
GG = G; % use temporary copy, to avoid modifying G below
GG.nodes.coords(:,3) = GG.nodes.coords(:,3) + 100; % shift grid downwards
figure;
plotGrid(GG, 'facecolor', 'green'); % visualize 3D grid
plotGrid(Gt, 'facecolor', 'red');   % visualize top surface grid
view(-65,33);
set(gcf, 'position', [531 337 923 356]); axis tight;
```

The resulting plot is shown on Figure 3.5. The next step is to define initial aquifer state. Our VE model will use the top surface (caprock) as reference depth, so we set hydrostatic pressure accordingly:

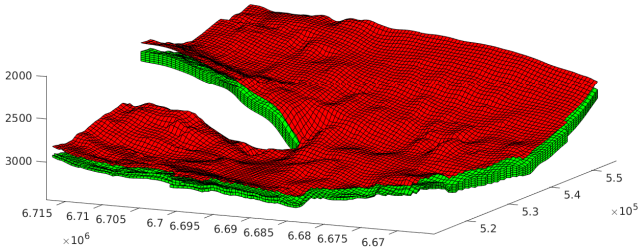


Figure 3.5: The Johansen simulation 3D grid (green) and corresponding top surface grid (red).

```
% Setting initial reference pressure.  Gt.cells.z gives the caprock
% depth of each cell in the 2D grid.
initState.pressure = rhow * g(3) * Gt.cells.z;

% Initial saturation is 1 for brine and 0 for CO2 in all cells.
initState.s = repmat([1, 0], Gt.cells.num, 1);
initState.sGmax = initState.s(:,2);
```

A VE simulation requires a special fluid object that supports the associated upscaled properties (i.e., saturation, mobility and capillary pressure). The construction of such an object is taken care of by the `MRST-co2lab` function `makeVEFluid`. Since we intend to run a simulation that includes the effect of a capillary fringe (c.f. the discussion of extended models below), we also need to provide the fine-scale capillary pressure and relative permeability functions. In the code listing below, these are specified to be equivalent to those used in the 3D simulation. However, `makeVEFluid` requires the *inverse* of the capillary pressure function (i.e., saturation as a function of capillary pressure):

```
% Specifying fine-scale relative permeability and (inverse of)
% capillary pressure curve, needed to compute the capillary fringe.
invPc3D = @(pc) (1-srw) .* (pe./max(pc, pe)).^2 + srw;
kr3D = @(s) max((s-src)./(1-src), 0).^2; % uses CO2 saturation

% Constructing fluid object suitable for a VE simulation
fluid = makeVEFluid(Gt, rock, 'P-scaled table'     , ...
                    'co2_mu_ref'   , muco2         , ...
                    'wat_mu_ref'   , muw           , ...
                    'co2_rho_ref' , co2_rho        , ...
                    'wat_rho_ref' , rhow           , ...
                    'co2_rho_pvt' , [co2_c, p_ref] , ...
                    'wat_rho_pvt' , [wat_c, p_ref] , ...
                    'residual'     , [srw, src]    , ...
                    'pvMult_p_ref', p_ref          , ...
                    'pvMult_fac'   , c_rock        , ...
                    'invPc3D'      , invPc3D       , ...
                    'kr3D'         , kr3D          , ...
                    'transMult'    , transMult);
```

Here, a "P-scaled table" is used to model the impact of the capillary fringe. Details are provided in Paper IV.

In order to construct the schedule, we must first convert the well specified on the 3D grid to a version that can be used in the VE simulation, using the function `convertwellsVE`:

```
W2D  = convertwellsVE(W, G, Gt, rock2D);
```

We then proceed to define the hydrostatic pressure conditions and the rest of the schedule, in a way very similar to what was done in the 3D case in Chapter 2 (`bcIxVE` contains indices to the relevant boundary faces in the top surface grid):

```
% Hydrostatic pressure conditions for open boundary faces
p_bc = Gt.faces.z(bcIxVE) * rhow * g(3);
bc2D = addBC([], bcIxVE, 'pressure', p_bc);
bc2D.sat = repmat([1 0], numel(bcIxVE), 1);

% Setting up two copies of the well and boundary specifications.
% Modifying the well in the second copy to have a zero flow rate.
schedule.control    = struct('W', W2D, 'bc', bc2D);
schedule.control(2) = struct('W', W2D, 'bc', bc2D);
schedule.control(2).W.val = 0;

% Specifying length of simulation timesteps
schedule.step.val = [repmat(year, 100, 1); ...
                     repmat(10*year, 100, 1)];

% Specifying which control to use for each timestep.
% The first 100 timesteps will use control 1, the last 100
% timesteps will use control 2.
schedule.step.control = [ones(100, 1); ...
                         ones(100, 1) * 2];
```

Finally, we specify the complete model and run the simulation. `CO2VEBlackOilTypeModel` represents a VE model based on the s-formulation. This model also supports instant or rate-driven dissolution of CO_2 into brine, although we do not use that here:

```
% Constructing the complete VE model
model = CO2VEBlackOilTypeModel(Gt, rock2D, fluid);

% Run simulation
[wellSol, states] = simulateScheduleAD(initState, model, schedule);
```

After the simulation has completed, we can inspect the results in a similar way as for the 3D case in Section 2.2.4. The returned saturation values represent the upscaled (vertically integrated) variables, and can be plotted directed on the top surface grid `Gt` using the `plotCellData` function. However, to allow for a better direct comparison with the 3D solution in Section 2.2.4, we here choose to map the upscaled saturation values back to the cells in the 3D grid, using the function `height2Sat`:

```
% Plotting CO2 saturation for timestep 100 (100 years after start)
tstep = 100; % timestep 100, i.e., after 100 years of injection
[h, h_max] = ...
   upscaledSat2height(states{tstep}.s(:,2), states{tstep}.sGmax, Gt, ...
                      'pcWG', fluid.pcWG, ...
                      'rhoW', fluid.rhoW, ...
                      'rhoG', fluid.rhoG, ...
                      'p', states{100}.pressure);
plotCellData(Gt.parent, ...
             height2Sat(struct('h', h, 'h_max', h_max), Gt, fluid));
colorbar; view(-63, 68); set(gcf, 'position', [531   337   923   ...
    356]); axis tight;
```

The conversion from upscaled saturation to fine-scale saturation in the 3D grid is carried out in two steps. First, the function `upscaledSat2height` converts the upscaled saturations (current and max) into the equivalent plume thickness values, i.e., the h-formulation discussed in Section 3.1. Then, the height values are converted into fine-scale saturation values for the 3D grid using the function `height2Sat`. Finally, the result is plotted on the 3D grid using `plotCellData`[6]. By carrying out the code snippet above using `tstep = 200`, we also plot the result for the last time step, i.e., after an additional 1000 year of migration. The results for both cases are shown on Figure 3.7.

It is instructive to compare Figure 3.7 with Figure 2.10. Some similarities can be noted, but also some important differences. In both plots the plume shape and extent is similar at the end of injection, but with a more diffuse outer boundary in the case of the VE formulation. The primary reason for this has to do with the limited vertical grid resolution in the 3D model, which is not capable of resolving the thin CO_2 tongue that rapidly spreads out beneath the caprock (this is also apparent by looking at Figure 3.1). The VE model is not limited by vertical resolution and is therefore able to model the spread of this tongue. During the injection phase (up to and including timestep 100), fluid flow is primarily driven by the pressure gradient induced by the injection well, and most of the CO_2 plume remain relatively thick. In contrast, during the ensuing migration phase, flow is primarily gravity driven, and migration takes place almost exclusively in the form of a thin layer of CO_2 along the top of the aquifer. In this case, the low vertical resolution of the 3D model severely impacts the final result. This is clearly visible by comparing the lower plots of Figure 2.10 (repeated for convenience here in Figure 3.6) and Figure 3.7. On the latter, the CO_2 plume has spread out much more extensively.

Extended models

The VE modeling framework is capable of including a large number of physical effects influencing migration, not included in the simple VE model described by (3.7) above. With `MRST-co2lab`, most of the effects mentioned below can be modeled within the same fully implicit framework.

[6] The original 3D grid is referenced by the top surface grid `Gt`, and can be accessed using `Gt.parent`.

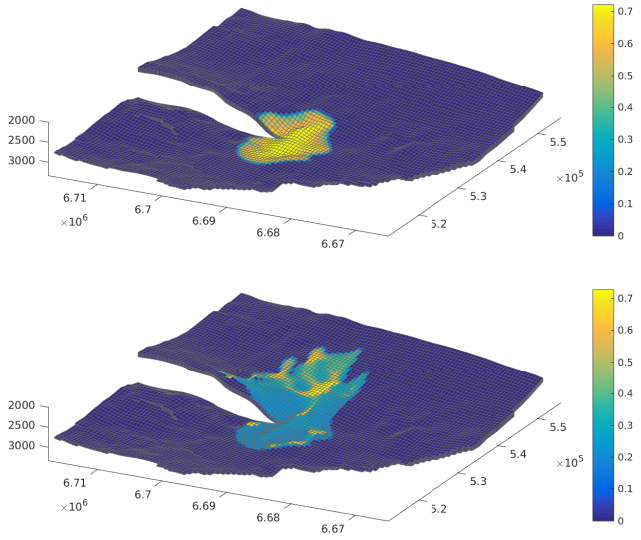


Figure 3.6: CO_2 saturation in the Johansen injection scenario, computed using the 3D simulation model (repeated from Figure 2.10). *Top:* After 100 years (end of injection). *Bottom:* After 1100 years (end of simulation).

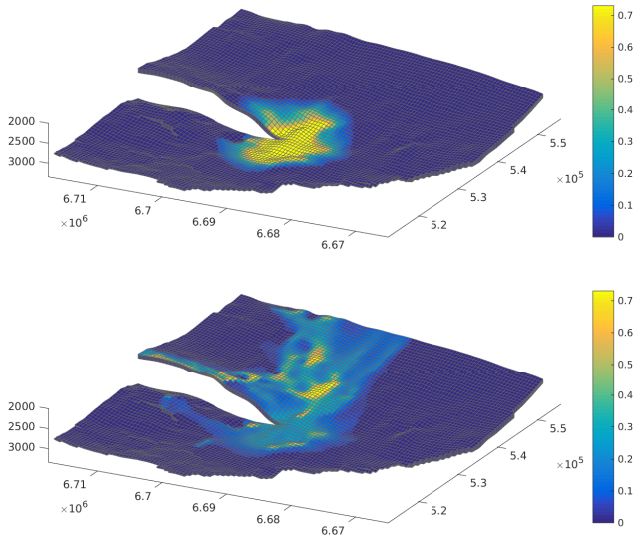


Figure 3.7: CO_2 saturation in the Johansen scenario, computed using the VE model. *Top:* after 100 years (end of injection). *Bottom*: after 1100 years (end of simulation).

Fine-scale capillarity

An important effect not described by the sharp interface model presented above is the impact of fine-scale capillary pressure on the CO_2 migration. A VE formulation includ-

ing this effect was presented in detail and discussed in [81]. Under the VE assumption, the consequence of non-negligible capillary pressure is the presence of a transition zone of finite thickness (capillary fringe) between the CO_2 and brine phases. In other words, the vertical CO_2 saturation profile will no longer be discontinuous, but will transition smoothly from the CO_2 endpoint saturation within the plume region to zero in the brine phase below. A review of capillary pressure/saturation relationships used in CO_2 storage modeling literature was used to estimate the corresponding capillary fringe thicknesses in [27]. The paper shows that in a large majority of the studied cases, the transition zone would extend over a significant part (> 0.1) of the total aquifer thickness. Thus, a sharp interface cannot in general be assumed. The presence of a capillary fringe will not change the VE equations (3.7) themselves, but will lead to more complicated expressions for upscaled saturation and mobilities (3.8)-(3.17). In particular, the handling of residual saturation becomes more complex, and will require a model for hysteresis at the fine scale [34, 35]. This and other computational issues related to the practical modeling of capillarity within the VE framework is discussed in detail in Paper IV. An estimate of the time scale required for the capillary fringe to form is provided in [81]. Naturally, for a VE model that includes a capillary fringe to remain valid, this time scale must be small compared to the time scale studied when using the model for simulation.

CO_2 dissolution

For CO_2 storage, *dissolution of CO_2 into brine* is another potentially very important effect. Simulation studies indicate that under the right circumstances, dissolution may immobilize most of the injected CO_2 within a few centuries [44]. The dissolution of CO_2 into brine is fast when the fluids are in immediate contact with each other. However, the brine in contact with the CO_2 plume will quickly saturate, bringing further dissolution to a halt. Transport of dissolved CO_2 away from the CO_2-brine interface is a diffusion-controlled, extremely slow process [38]. Brine saturated with CO_2 is denser than the underlying, unsaturated brine. This creates a gravitationally unstable system which, after an initial incubation time, may lead to the establishment of convection cells in the brine phase. Within these cells, heavy, saturated brine flows downwards, and is replaced by unsaturated brine that rises towards the CO_2 region. This process, known as *convective mixing*, establishes a downward transport of dissolved CO_2 away from the CO_2-brine interface at a rate that can be considered roughly constant in time [38]. Within the VE framework, this process can be modeled as CO_2 dissolving into the brine phase at a constant rate for a given vertical pillar. A VE model that includes the effect of dissolution was initially presented in [45], which also uses this model to simulate a migration scenario for the Johansen formation. The model was also used in simulation studies for the same formation in [44] and [46]. A black-oil formulation of the VE equations that includes dissolution is presented in Paper III and implemented in `MRST-co2lab`. It is used to demonstrate the potential impact of dissolution on long-time migration in the Utsira formation in Paper V and Paper VII, and for the Skade aquifer in Paper VI.

Although dissolution has the potential to play a key role for the trapping of CO_2 in storage scenarios, its practical impact will depend on the onset and strength of convective mixing, which is hard to estimate in advance. Some theoretical studies indicate

that onset of convective mixing occurs in less than a year for "typical systems" [45]. However, depending on the parameters of the system, the onset time may vary by several orders of magnitude, from days to thousands of years [54]. In addition, evidence from the study of a natural CO_2 storage analogue, the Bravo Dome field in New Mexico, suggests that convective mixing, while real, may be too slow to play any significant role in CO_2 storage for sites with similar geology [95]. On the other hand, observations from the on-going CO_2 storage operations at the Sleipner site off the coast of Norway suggest that significant CO_2 dissolution is indeed taking place there [33].

Compressibility

The VE formulation in (3.7) was derived under the assumption of constant fluid density. However, in reality, the density of CO_2 is highly dependent on temperature and pressure, and significant spatial and temporal variability is to be expected for realistic CO_2 storage scenarios. This is particularly true for conditions similar to those found at Sleipner, where aquifer temperature and pressure are considered to be close to the liquid-vapor critical point of CO_2 [21]. For this reason, a VE formulation used to describe fluid flow in general CO_2 storage scenarios would need to account for variable fluid densities. The inclusion of variable density will lead to a more complicated VE formulation than (3.7). First, a new upscaled variable for density needs to be introduced. Moreover, since fluid density and fluid-static pressure fields are interdependent, the computation of upscaled density, as well as the volumetric (or mass) fluxes, will involve numerical resolution of ordinary differential equations. The technical details are presented in Paper I, which also assesses the resulting impact and computational aspects. This paper concludes that for most cases, the impact of vertical variations in density remains low and could be neglected. However, lateral density variations may still be significant and should be captured. For this reason, the use of a simplified pressure model that includes only lateral density variations is suggested. Under this model, fluid densities are considered constant within each vertical column. This simplified treatment of density was previously employed in [41], although the formal analysis and justification of neglecting vertical density variation was not addressed in depth.

Subscale caprock undulations

Since long-term CO_2 migration is primarily gravity driven, taking place in the form of thin CO_2 plumes flowing beneath a confining caprock, the topography of this confining layer plays a central role for determining the long-term migration pattern. Caprock topography may direct the flow along certain channels, or slow down migration by retaining some of the migrating CO_2 in structural traps of various size and shape along the way. When these traps are below grid resolution, the associated effect on CO_2 flow cannot be explicitly captured by the geometry of the simulation model, but has to be accounted for in an upscaled manner. The practical impact on the VE formulation is modification of upscaled permeability and relative permeability. Different models for upscaling and assessment of practical impact has previously been addressed in [42] and [43]. In general, it is shown that the presence of subscale trapping leads to slower migration, self-sharpening in the tip of the advancing CO_2 plume, and additional structural trapping. In Paper V, an attempt is made to estimate the amount of subscale trapping

for the Utsira aquifer, expressed as a function of local aquifer tilt and based on the availability of datasets with different resolution.

Regions with significant vertical flow

In areas where significant vertical flow is expected, in particular in the vicinity of an injecting (or extracting wells), the VE assumption no longer holds. Different models must be used to capture the effect of vertical flow in these areas. In such cases, VE models could still be used to represent flow at the large scale, coupled to more exact models that can capture the effect of vertical flow in the regions where it cannot be neglected. One idea is to employ full, numerical 3D simulation in regions around injection wells, where the outer boundary of the 3D simulation is positioned far enough from the well so that pressure equilibrium is practically observed. At this distance, a VE model formulation using a vertically integrated grid could take over. Such a solution would nevertheless require computationally costly 3D simulations (albeit of reduced domain size) with high vertical resolution, as discussed at the beginning of this chapter. Moreover, within a typical large-scale simulation domain, there could be a large number of wells (hundreds, in some areas of the world). Less costly alternatives around wells than full 3D simulations have however been proposed. In the VESA model presented in [41], a hybrid numerical-analytical approach is employed, where a vertical equilibrium model at the aquifer scale is coupled with analytical solutions around wells. In [48], a dynamic reconstruction model is proposed where the upscaled governing equations are still obtained by vertical integration, but without the assumption of vertical equilibrium. Instead, the fine-scale pressure and phase distributions required to compute the upscaled variables are reconstructed taking vertical dynamics into account, which for each timestep requires solving a 1D flow problem on each vertical pillar. This approach has the advantage to apply not only on areas associated with vertical flow around wells, but also on regions where the VE assumption might be inappropriate at the considered timescale for other reasons, such as low vertical permeability and/or low density contrast between CO_2 and brine. The dynamic reconstruction approach was later extended to model layered formations, initially in [49] and further developed in [50].

Diffuse leakage of brine through caprock

Ideally, capillary exclusion will prevent CO_2 from entering the caprock and migrate upwards out of the aquifer. However, the overpressure in the aquifer induced by CO_2 injection may cause diffuse leakage of brine into the caprock or lower confining layer. While the movement of brine through the low-permeability caprock will be very slow, the impact on the aquifer pressure field may be significant [80]. The *inclusion of diffuse leakage* can be modeled in a VE setting by introducing sources and sink terms in the VE-formulation, associated with regions where mobile brine is in contact with the confining layers. Leakage rates can be estimated using Darcy's law, by considering the pressure differential across the confining layer (aquitard). In many cases, the flow through the aquitard can be modeled as vertical, and therefore one-dimensional[7], which lowers the cost of computation.

[7]The dynamic reconstruction framework presented in [50] discusses how the proposed model could also be used to include lateral flows within aquitards, if considered significant.

Thermal effects

In general, CO_2 injected into a geological formation will not be in thermal equilibrium with its surroundings. This is the case both for the Sleipner, In-Salah and Snøhvit injection operations [37]. The injection of cold, liquid CO_2 has also been considered as an attractive storage option, for operational and economic reasons [109]. In many cases, the inclusion of thermal modeling may be necessary, since an overly inaccurate representation of temperature may cause significant errors in fluid densities and viscosities, which again may lead to inaccurate predictions of plume footprint and temporal evolution. The inclusion of *thermal effects* within the VE framework has been investigated in [40]. In the model presented therein, energy conservation is modeled using a simplified heat transport model where heat convection only takes place within the (vertically integrated) CO_2 plume, whereas heat conduction happens both in brine, CO_2 and rock. As pointed out in the article, certain simplifications made may lead to significant temperature error in some cases. In particular, a simplified but accurate representation of heat transfer between the CO_2 plume and its surroundings remains a subject of further investigation.

3.2 Spill-point analysis

3.2.1 Motivation and general description

The shape and slope of the confining surface plays a significant role in directing long-term CO_2 migration, as can be noticed on the plots from our Johansen example in Figure 3.7. In the absence of background flow, long-term CO_2 migration is driven primarily by gravity and capillary forces, and will make the plume spread out as a thin layer and migrate upwards along the steepest gradient under the confining layer(s). Along the way, CO_2 will accumulate in any structural trap encountered and may also temporarily collect in flat areas where migration can slow down almost to a halt due to vanishing gravity drive. These phenomena of local accumulation are exemplified by the yellow spots that indicate heightened CO_2 saturation on the lower plot of Figure 3.7.

Assuming that CO_2 migration is predominantly driven by gravity and capillary forces, it is possible to predict the path taken by long-term migration of CO_2 solely by analysis of the topography of the confining layers (of which there may be more than one, e.g., in the case of intra-aquifer layers/baffles of low permeability as observed at Sleipner [20]). Starting from a given point of origin, a migration path can be traced out by following the locally steepest upward gradient of the confining surface at all times. Occasionally, this path may lead to a local maximum in the top surface elevation, which signifies the presence of a structural trap where some CO_2 will accumulate indefinitely (or at least until eaten away by dissolution or other long-term processes). A topographical analysis can be used to determine the location and volumes of all such traps, as well as the locations of their *spill-points*. The spill-point of a trap indicates where CO_2 will spill over and flow out once the trap has been completely back-filled. When tracing out the upwards path taken by CO_2 from some given location, the presence of a structural trap signifies that the upward path will resume from its spill-point once the trap has been completely filled. In this manner, a migration path can pass through a number of structural traps before eventually reaching an external boundary. The process is

illustrated in Figure 3.8.

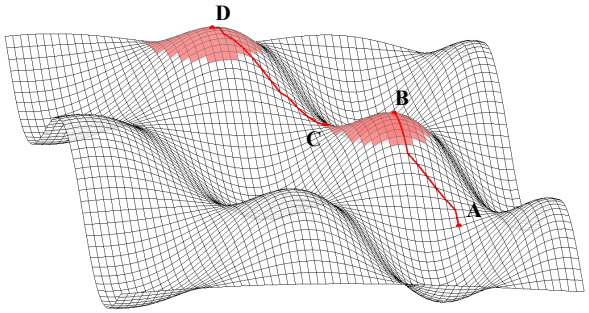


Figure 3.8: Synthetic example illustrating the path taken by gravity-driven migration of CO_2 below a confining caprock. CO_2 originates from **A**, flows uphill until it reaches a local maximum **B**, at which point it starts to accumulate in the associated structural trap. When the trap is filled up to point **C**, CO_2 spills out and migrates further upwards until a new maximum **D** is reached.

By specifying the total amount of CO_2 injected, as well the capillary entry pressures for confining layers and other potential barriers, the final distribution of CO_2 in structural/stratigraphic traps of the aquifer can be determined by a largely geometrical reasoning. Estimates on possible losses from other trapping mechanisms (residual trapping, rate-driven resolution) can also be taken into account. In the context of basin modeling, this type of analysis is referred to as invasion percolation, and is typically used to study regional hydrocarbon migration at geologic time scales [104, 105]. The study of long-term CO_2 migration (thousands of years) can be thought of as an intermediary between reservoir modeling (Darcy flow, human timescales) and the study of hydrocarbon migration in basin modeling (percolating flow, millions of years). As such, it is natural to consider the use of invasion percolation-type analyses also for CO_2 migration scenarios. One recent example of employing software developed for basin modeling to the context of CO_2 storage for a specific formation offshore Norway is given by [67], where results are compared with corresponding outcomes from reservoir modeling, with reasonable match. On a more general level, the potential impact of top surface morphology on structural trapping capacity and CO_2 migration under different geological assumptions was studied using stochastically generated ensembles in [77].

`MRST-co2lab` provides a simplified invasion-percolation functionality, where gravity-driven CO_2 migration under an impermeable caprock can be deduced by geometric analysis, here referred to as *spill-point analysis*. Computationally, this type of modeling is inexpensive compared to a full numerical simulation (whether or not based on vertical equilibrium). As will be demonstrated in Section 3.2.3, a complete topographical study of a caprock with 100,000 cells can be carried out in a couple of seconds, providing not only the location and sizes of all structural traps, but also all required information to immediate identify the migration path from any given location and the amount of structural trapping reached along the way. Details on the algorithms used and demonstration on real formation models is presented in Paper II.

When studying the post-injection phase, spill-point analysis tends to provide predictions that compare well with results from numerical simulations. It also gives estimates

on the size and location of structural traps, and how they can be reached. During a project planning phase, this information can be used to guide the placement of injection wells to maximize the structural trapping that can ultimately be exploited.

For certain types of conditions (high aquifer permeability, strong gravity segregation, low injection overpressure), the use of invasion-percolation tools may also be a useful tool for modeling plume development during the injection phase. This is the case of the Sleipner CO_2 storage site, where a study comparing measured data with simulation runs on a benchmark model of its upper layer concluded that a simulation approach based on invasion percolation best matched plume shape observations away from the injection point, outperforming black-oil and compositional models [99]. Later studies, which also include modeling of the multiple underlying intra-aquifer shale barriers, confirm the validity of the invasion percolation approach, both by theoretical discussion and in terms of the ability to match real observations [20, 21].

The remaining part of this section will present general concepts and terminology associated with the spill-point analysis as it is applied in `MRST-co2lab`, and provide some examples using the Johansen model as well as the Utsira aquifer. The explicit mapping of structural traps in the Johansen caprock will also permit the construction of a *trapping inventory* of the injected CO_2 in the Johansen injection scenario simulated in the previous section. In this context, a trapping inventory is a graph tracking the CO_2 trapping state as a function of time.

3.2.2 Basic concepts; analogy with surface water hydrology

The use of spill-point analysis to predict CO_2 migration patterns has a strong analog in surface water hydrology, where flow of precipitated water in a terrain is organized into a hierarchical system or rivers and lakes determined by local topography. In many ways, the system describing gravity-driven CO_2 flow could be interpreted as an inverted surface water system, where flow is directed upwards rather than downwards (Figure 3.9). In the following text, this analog is used to illustrate the introduced concepts, which are all presented under the assumption of a completely impermeable confining surface and gravity-driven, infinitesimal flow.

A *spill path* is the upward path followed by migrating CO_2 from any specified location in the aquifer. It follows the locally steepest ascent at every point. The path includes the structural traps encountered along the way upwards until the boundary of the domain under study is reached. In the analog with hydrology, a spill path thus roughly corresponds to a sequence of rivers and lakes.

A *spill region* consists of all points on the surface from which flow will be directed towards a common local maximum, or set of local maxima above the region's *spill-point*. The spill-point of the region is the highest point on the region's boundary. All points of the spill region above this point (which will include the local maximum or set of maxima) is referred to as a *structural trap*. Any CO_2 flowing into such a trap will start back-filling the trap until the spill-point is reached, at which point any additional CO_2 entering the trap will lead to an equivalent amount spilling out from the trap at the spill-point location and into the neighboring spill region. In other words, all CO_2 entering a spill region will be funneled towards and eventually exit from its spill-point once the associated structural trap has been completely filled (see Figure 3.8). In hydrology, the analogy to a spill region would be a catchment area, a structural trap would cor-

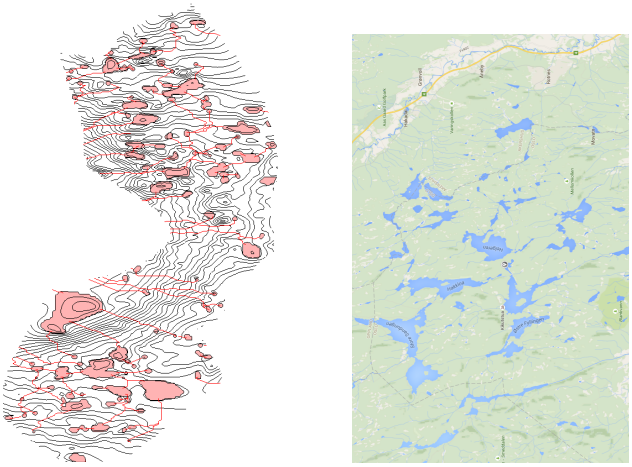


Figure 3.9: Plots illustrating the analogy between surface water hydrology and spill-point analysis for modeling CO_2 migration. *Left:* A topographical map of the Utsira aquifer top surface, with structural traps and spill paths indicated. *Right:* A map of a section of the woodlands outside Oslo (Map data: Google Maps). A similar structure can be observed between the plots.

respond to a lake, and the spill-point to a lake outlet. The boundary of a spill region would correspond to a drainage divide.

Since infinitesimal flow is considered, a spill region should in most cases have one unique spill-point. In degenerate cases it might have two or more (when more than one point at its boundary share the same, highest elevation value). In such situations, flow out of the spill region cannot be uniquely determined. Due to inevitable imprecision in the data describing the confining surface, this problem would arise in practice also for cases where there is an unique spill-point, but where the boundary of the spill region contains at least one other point whose elevation value approaches that of the spill-point, within data uncertainty.

Using the vocabulary of graph theory, the system of structural traps (with their associated spill regions) and spill paths connecting them form a directed graph with no cycles, i.e., a *tree*. This is referred to as a *spill tree*, which describes a hierarchical relationship between traps that spill into each other, as illustrated in Figure 3.10. A confining surface might have more than one spill tree, which would represent disjoint flow systems.

The analogy between surface water hydrology and CO_2 spill-point analysis could be further extended by comparing surface water evaporation to CO_2 dissolution in a geological system, in the sense that both represent a gradual, diffuse depletion over time. The spill-point analysis tools in `MRST-co2lab` do not, however, take dissolution into account.

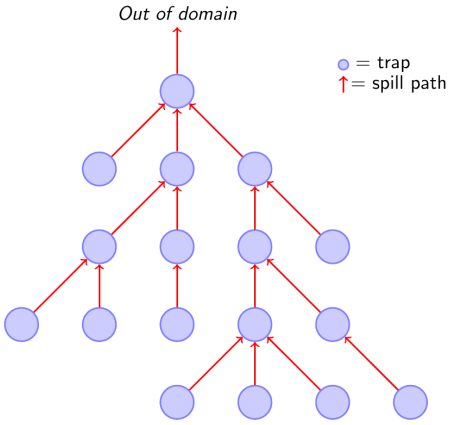


Figure 3.10: A schematic illustration of a spill tree, consisting of structural traps connected by spill paths.

3.2.3 Spill-point analysis using `MRST-co2lab`

The Johansen simulation grid used in the previous examples is relatively coarse, with limited topographic detail in its top surface. We will therefore start out by analyzing a different grid representing the Utsira formation, based on data provided by the Norwegian Petroleum Directorate [51][8]. In its uncoarsened form, the top surface of this grid has a spatial resolution of 500 x 500 meters, with a total of 98356 cells. The `getFormationTopGrid` script can be used to generate top surface grids of various formations in the North Sea. We here use it to construct the Utsira grid:

```
% Construct and load the Utsira grid
coarsening_level = 1; % downsampling factor (1 - no downsampling)
Gt = getFormationTopGrid('Utsirafm', coarsening_level);
```

The returned grid, `Gt`, is a top surface grid similar to those obtained by applying `topSurfaceGrid` on an existing 3D grid, as used in the example in Section 3.1. It can thus also serve as a grid for vertical equilibrium simulations.

We visualize the top surface using the following code snippet, where color is used to indicate depth. To highlight the topographical variations, we include some simple lighting commands in the generation of these plots, whose results are shown in Figure 3.11. The complex nature of the topography is clearly visible on these plots.

[8] Adding to the demonstration in the present section, an interactive tool for applying spill-point analysis on North Sea formations is provided in `MRST-co2lab` and presented in Appendix C of this dissertation.

```
% Visualize the Utsira grid, zenith view
plotCellData(Gt, Gt.cells.z, 'edgecolor', 'none')
view(90, 90); axis tight;
c = colorbar;  c.Label.String = 'Depth (m)';
set(c, 'fontsize', 20); set(gca, 'fontsize', 20);
set(gcf, 'position', [0 0 1400 600]);
camlight headlight; material dull; % use lighting to enhance detail

% To get an inclined view, run the following line
view(80, 50);
```

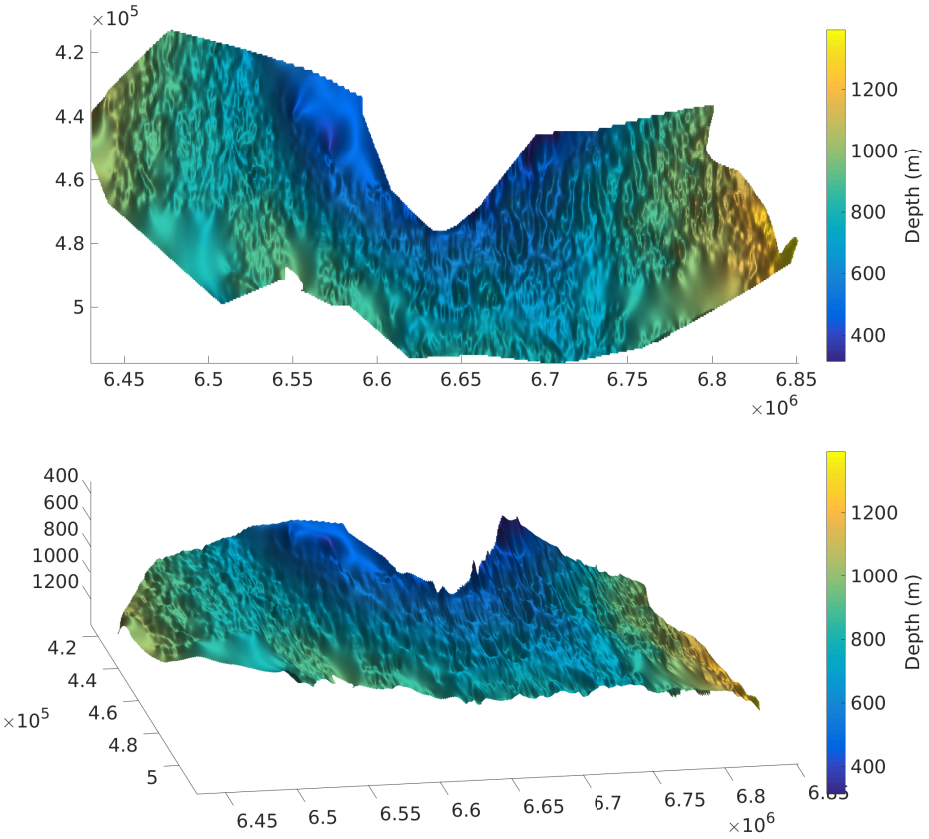


Figure 3.11: Top surface of the Utsira formation, viewed from above (upper plot) and inclined (lower plot). Color represents depth below sea level.

The `trapAnalysis` command performs a geometrical analysis that identifies all structural traps, spill regions, spill-points and spill trees. There are two flavors of this algorithm, depending on whether the geometry is to be analyzed in terms of nodes and edges or in terms of cell centers. We here apply the node-based variant:

```
% Compute trapping structure

% The second argument indicates node-based (as opposed to cell-based)
% interpretation of geometry
ta = trapAnalysis(Gt, false)
```

```
  ta =
            traps: [98356x1 double]
          trap_z: [193x1 double]
    trap_regions: [98356x1 double]
        trap_adj: [193x193 double]
      cell_lines: {1x193 cell}
             top: [266x1 double]
```

The `traps` field is an integer vector whose components correspond to each cell in `Gt`. A zero value indicates that the corresponding cell does not belong to a structural trap. Otherwise, the component value will be a positive integer representing the index of the trap that the cell belongs to. A similar representation is used for the `trap_regions` field, which indicates the index of the trap region each cell belongs to (if any). The `trap_z` field indicates the spill-point depths for each structural trap (in this example, there are 193 of them). `trap_adj` is a connectivity matrix representing the spill trees, whereas `cell_lines` indicates which cells belong to rivers between traps. All local maxima are listed in the `top` field. Since each structural trap can have more than one local minima, there are usually more components to this vector than to the `trap_z` vector (in our case 266 vs. 193).

The `traps` field can be used directly to visualize the location and shapes of all structural traps:

```
% Visualize structural traps of the Utsira grid, zenith view
plotCellData(Gt, ta.traps, 'edgecolor', 'none');
view(90, 90); axis tight;
colormap lines; % we choose a colormap that is not gradient-based,
                % since each trap is a discrete entity.
set(gca, 'fontsize', 20);
set(gcf, 'position', [0 0 1400 600]); camlight headlight; material dull

% To get an inclined view, run the following line
view(80, 50);
```

From the resulting plots shown in Figure 3.12, a large number of structural traps of varying sizes can be seen, apparently spread out evenly across the formation. Only the spatial distribution can be appreciated from this plot; it does not provide information on trap capacities, associated spill regions or topological connections. To visualize the spill regions and rivers connecting traps, we can use a more complex plotting script (result shown in Figure 3.13):

```
% Visualize traps, spill regions and rivers

% Construct color field identifying traps
trapfield = ta.traps;
trapfield(trapfield==0) = NaN; % set non-trap cells invisible
```

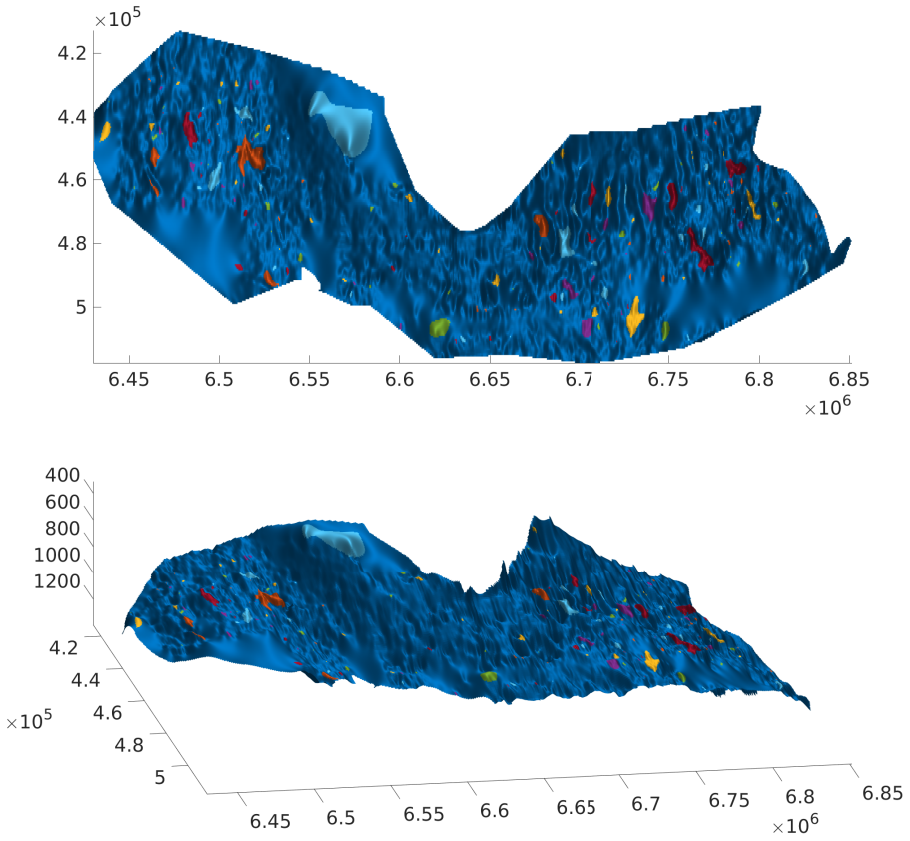


Figure 3.12: Top surface of the Utsira formation, with structural traps indicated. Colors are used to distinguish separate traps, but do not convey any further information.

```
% Construct color field identifying rivers
river_field = NaN(size(ta.traps));
for i = 1:numel(ta.cell_lines) % loop over traps and rivers
   rivers = ta.cell_lines{i};
   for r = rivers
      river_field(r{:}) = 1; % make river cells visible
   end
end

% Plot traps in solid color
clf; plotCellData(Gt, trapfield, 'edgecolor', 'none');

% Plot regions in semi-transparent color
plotCellData(Gt, ta.trap_regions, 'facealpha', 0.5, 'edgecolor', 'none');

% Add rivers
plotCellData(Gt, river_field, 'edgealpha', 0);

% Top view
```

```
view(90, 90); axis tight; colormap lines;
set(gca, 'fontsize', 20);
set(gcf, 'position', [0 0 1400 600]); camlight headlight; material dull

% To get an inclined view, run the following line
view(80, 50);
```

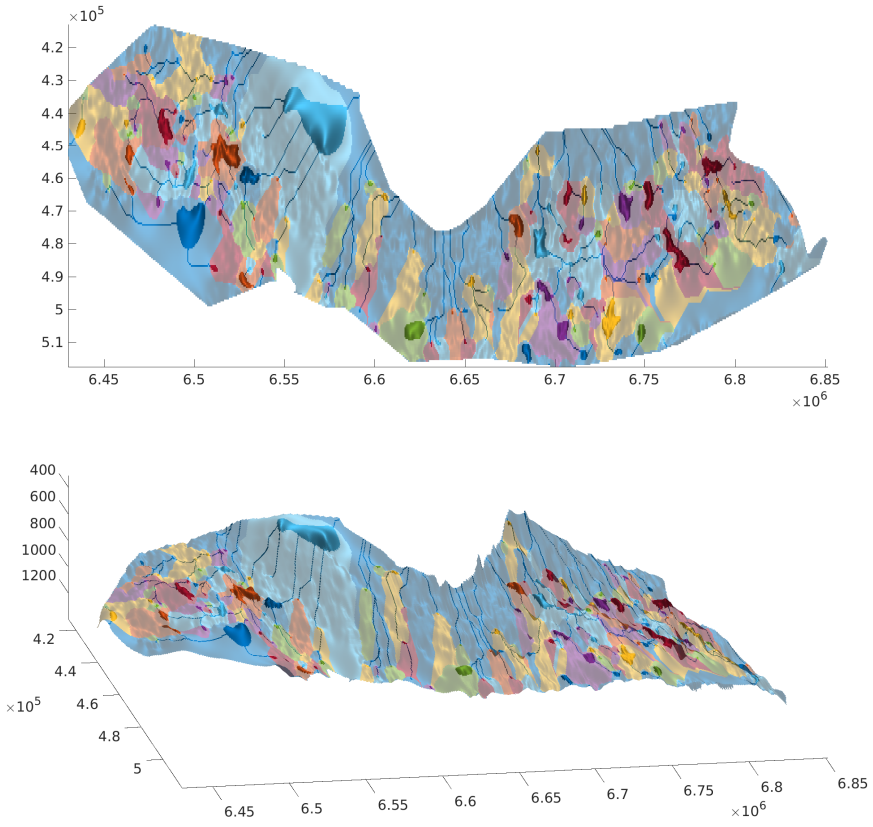


Figure 3.13: Top surface of the Utsira formation: structural traps, spill regions and spill paths.

Each trap is here shown with associated spill region in a fainter color. It now becomes clear what trap, if any, CO_2 injected in a particular location will flow into, and what path will be followed thereafter once the trap is filled (assuming gravity-driven migration).

The spill tree can also be visualized as a topographic map, which is perhaps easier to read for a visual examination of the spill tree system (resulting plot in Figure 3.14):

```
% Topographical map plot indicating traps and rivers
h = figure;
mapPlot(h, Gt, 'traps', ta.traps, 'rivers', ta.cell_lines);
set(gca, 'fontsize', 20);
set(gcf, 'position', [0 0 1400 600]); view(90, 90);
```

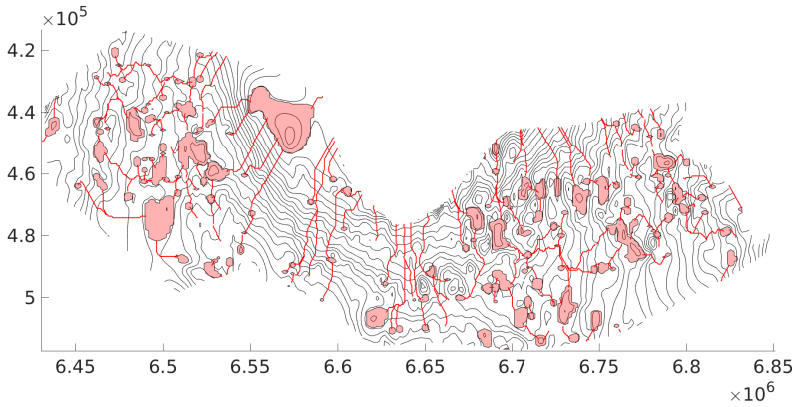


Figure 3.14: Topographic map of Utsira top surface, with traps and spill paths indicated.

As mentioned earlier, the `trapAnalysis` algorithm identified at total of 193 structural traps in the Utsira top surface. Moreover, previous plots indicate that the traps are evenly spread out across the aquifer. However, the storage capacity of each individual trap remains unclear. The ability of a trap to store CO_2 not only depends on its lateral extent, but also on its depth, the local porosity of the rock, and the difference in depth between its spill-point and interior top surface. The pore volume of a trap is relatively easy to compute. To also compute the capacity in mass terms, additional assumptions on CO_2 density, which varies according to local pressure and temperature, must be made. This is shown in the following code:

```
% An initial entry added to vector of spill-point depths,
% for indexing purposes (explained below)
depths = [0; ta.trap_z];

% The line below gives the correct "trap height" value for cells
% within traps.  Since we have added an entry to the front of the
% 'depths' vector, we increase indices by 1 (necessary as MATLAB
% arrays are indexed from 1, not 0).
h = min((depths(ta.traps+1) - Gt.cells.z), Gt.cells.H);
h(ta.traps==0) = 0; % cells outside any trap have zero trap volume

% Bulk cell volumes simply obtained by multiplying by respective
% areas. Accumulating individual cell volumes provides trap volumes.
cell_tvols = h .* Gt.cells.volumes;
tvols = accumarray(ta.traps+1, cell_tvols);

% The first entry of 'tvols' contain the combined trap volume of all
% non-trap cells.  Obviously, this value should be zero.
assert(tvols(1)==0);
tvols = tvols(2:end);

% Parameters necessary to compute pore volume and CO2 density
porosity       = 0.36;
seafloor_temp  = 7 + 273.15; % seafloor temperature, degrees Kelvin
seafloor_depth = 80;   % in meters
```

```
temp_grad      = 35.6; % thermal gradient (deg. Kelvin/kilometer)
rho_brine      = 1000; % brine density (kilogram per cubic meter)

% Computing temperature field and hydrostatic pressure
T = seafloor_temp + temp_grad .* (Gt.cells.z - seafloor_depth)/1000;
P = 1 * atm + rho_brine * norm(gravity) * Gt.cells.z;

% Making stripped-down fluid object with only CO2 density function
fluid = addSampledFluidProperties(struct, 'G');

% Computing local CO2 density (at caprock level) and trap capacity
% in mass term for each cell
CO2_density = fluid.rhoG(P, T);
cell_tmass = cell_tvols .* porosity .* CO2_density;

% Accumulating cell values to get mass trap capacity per trap
tmass = accumarray(ta.traps+1, cell_tmass);

% Bar plot where trap volumes shown in descending order
subplot(1,2,1);
bar(sort(tvols, 'descend'))
xlabel('Traps (sorted by volume)');
ylabel('Cubic meters'); set(gca, 'fontsize', 20);
title('Capacity (volume)');

% Bar plot with trap capacities in mass shown in descending order
subplot(1,2,2);
bar(sort(tmass/1e9, 'descend'));
xlabel('Traps (sorted by mass capacity)');
ylabel('Megatonnes'); set(gca, 'fontsize', 20);
title('Capacity in mass terms');
set(gcf, 'position', [0 0 1300 440]);
```

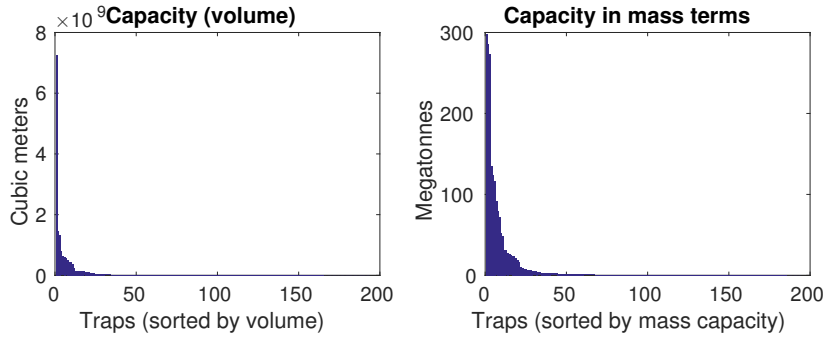


Figure 3.15: Trapping capacity (in volumetric or mass terms) of individual structural traps in the Utsira aquifer, sorted in descending order.

The bar plots generated at the end of this code snippet are shown in Figure 3.15. These plots indicate that there are a few very large traps, and a long tail of traps with negligible size. To optimize utilization of available trapping capacity, targeting the handful of large traps should be prioritized. To visualize them, we can plot traps colored according to total capacity in mass term (Figure 3.16):

```
% Plot traps according to capacity in mass terms (megatonnes)
figure;
plotCellData(Gt, tmass(ta.traps+1) / 1e9, 'edgealpha', 0.1);
view(90, 90); axis tight; colormap cool;
c = colorbar; c.Label.String = 'Megatonnes';
set(c, 'fontsize', 20); set(gca, 'fontsize', 20);
set(gcf, 'position', [0 0 1400 600]);
```

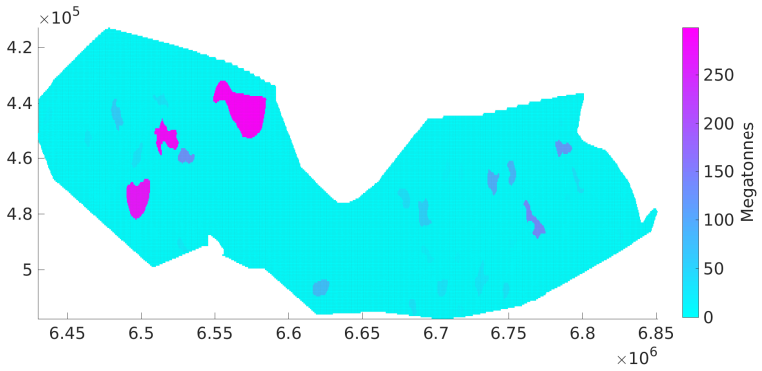


Figure 3.16: Structural traps in the Utsira aquifer, colored by capacity in mass terms.

From the resulting plot, it is clear that most of the identified structural capacity is concentrated in a handful of large traps in the southern part (left) of the aquifer, whereas the northern part provides only a modest amount of additional capacity. To exploit the capacity of these traps, it is of course possible to inject directly into them. However, by injecting somewhere farther downslope along the spill tree they belong to, additional traps could also be covered. To visualize this, we can generate a plot that indicates the total capacity reachable by gravity-driven migration from any given location in the aquifer. In other words, the generated plot will indicate the total capacity of the part of the spill-tree that is upslope from any given point:

```
% Make a plot of reachable structural capacity

% The first entry of 'tmass' represent cells not belonging to any
% spill region, so we remove it.
tmass = tmass(2:end);

cum_reachable = zeros(size(ta.traps));

% This loop will accumulate the capacity of each trap to its spill
% region and all downstream regions
for trap_ix = 1:max(ta.traps)

   region = ta.trap_regions==trap_ix;

   % Attributing this trap's capacity to all cells in its spill region
   cum_reachable(region) = cum_reachable(region) + tmass(trap_ix);

   visited_regions = trap_ix;
```

```
   % Computing contribution to cells associated with downstream traps
   downstream = find(ta.trap_adj(:,trap_ix));
   while ~isempty(downstream)

      region = ta.trap_regions == downstream(1);
      cum_reachable(region) = cum_reachable(region) + tmass(trap_ix);

      visited_regions = [visited_regions;downstream(1)];

      % Downstream(1) has now been processed, so we remove it from the
      % downstream vector.
      downstream = [downstream(2:end); ...
         find(ta.trap_adj(:,downstream(1)))];
      downstream = setdiff(downstream, visited_regions);
   end
end

% We divide 'cum_reachable' by 1e9 to have the result in megatonnes
cum_reachable = cum_reachable / 1e9;

% Plot result (normal view)
plotCellData(Gt, cum_reachable, 'edgealpha', 0.1);
view(90, 90); axis tight; colormap cool;
c = colorbar; c.Label.String = 'Megatonnes';
set(c, 'fontsize', 20); set(gca, 'fontsize', 20);
set(gcf, 'position', [0 0 1400 600]);
```

The resulting plot is shown on Figure 3.17, which indicates that, from the perspective of exploiting structural trapping by gravity driven migration, the best injection locations are found in the south-eastern part of the aquifer (bottom left on the plot), where more than half a gigatonne of structural capacity might potentially be reached. Injection into the northern part, on the other hand, would only reach a fraction of this amount.

It should be noted that the simple analysis of this example says nothing regarding other trapping mechanisms. Also, as pointed out in the first part of this section, spill-point analysis is sensitive to errors in the geometry description, so it is important not to place too much confidence in any single analysis like this, especially when considering long migration distances and complex spill trees. Carrying out the analysis a large number of times with stochastic geometrical variations within the expected bounds of uncertainty would allow us to estimate the stability of individual spill path and the level of confidence that could be placed in the result.

We now return to the Johansen example, to demonstrate a reporting feature that requires the result from spill-point analysis before it can be invoked: the construction of a trapping inventory, i.e., a graph showing the trapping state of injected CO_2 in the aquifer over time. We start by performing the `trapAnalysis` on the top surface grid of the Johansen formation. We therefore return to the code example of Section 3.1 and execute the following code lines:

```
% Performing spill-point analysis on the Johansen grid
ta = trapAnalysis(Gt, false);
```

We then rerun the code used above to generate Figure 3.13, to make the correspond-

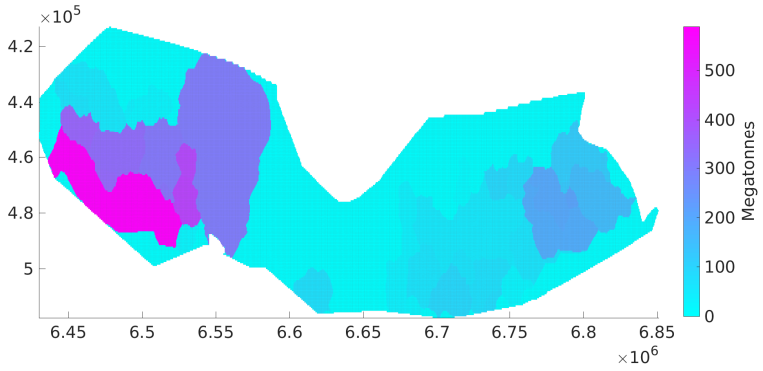


Figure 3.17: Plot indicating the amount of structural capacity reachable by gravity driven migration from each point in the Utsira aquifer.

ing plot for the Johansen formation. The result is shown in Figure 3.18. As the plot shows, the top surface of the Johansen simulation grid is very limited in detail, and only a few small traps are identified (plotted along with their spill regions and rivers).

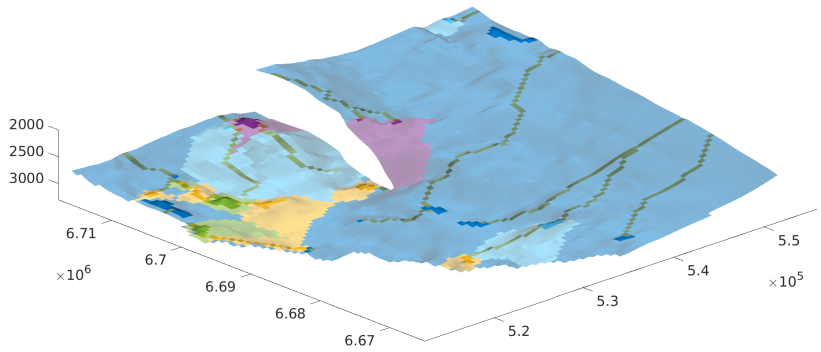


Figure 3.18: Top surface of the Johansen model: structural traps, spill regions and spill paths.

Having identified what little structural trapping there is, we can use the resulting information to generate the trapping inventory associated with the simulation we ran in Section 3.1. This is done by calling the function `plotTrappingDistribution`. This function requires that the results from the simulation is packaged along with trapping and other information into a report structure that we need to generate first. The following code does the trick:

```
% Generate report structure from grid, simulation result, rock and
% fluid objects, schedule and trapping information.
reports = makeReports(Gt, {initState states{:}}, model.rock, ...
                      model.fluid, schedule, [srw, src], ta, []);

% Generate inventory plot
h1 = figure; plot(1); ax = get(h1, 'currentaxes');
plotTrappingDistribution(ax, reports, 'legend_location', 'northwest');
set(gcf, 'position', [0 0 1100, 740])
set(gca, 'fontsize', 20);
```

The generated trapping inventory is shown in Figure 3.19. The horizontal axis represents time and the vertical axis amount of trapped CO_2. The colored areas thus indicate how the trapping state of injected CO_2 changes over time. During the first 100 years, injection is still ongoing and we see the total amount of CO_2 steadily increasing. Afterwards, changes in trapping state result from ongoing migration. The orange field represents freely flowing CO_2 in the plume, whereas the green fields represent residually trapped CO_2, whether in the imbibed regions, in structural traps, or still in the plume[9]. The yellow field represents CO_2 caught in structural traps and not already accounted for as residual trapping, whereas the red field represents CO_2 that has exited the simulation domain. As expected after having seen Figure 3.18, the amount of structurally trapped CO_2 in this simulation is so tiny as to be barely visible on the inventory plot.

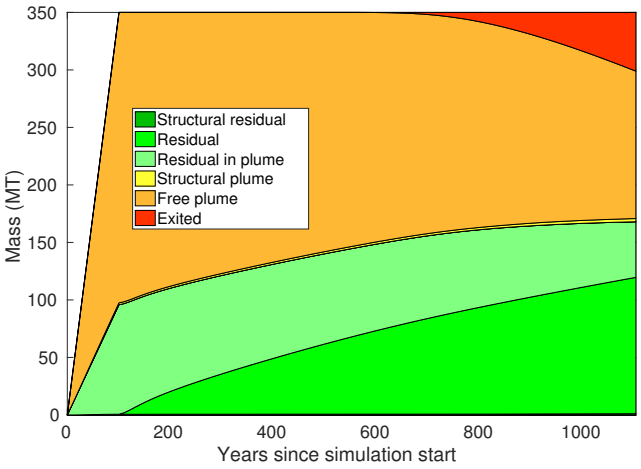


Figure 3.19: Trapping inventory constructed from the Johansen simulation example.

To round off the discussion on spill-point analysis, the inherent limitations of the method will be will briefly mentioned. These should be taken into account when applying the tool in a larger context. The method only provides information on structural trapping - other trapping mechanisms (residual trapping, dissolution) are either ignored or estimated by some simple calculations, as described in e.g., [67]. Moreover, the

[9] This CO_2 is not residually trapped yet, but will be so if the plume is later displaced by brine.

temporal aspect is completely missing - the result provided can be understood as the limit reached as time approaches infinity. In its simplest form, infinitesimal flow is assumed, i.e., the migration path is considered to be basically one-dimensional, which would only be true for a CO_2 plume with zero width. Last, but not least, the results can be highly sensitive to small, local changes in caprock shape, especially in relatively flat areas and near bifurcation points. This is important to be aware of, since caprock shape data may contain considerable uncertainty to begin with, and since additional inaccuracies will be introduced by the gridding, especially for coarse grids. Specifics on how geometrical data is interpreted by the algorithm may also lead to significant differences in the result, as further discussed in Paper II.

3.3 Simplified geomechanics

In Section 2.2.5, the equations describing a linear poroelastic system were presented. These equations couple the continuity equation for fluid flow with the equations for linear elastic behavior of a solid in static equilibrium. When the full mechanical impact on fluid flow in a geologic formation needs to be modeled, such a coupling is generally necessary. However, as also discussed in Section 2.2.5, the combined poromechanical system is significantly more expensive to solve than a flow model in isolation[10] due to the introduction of new unknowns, a significant extension of the computational domain, and a larger system of equations to solve. To solve the system, sequential splitting techniques are frequently employed. Under these approaches, dedicated solvers for flow and mechanics are used together, with information passed back and forth in a sequential or iterative manner designed to converge to the correct solution [30, 47, 59, 66, 97, 98]. In particular, the "fixed stress split" approach has proved to exhibit good convergence properties [59, 70]. A significant advantage of operator splitting techniques is that highly sophisticated existing solvers can be reused, which allows using state-of-the-art modeling capabilities and numerical methods both for flow and mechanics calculations. Moreover, the sequential splitting strategy may sometimes provide computational advantages over the alternative approach where the full set of equations including all the unknowns for mechanics and flow is solved as one monolithic system. Nevertheless, even the use of well-adapted sequential splitting schemes incur significant additional computational cost to reservoir simulation, and moreover require access to a dedicated mechanics solver during the course of running the simulation. As previously argued, many questions related to CO_2 storage are most practical to investigate using simplified, lightweight tools, and the inclusion of computationally intensive mechanics does not fit well in this setting. Some reduction in computational cost can be obtained using the poroelastic linear vertical deflection model proposed in [15], although this still requires the coupling with a mechanics solver. This section discusses some alternative approaches that allow us to include the impact of mechanics of fluid flow in an approximate sense without having to solve the mechanics equations during the simulation procedure.

[10] This is even more true if nonelastic mechanical behavior is included.

3.3.1 Pore volume compressibility and Geertsma's uniaxial poroelastic expansion coefficient

In Section 2.1.3, the concept of pore volume compressibility was introduced. Equation (2.3) presented a simple model for associating changes in local pressure with changes in local pore volume. Since deformations are generally small, a linearized version of Equation (2.3) is often sufficient:

$$\phi(p) = \phi_0 \left[1 + c_\phi (p - p_0)\right]. \tag{3.18}$$

The use of pore volume compressibility coefficients in reservoir modeling can be seen as a shortcut to take the impact of rock deformation on pore volume into account without actually having to include the equations for mechanics in the model. The coefficient $c_\phi = \frac{1}{\phi}\frac{d\phi}{dp}$ is supposed to capture not only bulk volume changes of the rock, but also the compressibility and packing of the individual grains at the microscopic scale. By revisiting the flow equation (2.37) of the poroelastic system presented in Chapter 2, one can see that the complete accumulation term may be written:

$$\frac{d\zeta}{dt} = \frac{\phi}{K_f}\frac{dp}{dt} + \frac{\phi}{K}(1-\alpha)\left(\frac{\alpha}{\phi} - 1\right)\frac{dp}{dt} + \alpha\frac{d(\nabla\cdot\mathbf{u})}{dt}. \tag{3.19}$$

Moreover, according to (2.34):

$$\frac{d\zeta}{dt} = \frac{1}{\rho}\frac{d(\phi\rho)}{dt} = \phi\frac{1}{\rho}\frac{d\rho}{dt} + \frac{d\phi}{dt}. \tag{3.20}$$

The first of the three terms in (3.19) is related to fluid compressibility and corresponds to the first term in (3.20). The second and third terms in (3.19) respectively represent grain compressibility (at the microscopic level) and changes in bulk volume (volumetric strain). These correspond to the second term in (3.20), i.e., $\frac{d\phi}{dt}$, and are thus what c_ϕ is intended to model. Since c_ϕ is defined as a pressure derivative, it is clear that one cannot expect it to accurately represent the contribution of the third term in (3.19), which does not depend solely on local pressure but is obtained from the poroelastic equations that involve the global pressure field. However, by temporarily assuming that $\nabla\cdot\mathbf{u}$ can be approximated as a function of local pressure only, the term can be rewritten as:

$$\frac{d(\nabla\cdot\mathbf{u})}{dt} \approx \frac{d(\nabla\cdot\mathbf{u})}{dp}\frac{dp}{dt}, \tag{3.21}$$

in which case one has:

$$\frac{d\phi}{dt} = \frac{d\phi}{dp}\frac{dp}{dt} = \phi c_\phi \frac{dp}{dt} = \phi\left[\frac{1}{K}(1-\alpha)\left(\frac{\alpha}{\phi} - 1\right) + \frac{\alpha}{\phi}\frac{d(\nabla\cdot\mathbf{u})}{dp}\right]\frac{dp}{dt}, \tag{3.22}$$

so that:

$$c_\phi = \frac{1}{K}(1-\alpha)\left(\frac{\alpha}{\phi} - 1\right) + \frac{\alpha}{\phi}\frac{d(\nabla\cdot\mathbf{u})}{dp}. \tag{3.23}$$

The obvious problem with (3.23) is that it only makes strict sense if $\nabla\cdot\mathbf{u}$ can be considered a function of local pressure, which is not generally true. However, under the

assumption of zero lateral strain and constant vertical stress, it *is* true, in which case one has the relation [110]:

$$\frac{d}{dp}(\nabla \cdot \mathbf{u}) = \frac{\alpha}{K_v} = c_m. \tag{3.24}$$

Here, K_v is the rock's *uniaxial bulk modulus*, which can be directly computed from the other elastic moduli. For instance, by using bulk K and shear G modulus, we have $K_v = K + 4/3G$. The constant c_m is called *Geertsma's uniaxial poroelastic expansion coefficient*.

The assumption of zero lateral strain and constant vertical stress is exact in the case of a uniform pressure increase in a laterally infinite, flat and homogeneous aquifer embedded in a larger rock matrix with a free-moving top surface at constant load, in which case by symmetry all mechanical displacement has to occur in the vertical direction only. While not strictly true in the general case, it is often an adequate approximation, especially for relatively smooth pressure fields. Choosing the pore volume compressibility coefficient c_ϕ as in (3.23) using the substitution in (3.24) can in that case offer a reasonable modeling choice.

3.3.2 Numerical generalization of Geertsma's coefficient

The use of Geertsma's poroelastic expansion coefficient to represent the relation between local pressure and volumetric strain can be generalized in order to take general boundary conditions, aquifer geometry and material heterogeneities into account. To do this, we start from the interpretation of c_m as the volumetric expansion resulting from a uniform, unit pressure increase in a flat, infinite aquifer. For a specific simulation model with arbitrary boundary conditions, a mechanics solver can be used to numerically compute the aquifer's volumetric strain response to a uniform pressure increase throughout the model domain. Whereas this approach will reproduce c_m for the finite, uniform case, the response for a general simulation case will typically be heterogeneous, reflecting specifics of the chosen boundary conditions and model irregularities. For instance, all else being equal, cells close to a clamped boundary will undergo less change than cells in the middle of the domain. A 2D example is shown in Figure 3.20, where the approach has been run twice on a simple aquifer model, using two different choices of boundary conditions and a unit pressure increase. When boundaries are clamped (zero displacement), the change in the volumetric strain field remains close to the theoretical value of c_m in the middle of the domain, but is considerably attenuated towards the edges. With roller-type boundary conditions (zero lateral displacement, constant vertical stress), c_m is reproduced throughout[11].

From here on, c_m^{num} is used to refer to the response numerically obtained using this method. In the same way as c_m, it can be used to establish the pore volume compressibility coefficient c_ϕ according to (3.23). The obtained value of c_ϕ will generally vary from one cell to the next. Each cell is thus modeled with its own poroelastic response to a given pressure change, reflecting the impact of boundary conditions and local variation in geometry and elastic properties. This approach can also be employed

[11] The choice of lateral boundary conditions for geomechanical problems will introduce some non-physical artifact to the model, since no well-defined lateral boundaries (usually) exist in reality. In real situations, the behavior at the model boundaries will typically be somewhere between 'clamped' and 'roller'. Generally, the best strategy is to keep lateral boundaries sufficiently far away so as to not significantly impact the local solution.

in cases where the use of Geertsma's poroelastic expansion coefficient is inapplicable due to non-constant poroelastic properties. However, it should be noted that although the use of c_m^{num} in the model to some degree incorporates information about boundary conditions and heterogeneities, volumetric strain is still treated as a function of local pressure, so the approach necessarily remains an approximation.

In Paper IX, this "local" approach is compared against a fully coupled model and a model based on precomputed response functions, described in the following subsection.

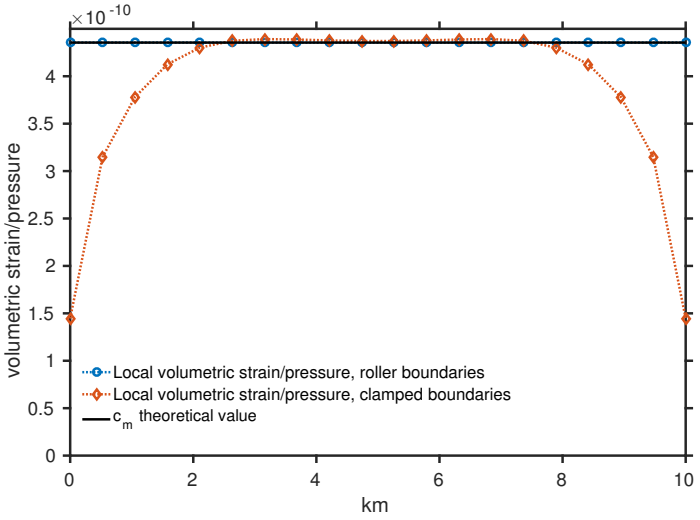


Figure 3.20: Volumetric response to a uniform, unit pressure increase throughout a two-dimensional flat and horizontal aquifer, assuming linear poroelasticity. The x-axis represents spatial position along the aquifer, whereas the y-axis represents the local change in volumetric strain.

3.3.3 Precomputed response functions

When solving the fully coupled poroelastic system, volumetric strain at any given point in the aquifer depends in theory on the whole fluid pressure field. On the other hand, as long as boundary conditions and body forces (i.e., gravity) remain constant, changes in volumetric strain will only depend on changes in the fluid pressure. Any given pressure field p can thus be associated with an unique volumetric strain field $\varepsilon(p)$ (for convenience, the symbol ε is used rather than $\nabla \cdot \mathbf{u}$). Moreover, since the mechanical force balance equations (2.33) are linear, the principle of superposition can be used. For a discretized simulation model, the pressure field will typically be described as a linear combination of some defined set of basis functions $\{\phi_i\}_{i=1\ldots m}$, i.e.,

$$p(x) = \sum_{i=1}^{m} p_i \phi_i(x). \tag{3.25}$$

Initial pressure p^0 and initial volumetric strain ε^0 are assumed to be in equilibrium. The initial volumetric strain is determined not only by pressure but also on the specified

boundary conditions **bc** and body forces **f**. Since everything is linear, one can formally write:

$$\begin{aligned}\varepsilon &= \varepsilon(p, \mathbf{bc}, \mathbf{f}) \\ &= \varepsilon(p^0, \mathbf{bc}, \mathbf{f}) + \varepsilon(p - p^0, 0, 0) \\ &= \varepsilon^0 + \tilde{\varepsilon}(\tilde{p}),\end{aligned}$$

where $\tilde{\varepsilon} = \varepsilon - \varepsilon^0$ and $\tilde{p} = p - p^0$. Also, a zero in the place of **bc** should be interpreted as zero displacement and zero stress on the related parts of the boundary. By further defining $\tilde{p}_i = p_i - p_i^0$, one can write:

$$\tilde{\varepsilon}(\tilde{p}) = \tilde{\varepsilon}\left(\sum_{i=1}^{m} \tilde{p}_i \phi_i\right) = \sum_{i=1}^{m} \tilde{p}_i \tilde{\varepsilon}(\phi_i). \tag{3.26}$$

In this notation, $\tilde{\varepsilon}(\phi_i)$ represents the global volumetric strain response in the aquifer resulting from an unit change in pressure associated with basis function ϕ_i. Knowledge of the full set of response functions $\{\tilde{\varepsilon}(\phi_i)\}_{i=1\ldots m}$ will therefore make it possible to quickly determine the volumetric strain field resulting from any pressure field that can be expressed using basis functions $\{\phi_i\}_{i=1\ldots m}$.

One can now write:

$$\frac{\partial \varepsilon}{\partial p_i} = \frac{\partial \tilde{\varepsilon}}{\partial \tilde{p}_i} = \tilde{\varepsilon}(\phi_i). \tag{3.27}$$

Thus by using $\tilde{\varepsilon}(\phi_i)$ instead of c_m to model pressure-dependent changes in volumetric strain, one can establish a set of *non-local*[12] pore volume compressibility coefficients c_ϕ (3.23) that represent the *exact* impact of mechanics on flow, thereby eliminating the need for solving the full system of poroelastic equations. This also eliminates the need for coupling with a mechanical solver when the simulation is run. The response functions $\{\tilde{\varepsilon}(\phi_i)\}_{i=1\ldots m}$ can be computed at the time the simulation grid is first generated, and stored along with the grid description. Although volumetric strain is defined on the whole mechanical domain Ω, it is sufficient to store the part that covers the flow domain $\Omega_f \subset \Omega$. Assuming that the response functions are made available as part of the input data, the only substantial change required for a standard flow simulator to model geomechanical impact using this approach is added support for non-local pore volume compressibility coefficients.

An important observation, crucial for this approach to be practical on large grids, is the *relatively* local nature of the response functions $\tilde{\varepsilon}(\phi_i)$. Although the support of $\tilde{\varepsilon}(\phi_i)$ is global, the response tends to decay quickly as a function of distance from the support of ϕ_i, a behavior that can to some degree be attributed to the free-moving top surface (considering that although the aquifer may be deep, the mechanical domain still extends up to the ground surface where a constant stress boundary condition applies). This is illustrated on Figure 3.21, where $\tilde{\varepsilon}(\phi)$ is plotted for a radially symmetric case, as a function of distance from the central pressure perturbation. The perturbation in this case consists of a unit pressure increase within a cylinder with radius 10 m and height 100 m (which also corresponds to the thickness of the aquifer). The actual shape of the response curve depends on several factors, including aquifer depth, thickness and

[12] "Non-local" here means that a pressure change in a given cell will also affect pore volume in other cells.

elastic moduli, but the example reflects what is qualitatively observed in typical cases. The region within the support of ϕ undergoes by far the largest expansion, followed by a compression zone and then an expansion region caused by caprock uplift (assuming caprock is stiffer than aquifer) that decays towards zero. Paper IX further discusses the shape of this curve and how it can be interpreted.

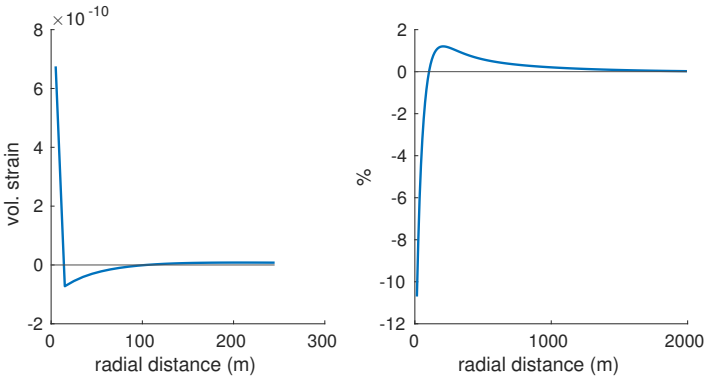


Figure 3.21: Example of a response function. The curve represents volumetric strain as a function of distance from a central, unit pressure perturbation. The left plot includes the value of the perturbed grid cell itself. On the right plot, the central value is suppressed and the curve re-scaled to show the percent-wise magnitude compared with the (removed) central value.

Since most of the impact of a response function tends to be relatively localized, the response can be truncated at a certain distance from the central perturbation when its absolute value permanently drops below some small threshold (e.g., 0.1% of its central value). A truncation must be followed by a re-scaling of the remaining part in order to exactly preserve the total volumetric change it represents. The advantage of truncation is that a sparse rather than a full matrix can be used to store the response functions, which is crucial for large simulation grids both in terms of storage and for the practicality of employing iterative linear solvers. Interestingly, if ϕ_i is defined as the indicator function on gridcell i, as would be the typical choice for a finite-volume formulation, it can be shown (c.f. the appendix of Paper IX) that the following relation always holds:

$$\int_{\Omega_A} \tilde{\varepsilon}(\phi_i)\mathrm{dx} = (c_m^{\mathrm{num}})_i, \tag{3.28}$$

where $(c_m^{\mathrm{num}})_i$ denotes the value of c_m^{num} on cell i. This means that the approach based on precomputed response functions with cell indicator functions as a pressure basis has the following two limit cases: (1) it reproduces the fully coupled model when the truncation threshold goes to zero; (2) it becomes equivalent with the local approach described in Section 3.3.2 when the truncation threshold is raised high enough to concentrate the whole response within a single gridcell.

The initial computation of response functions is expensive, since each $\tilde{\varepsilon}(\phi_i)$ (in principle) requires a separate solution of the linear elasticity equations. However, the complete task needs to be carried out only one time, when the grid has been initially generated. The procedure can also be made more efficient by computing several responses at the same time (provided they are sufficiently spatially distinct that they do

not practically interfere), by investing some initial computing time in constructing a good preconditioner for the linear system, and to parallelize the computing task across multiple cores. Parallelization is straightforward, since each response function can be computed completely independently of the others.

3.4 Optimization and integrated workflows for large-scale utilization

Simplified models can be combined with each other or with other mathematical methods in order to address larger storage-related questions. In this last section of the chapter, a very simple but complete code example is provided where VE simulation is combined with a non-linear optimization algorithm in order to determine an optimal injection rate for our Johansen example. This very simple example is followed by a discussion of more complex workflows.

The combination of simplified modeling and other computational tools is the topic of several of the articles included in this dissertation, in particular in Paper V, Paper VI and Paper VII. The most advanced exampled so far in using `MRST-co2lab` (and `MRST`) methods within larger workflows is however found in [8] and [75]. The first of these two papers investigates optimal injection strategies for several formations situated on the Norwegian Continental Shelf, taking pressure buildup, CO_2 leakage and economic considerations into account. The second paper uses VE modeling as a key component within a workflow for history matching of key parameters of a simulation model of the Sleipner storage site, based on input in the form of time-lapse seismic data.

For all the workflows presented in this chapter and in the papers mentioned above, non-linear optimization plays a key role. The discussion of our Johansen example below will therefore start as a very brief introduction to this computational topic, while sticking to the tutorial form of earlier sections.

3.4.1 Specifying a criterion for optimality

When searching for optimal injection rates, the first thing that needs to be specified is what is meant by 'optimal'. To formulate this in the language of mathematical non-linear optimization, one defines an *objective function*: $J : S \subseteq \mathbb{R}^n \to \mathbb{R}$, where n is the number of input variables and S the subset of $\mathbb{R}^N$ of the admissible values. For our Johansen example, the input variables represent the injection rates we seek to optimize. If the vector of input variables is denoted $\mathbf{u} \in S$, the corresponding (scalar) value of the objective function is $J(\mathbf{u})$. As the objective function measures the "desirableness" of the chosen values for $\mathbf{u}$, the goal is to find a set of values $\mathbf{u}^* \in S$ that maximize J, i.e.

$$J(\mathbf{u}^*) \geq J(\mathbf{u}) \quad \forall \mathbf{u} \in S. \tag{3.29}$$

Since an objective function can be of any shape, it is not in general possible to determine whether the global maximum $\mathbf{u}^*$ has been found. Instead, one might have to settle for a local maximum, such that:

$$J(\mathbf{u}^*) \geq J(\mathbf{u}) \quad \forall \mathbf{u} \in \mathcal{V}(\mathbf{u}^*) \subseteq S, \tag{3.30}$$

where $\mathcal{V}(\mathbf{u}^*)$ is some local, open neighborhood around $\mathbf{u}^*$. Assuming that J is differentiable at $\mathbf{u}^*$, a necessary condition for $\mathbf{u}^*$ to be a local optimum is that $\nabla J(\mathbf{u}^*) = \mathbf{0}$.

When this is true and if moreover J is twice differentiable at $\mathbf{u}^*$, a sufficient condition for local optimality is that the Hessian $\nabla^2 J(\mathbf{u}^*)$ is negative definite.

In the following, an injection scenario that involves W wells with time-varying rates is considered, where the simulated injection period is discretized into N timesteps of duration Δt. The total number n of rates to determine is $n = N \times W$, and the corresponding amount of CO_2 injected, Q_I, is:

$$Q_I = \sum_{i=1}^{N} \sum_{w=1}^{W} u_w^i \Delta t, \tag{3.31}$$

where u_w^i denotes the injection rate specified for well w at timestep i. To keep notation compact, the vector of injection rates at timestep i is here written $\mathbf{u}^i = [u_1^i, ..., u_W^i]^T$, and $\mathbf{u} = [\mathbf{u}^1, ..., \mathbf{u}^N]^T$.

If our goal were just to inject as much CO_2 as possible, we could imagine using Q_I as our objective function. However, Q_I does not take into account any physical limitations and has no maximum value, leading to "infinitely high" rates as the optimal solution - a physically meaningless result.

By taking another look at Figure 3.19, we note that our VE simulation of the Johansen example led to a certain migration of CO_2 out of the intended storage domain, from about 620 years (red color), which is obviously an undesirable outcome. Leakage across lateral boundaries can however be avoided by injecting less CO_2 in the first place. We therefore seek an injection rate that allows us to store as much CO_2 as possible while keeping migration out of the domain to a minimum.

If we use Q_L to denote the quantity of CO_2 migrating out of the domain, and l^i the total rate of leakage across domain boundaries at timestep i, we have:

$$Q_L = \sum_{i=1}^{N} l^i \Delta t. \tag{3.32}$$

We can now specify an objective function J that uses Q_L to counterbalance Q_I. In our example, we define optimal injection rates to be the set of rates that maximizes this function:

$$\begin{aligned} J(\mathbf{u}) &= Q_I(\mathbf{u}) - C Q_L(\mathbf{u}) \\ &= \sum_{i=1}^{N} \left(\sum_{w=1}^{W} u_w^i - C l^i \right) \Delta t \\ &= \sum_{i=1}^{N} J^i(\mathbf{u}^i). \end{aligned}$$

C is here a scalar weighting factor, and $J^i(\mathbf{u}^i)$ is a shorthand for the corresponding term in the summation above. It is clear that for sufficiently small injection rates there will be no leakage, $Q_L = 0$, and J will increase with increasing rates. However, once leakage occurs, its magnitude will also increase with increasing rates, so for a given $C > 1$ and sufficiently high values of $\mathbf{u}$, we expect J to *decrease* with rates. This suggests the

existence of a maximum value of J, at which an infinitesimal increase in rates will be exactly counterbalanced by C times the associated infinitesimal increase in leakage.

This choice of optimality does not account for the risk of dangerous overpressure associated with high injection rates. It is possible to include pressure limitation as an additional optimization criterion, but in our case there is no need to, since we already know that the initial rate in the Johansen example is higher than the optimal rates we seek (given the presence of significant leakage). Even if this were not the case, it would always be possible to inspect the pressure post-hoc to confirm whether the result is acceptable.

The computation of Q_I is simply a matter of adding up injection rates. The computation of Q_L, however, require us to determine the leakage rates l^i, which depend on the corresponding state variables (pressure, saturations, etc.) and thus cannot be known without running an actual simulation. Since an optimization procedure generally necessitates a (often high) number of objective function evaluations, the advantage of "rapid" simulations offered by the use of VE models is clear.

3.4.2 Specification of constraints and the general problem formulation

Unless the subset S in (3.29) equals all of $\mathbb{R}^n$, it can be specified by introducing specific restrictions on the input variables $\mathbf{u}$. These restrictions are referred to as *constraints*, and come in two different types, *equality constraints* and *inequality constraints*. An equality constraint specifies a functional relationship between input variables that must be fulfilled at all times, and is formally written $c(\mathbf{u}) = 0$, where $c : \mathbb{R}^n \to \mathbb{R}$. Similarly, an inequality constraint specifies an algebraic relation that has to remain non-positive: $d(\mathbf{u}) \leq 0$, with $d : \mathbb{R}^n \to \mathbb{R}$. All individual equality and inequality constraints can be summarized using vectors of functional relationships: $\mathbf{c}(\mathbf{u}) = \mathbf{0}$ with $\mathbf{c} : \mathbb{R}^n \to \mathbb{R}^p$, and $\mathbf{d}(\mathbf{u}) = 0$ with $\mathbf{d} : \mathbb{R}^n \to \mathbb{R}^q$, where p is the total number of equality constraints and q the total number of inequality constraints. Using the above notation, the general optimization problem can be stated in mathematical terms as follows:

$$\mathbf{u}^* = \arg\max_{\mathbf{u}} J(\mathbf{u})$$

subject to:

$$\mathbf{c}(\mathbf{u}) = \mathbf{0}$$
$$\mathbf{d}(\mathbf{u}) \leq \mathbf{0}.$$

While the above formulation describes a maximization problem, optimization problems are often formulated as a minimization problem instead. This is however just a convention, since one problem can be turned into the other by changing the sign of J.

3.4.3 Gradient-based nonlinear optimization

As discussed above, a necessary optimality condition for an unconstrained optimization problem is that $\nabla J(\mathbf{u}^*) = \mathbf{0}$, provided that J is differentiable at $\mathbf{u}^*$. If equality constraints are involved[13], a similar requirement can be stated by defining the Lagrangian

[13] The more complicated case of inequality constraints is here set aside.

function:

$$\mathcal{L}(\mathbf{u},\boldsymbol{\lambda}) = J(\mathbf{u}) + \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{u}). \tag{3.33}$$

Here, the components of $\boldsymbol{\lambda} \in \mathbb{R}^p$ are referred to as *Lagrange multipliers*. The necessary condition for optimality in the constrained case is that $\nabla\mathcal{L}(\mathbf{u},\boldsymbol{\lambda}) = [\nabla_{\mathbf{u}}\mathcal{L}, \nabla_{\boldsymbol{\lambda}}\mathcal{L}] = \mathbf{0}$. Both the unconstrained and constrained optimization problem can thus be attacked by searching for *stationary points* of J or $\mathcal{L}$, i.e., points where the gradient vanishes. Various iterative strategies exist for finding such points, where one start at some initial guess $\mathbf{u}_{(0)}$ and gradually update it using some algorithm, generating a sequence $\{\mathbf{u}_{(0)}, \mathbf{u}_{(1)}, ...\mathbf{u}_{(k)}, ...\}$ that is set up to converge towards some stationary point (provided it exists and can be found). A very simple example is the *steepest ascent (descent)*, which converges towards a local maximum (or minimum) of J by searching along the local gradient at $\mathbf{u}_{(k)}$ to determine the new point $\mathbf{u}_{(k+1)} = \mathbf{u}_k + t\nabla J(\mathbf{u}_{(k)})$, where t is a scalar determining the appropriate step-length. This is an example of a local ascent (descent) algorithm iteratively seeking to increase (decrease) the value of the objective function. A more advanced example of this type of methods is the *nonlinear conjugate gradient* method [78]. Other methods aim to locate a stationary point directly using a quasi-Newton approach, where the Newton-Raphson method (c.f. Section 2.2.4) is applied using an iteratively improved approximation of the Hessian matrix ($\nabla^2 J(\mathbf{u})$ or $\nabla^2\mathcal{L}(\mathbf{u},\boldsymbol{\lambda})$. One prominent example is the BFGS (Broyden, Fletcher, Goldfarb, Shanno) method [78].

Regardless of whether the algorithm used is of the local ascent or the quasi-Newton type, it is necessary to be able to evaluate not only $J(\mathbf{u})$ but also the gradient $\nabla J(\mathbf{u})$. This is in general a nontrivial task, since computing the value of J will involve the state variables which can only be obtained from u by running a full simulation. A closed-form expression is thus out of the question, and so is computing an approximation of $\nabla J(\mathbf{u})$ by finite differencing, which would require many (costly) evaluations of J, in particular for large n. However, in Section 2.2.4, it was described how the Jacobian of the left-hand side of the discretized system equations $\mathbf{G}(\mathbf{x}^i, \mathbf{x}^{i-1}, \mathbf{u}^i) = \mathbf{0}$ could be immediately obtained when setting up the equation system, using automatic differentiation (see also Appendix A). Here $\mathbf{x}^i$ represents all state variables (pressure, saturations, etc.) at timestep i. (Similar to the notation for control variables, the notation $\mathbf{x} = [\mathbf{x}^1, ..., \mathbf{x}^{N_T}]^T$ is here used to represent the full vector of all state variables for all timesteps). It turns out that the knowledge of the Jacobian of $\mathbf{G}$ is just what is needed to compute the gradient of $J(\mathbf{u})$ in a very efficient manner, using the *adjoint-state method* [87]. To illustrate its principle, the following discussion outlines on how the method can be applied to obtain the gradient in the case of a simple, unconstrained optimization problem (such as the case of our Johansen example). A more general formulation that includes constraints and specified output variables, adapted to the reservoir simulation context, can be found in [57].

The starting point is to define the following Lagrangian function:

$$\begin{aligned}\mathcal{L}(\mathbf{u},\mathbf{x},\boldsymbol{\lambda}) &= J(\mathbf{u},\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{G}(\mathbf{x},\mathbf{u}) \\ &= \sum_{i=1}^{N_T} J^i(\mathbf{u}^i,\mathbf{x}^i) + (\boldsymbol{\lambda}^i)^T \mathbf{G}(\mathbf{x}^i,\mathbf{x}^{i-1},\mathbf{u}^i).\end{aligned}$$

In the above formulation, $\mathbf{u}$, $\mathbf{x}$ and $\boldsymbol{\lambda}$ are all considered independent variables, and $J(\mathbf{u},\mathbf{x})$ is written to emphasize the dependence of J on $\mathbf{x}$. The shorthand notation

$\mathbf{G}(\mathbf{x},\mathbf{u})$ is used to refer to the combined linear system representing *all* state variables and control variables for *all* timesteps. The vector of Lagrange multipliers $\boldsymbol{\lambda}^i$ associated with timestep i has the same dimension as $\mathbf{x}^i$ (which is also the number of linearized equations in $\mathbf{G}(\mathbf{x}^i,\mathbf{x}^{i-1},\mathbf{u}^i)$).

A stationary point of the Lagrangian function defined above represents a minimum (or maximum) of $J(\mathbf{u},\mathbf{x})$ under the constraint that $G(\mathbf{x},\mathbf{u})=\mathbf{0}$, i.e., that the state variables are tied to the control variables through the system equations. In other words, a stationary point of $\mathcal{L}(\mathbf{u},\mathbf{x},\boldsymbol{\lambda})$ also represents a minimum (or maximum) of $J(\mathbf{u})$ for the original, unconstrained optimization problem, where $\mathbf{x}$ depends on $\mathbf{u}$. The goal here is however not seek an optimal solution of $\mathcal{L}$ directly, but rather to use it to derive the gradient of the unconstrained objective function $J(\mathbf{u})$.

By using $\nabla_{\mathbf{x}^i}$ to denote the set of partial derivatives with respect to the variable set $\mathbf{x}^i$ (similar notation for $\mathbf{u}^i,\boldsymbol{\lambda}^i$), and considering the partial derivatives of $\mathcal{L}(\mathbf{u},\mathbf{x},\boldsymbol{\lambda})$ with respect to each set of variables, the following expressions are immediately obtained (using the notation $\mathbf{G}^i=\mathbf{G}(\mathbf{x}^i,\mathbf{x}^{i-1},\mathbf{u}^i)$):

$$\nabla_{u^i}\mathcal{L}=\nabla_{u^i}J^i+(\boldsymbol{\lambda}^i)^T\nabla_{u^i}\mathbf{G}^i \tag{3.34}$$

$$\nabla_{x^i}\mathcal{L}=\nabla_{x^i}J^i+(\boldsymbol{\lambda}^i)^T\nabla_{x^i}\mathbf{G}^i+(\boldsymbol{\lambda}^{i+1})^T\nabla_{x^i}\mathbf{G}^{i+1} \qquad \text{for } i\in\{1,...,N-1\} \tag{3.35}$$

$$\nabla_{x^N}\mathcal{L}=\nabla_{x^N}J^N+(\boldsymbol{\lambda}^N)^T\nabla_{x^N}\mathbf{G}^N \tag{3.36}$$

$$\nabla_{\lambda^i}\mathcal{L}=\mathbf{G}^i. \tag{3.37}$$

All the partial derivatives of $\mathbf{G}$ above are computable thanks to automatic differentiation. The partial derivatives of J^i with respect to $\mathbf{u}^i$ and $\mathbf{x}^i$ are usually also easy to compute, since $\mathbf{u}^i$ and $\mathbf{x}^i$ are here considered as independent variables. Various equations are obtained by requiring each set of partial derivatives of $\mathcal{L}$ to be zero. By equating (3.37) with zero, the original system equations (2.24) are obtained. By equating (3.35) and (3.36) with zero, one obtains the *adjoint-state equations*, which allow computing the *adjoint-state variables* $\boldsymbol{\lambda}$. It is relatively straightforward (c.f. [87]) to show that if $\mathbf{u}$ and $\mathbf{x}$ satisfy the state system equations ($\nabla_{\lambda^i}\mathcal{L}=\mathbf{0}$, $\forall i\in\{1...N\}$), and if $\boldsymbol{\lambda}$ satisfies the corresponding adjoint-state equations ($\nabla_{x^i}\mathcal{L}=\mathbf{0}$, $\forall i\in\{1...N\}$), then (3.34) equals the gradient we are looking for, i.e.

$$\begin{aligned}&\text{If: } (\mathbf{u},\mathbf{x},\boldsymbol{\lambda}) \text{ satisfy } \begin{cases}\nabla_{\lambda^i}\mathcal{L}=\mathbf{0} & \text{(system equations)}\\ \nabla_{x^i}\mathcal{L}=\mathbf{0} & \text{(adjoint-state equations)}\end{cases}\\ &\text{Then:}\qquad \nabla J(\mathbf{u})=\nabla_{u^i}\mathcal{L}(\mathbf{u},\mathbf{x},\boldsymbol{\lambda}) \quad \text{(obtained from (3.34)).}\end{aligned}$$

Step-wise, the procedure of computing the gradient $\nabla J(\mathbf{u})$ is thus as follows:

1. Compute $\mathbf{x}$ that satisfies the system equations by doing a forward simulation (this also allows us to compute the value of $J(\mathbf{u})$).
2. Compute $\boldsymbol{\lambda}$ by solving the adjoint-state equations. Start by computing $\boldsymbol{\lambda}^N$ using (3.36), and then successively compute $\boldsymbol{\lambda}^{N-1}$, $\boldsymbol{\lambda}^{N-2}$, ..., $\boldsymbol{\lambda}^1$ using (3.35).
3. Compute $\nabla J(\mathbf{u})$ using (3.34).

The `MRST` function `computeGradientAdjointAD` can be used to compute the gradient of an objective function associated with a specific simulation, using the adjoint-state equations.

3.4.4 Application on the Johansen example

We are now ready to demonstrate the approach on the Johansen example. For that, we use the objective function described in Section 3.4.1 which optimizes the amount of injected CO_2 while penalizing migration out of the domain, using the weighting factor $C = 10$.

In a first example, we seek to obtain the optimal, fixed injection rate for our single injection well. Since the rate is not allowed to vary, we have only a single unknown rate to optimize, i.e., $n = 1$. We revisit the example of Section 3.1, and assume it has already been run so that all data objects defined therein are present in memory. The injection rate used in that example, 3.5 Mt/year (amounting to 0.162 m^3/s for the specified reference density) is likely not optimal since the corresponding simulation shows it results in significant leakage during the migration period. We will use a optimization routine provided in `MRST-co2lab`, `optimizeRates`, to turn the rate down to the optimal level. The full code example, including plotting code, is:

```
% Fixed-rate optimization
min_rate = eps;  % minimum allowed rate should be (practically) zero
max_rate = schedule.control(1).W.val * 2; % maximum allowed rate

schedule.control(2).W.val = min_rate; % no injection during migration

% Compute optimized rates, using a leak penalty factor of 10
[optim, init, history] = ...
optimizeRates(initState, model, schedule, min_rate, max_rate, ...
              'last_control_is_migration', true,               ...
              'leak_penalty', 10);

% Plotting CO2 staturation for timestep 200 (1100 years after start)
[h, h_max] = upscaledSat2height(optim.states{end}.s(:,2), ...
                                optim.states{end}.sGmax,  ...
                                Gt,                       ...
                                'pcWG', fluid.pcWG,       ...
                                'rhoW', fluid.rhoW,       ...
                                'rhoG', fluid.rhoG,       ...
                                'p', optim.states{end}.pressure);
plotCellData(Gt.parent,                                   ...
             height2Sat(struct('h', h, 'h_max', h_max), ...
             Gt,                                          ...
             fluid));
colorbar; view(-63, 68);
set(gcf, 'position', [531, 337, 923, 356]); axis tight;


% Plot trapping inventory
ta = trapAnalysis(Gt, false);
reports = makeReports(Gt, {initState, optim.states{:}}, ...
                      model.rock, model.fluid,          ...
                      optim.schedule, [srw, src], ta, []);

h1 = figure; plot(1); ax = get(h1, 'currentaxes');
plotTrappingDistribution(ax, reports, 'legend_location', 'northwest');
set(gcf, 'position', [0 0 1100, 740]);
set(gca, 'fontsize', 20);
```

The function `optimizeRates` takes as arguments all information necessary to run the simulation (initial state, simulation model and schedule) as well as lower and upper bounds on the permitted rates. We specify the lower rate limit to be zero (a tiny positive value to avoid numerical problems), and the maximum rate to be twice the initial rate (we do not expect the rate to be adjusted upwards, but we set it a bit higher than initial just in case). When calling `optimizeRates` above, we also specify the weighting factor C (`leak_penalty`) and tell the routine not to adjust the (zero) rate associated with the migration period (`last_control_is_migration` set to `true`). It is also possible to specify a different objective function using the option `obj_fun`, but we stick with the default for this example.

The call to `optimizeRates` takes some time to complete, during which a number of simulations are run. When finished, a new injection schedule based on the optimized rate(s) is returned as a separate field of the `optim` structure, along with the associated simulation result and other information.

Finally, we use plotting code similar to that already used in Section 3.4.1 in order to visualize the optimized result. We choose to plot the saturation distribution at the end of the simulated period (Figure 3.22, upper plot), as well as the trapping inventory (Figure 3.23, upper plot). By comparing Figure 3.23 with Figure 3.19, we see that the total amount of injected CO_2 has decreased from 350 Mt using original rates to 103 Mt using the optimized rate (the volumetric rate set in the optimized schedule is 0.0477 m^3/s). We also see a significant reduction in CO_2 that has exited the domain, but also note that a large amount of mobile CO_2 remains in the formation, much of which is likely to migrate out of the domain in the future.

We now re-run the optimization procedure, this time allowing the injection rate to change every 25 years. This leads to an optimization problem with four unknown rates to optimize, i.e., $n = 4$. The new degrees of freedom are including by adding new controls to the initial injection schedule, and associating them with the corresponding timesteps, as shown in the six first code-lines of the following listing:

```
% Variable-rate optimization

% Add degrees of freedom to schedule: ability to change rate every 25
% years during injection
schedule.control = [repmat(schedule.control(1), 4, 1); ...
                    schedule.control(2)];
schedule.step.control(1:25)    = 1; % first 25 years, use wellrate 1
schedule.step.control(26:50)   = 2; % second 25 years, use wellrate 2
schedule.step.control(51:75)   = 3; % third 25 years, use wellrate 3
schedule.step.control(76:100)  = 4; % fourth 25 years, use wellrate 4
schedule.step.control(101:200) = 5; % migration period, shut off well

% Compute optimized rates, using a leak penalty factor of 10
[optim, init, history] = ...
   optimizeRates(initState, model, schedule, min_rate, max_rate, ...
                 'last_control_is_migration', true,              ...
                 'leak_penalty', 10);

% <-- plotting code is equal to the previous code example -->
```

When the new initial schedule has been set, the optimization procedure is run again. We plot the outcome using the exact same plotting code as in the previous case (Fig-

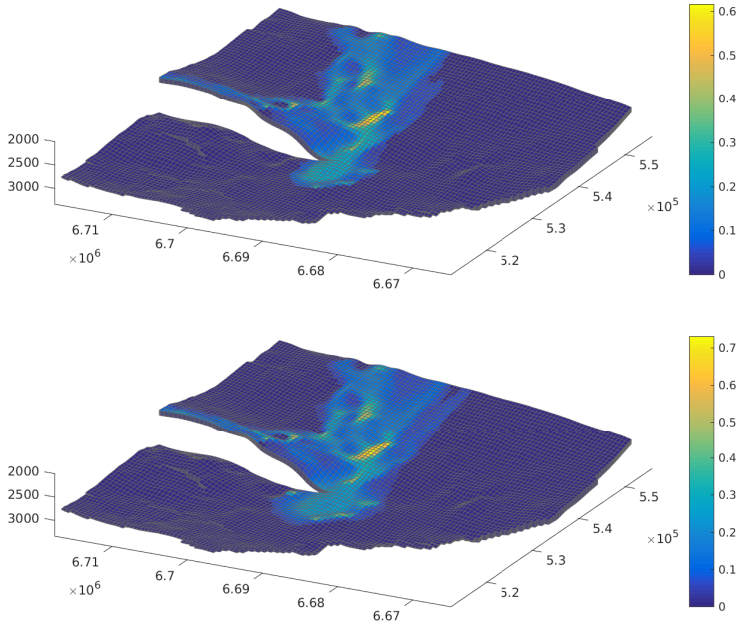


Figure 3.22: CO_2 saturation at the end of simulation in the Johansen scenario, using optimized rates. *Top:* optimized fixed rate; *Bottom:* optimized variable rate.

ure 3.22, bottom; and Figure 3.23, bottom). The inventory plot shows us that 146 Mt has now been injected, a significant increase over the optimal fixed-rate scenario. However, we also note that practically all CO_2 was injected during the last 25 years of the injection period. On Figure 3.22, we note that the CO_2 concentration around the injection site is spread out wider, suggesting the use of a higher injection rate. By inspecting the rates of the optimized schedule, we confirm that the three first rates, associated with years 1–75, have been set to zero, whereas the last rate, associated with years 76–100 has been set to 0.270 m^3/s. By injecting all CO_2 towards the end of the designated injection period, the total migration time becomes correspondingly shorter, which leaves less time for leakage within the total simulated period and allows more CO_2 to be injected. While this only shifts the leakage into the future and may therefore not represent a particularly useful improvement over the fixed-rate case, it serves to illustrate that the long-term storage capacity of a geological formation generally depends on the time horizon used.

3.4.5 More advanced workflows

The demonstration on the Johansen formation above is a very basic example on how simplified models can be used in combination with a nonlinear optimization algorithm to constitute a simple workflow. The example is simple enough that the full code example could easily be included in one section. However, the utility of simplified models can extend to considerably more complex workflows that integrate multiple modeling

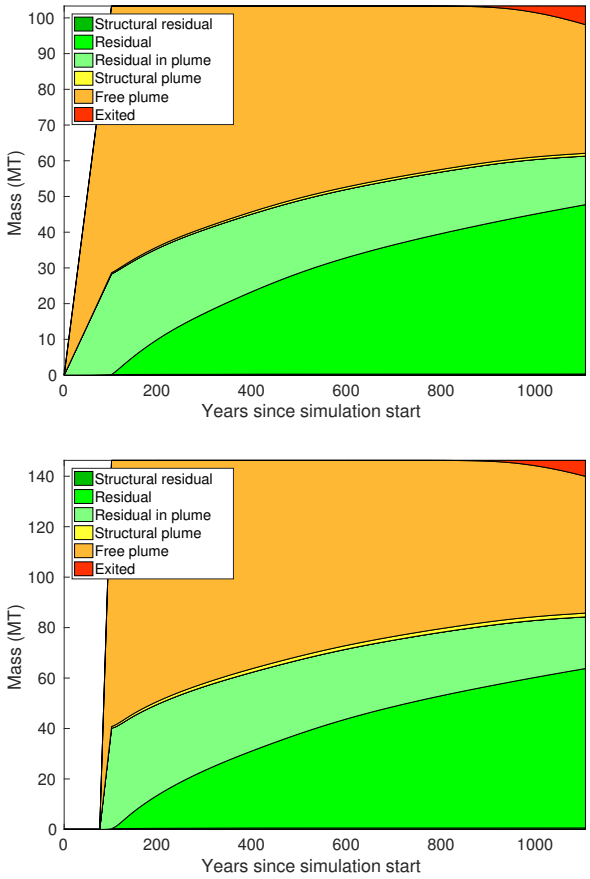


Figure 3.23: Trapping inventories constructed from the optimized Johansen examples. *Top:* optimized fixed rate; *Bottom:* optimized variable rate.

approaches and mathematical methods in non-linear ways. To illustrate this with another example, the rest of this section sketches a workflow used to investigate dynamic (i.e., simulation-based) storage capacity and optimize a storage scenario for the Statfjord aquifer, a formation off the coast of Norway at depths between 1700 and 6300 meters (c.f. Figure 3.24).

The example is only meant to illustrate the workflow, so specifics on the parameter values and other details are left out, and the simulation outcomes are left with little discussion. For the same reason, the corresponding MATLAB code is too long and detailed to be presented here. The focus here is on the process itself, not the results obtained. The user can still experiment with the Statfjord formation data using the interactive tools `exploreCapacity` and `exploreSimulation`, presented at the end of Appendix C).

The outline of the workflow to be discussed is shown in Figure 3.25. Components providing input or computation to the workflow (users, algorithms, tools) are indicated

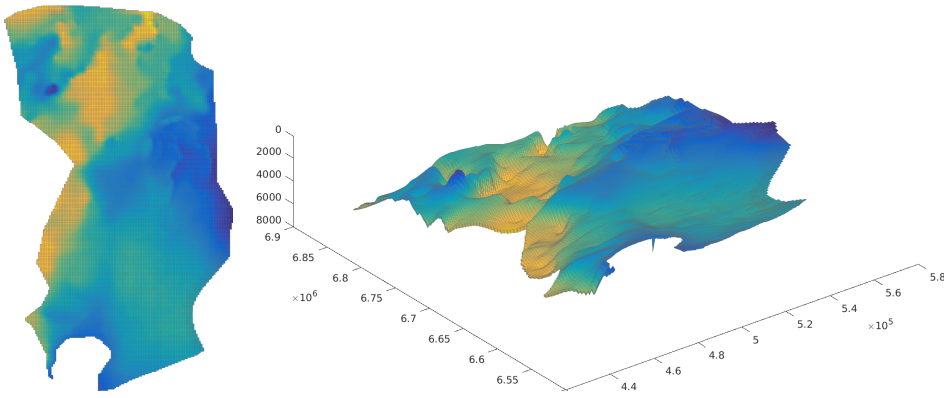


Figure 3.24: Illustration of the Statfjord formation. *Left*: Top view. *Right*: Inclined view.

with green, rounded boxes, whereas information generated and passed between components is shown as yellow rectangles. As the diagram is rather complex, it is presented below in a piece-wise manner, starting from the upper right.

Estimating upper limit of theoretical storage potential

The part of the workflow related to the estimation of theoretical storage potential[14] is shown in Figure 3.26. It consists of a fluid object (first discussed in Section 2.1.3), the spill-point analysis algorithm discussed in Section 3.2, and a simple script to estimate and add up trapping capacities by different mechanisms for a given formation (also used in the interactive example `exploreCapacity` presented in Appendix C). The spill point analysis provide information on structural traps, whereas the fluid objects provide CO_2 densities for the pressures and temperatures in the formation under study, which are necessary to provide storage estimates in mass terms. Combining this information with the simulation grid and corresponding rock model, the capacity computation script is able to produce an estimate of the formation's *retaining capacity*, which is a measure of how much CO_2 can be held indefinitely by various trapping mechanisms, assuming that CO_2 is free to leak across boundaries. It can be understood as the theoretical upper limit on the storage potential of a formation with completely open boundaries. Details on the methodology is presented in Paper VII.

The outcome of this procedure applied at the Statfjord formation is presented in Figure 3.27. The algorithm estimates that the formation is capable of retaining 80 gigatonnes of CO_2, broken down by trapping mechanism as shown in the table on the left of the figure. The retaining capacity is much larger than what one might realistically

[14] Regarding capacity estimation, it should be mentioned that there is not a single, universally used definition, and a number of different methodologies exist. *Static* estimates are not based on actual reservoir simulation, but derived directly from the physical characteristics of the formation. Such estimates are usually obtained by computing the total available pore space of a formation and multiplying by an *efficiency factor*, which depends on certain properties of the formation, such as the assumed nature of boundary conditions. *Dynamic* storage estimates, on the other hand, are based on numerical simulation of specific formations and usage scenarios, and are thus linked to particular, chosen strategies for storing CO_2 in the formation.

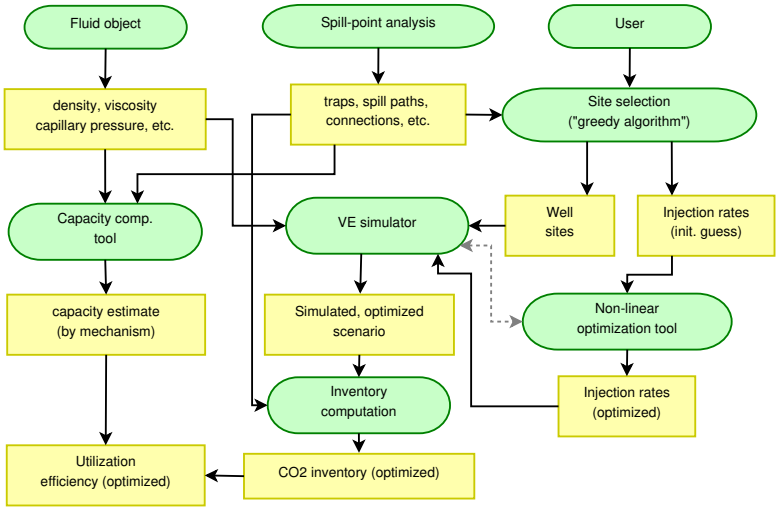


Figure 3.25: Overview of the Statfjord workflow discussed in this section.

expect to be able to store in the formation, since it assumes an initial, complete flooding of the whole formation with CO_2 (c.f. Paper VII). However, the estimate provides a theoretical upper limit, compares the potential magnitudes of different trapping mechanisms, and identifies the areas of the formation with highest ability to retain injected CO_2 (rightmost plot of the figure).

Placement of injection wells

The next step of the Statfjord workflow example consists of picking the locations of the injection wells. As the structural trapping capacity is the easiest to practically exploit, the placement strategy will use a heuristic approach where the goal is to reach as much structural trapping capacity as possible, under the assumption of gravity-driven migration. The associated part of the workflow diagram is shown in Figure 3.28. The key component here is a "greedy algorithm" used for suggesting well sites and corresponding injection rates. Based on the information obtained from the spill-point analysis, this algorithm generates a map of the aquifer where each grid cell in the top surface is attributed a value corresponding to the amount of structural trapping (in mass terms) reachable by gravity-driven migration from its location. This includes the size of the structural trap (if any) in its current spill region, as well as the combined sizes of all traps further upslope (c.f. Figure 3.10). Once this map (visualized in the left plot of Figure 3.29) has been constructed, the injection well location is chosen to be among the cells in the region with largest potential for reachable structural trapping (indicated in dark red on the figure). The corresponding injection rate is obtained by dividing the total reachable structural trapping by the duration of the injection period. In order to chose the position of the next well, all structural traps already covered by the first well are removed from the spill tree, and a new map is constructed based on the remaining traps. This process is repeated for each new injection site, until the requested number of sites has been covered. Variants of this selection procedure are discussed and used

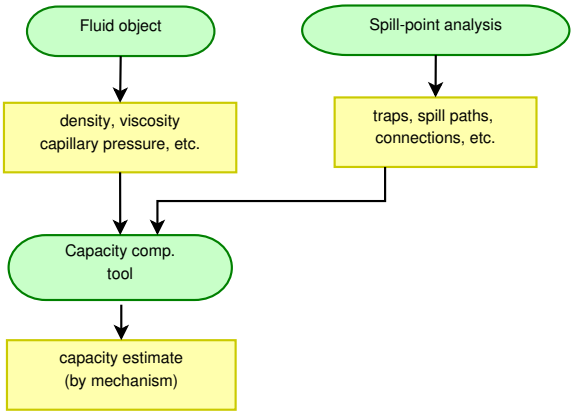


Figure 3.26: Part of the Statfjord workflow related to capacity estimation.

Theoretical retaining capacity:

Type	Value [Gt]	% of total
Structural	5.67	7.0
Residual	56.48	70.3
Dissolution	18.25	22.7
Total	80.40	100

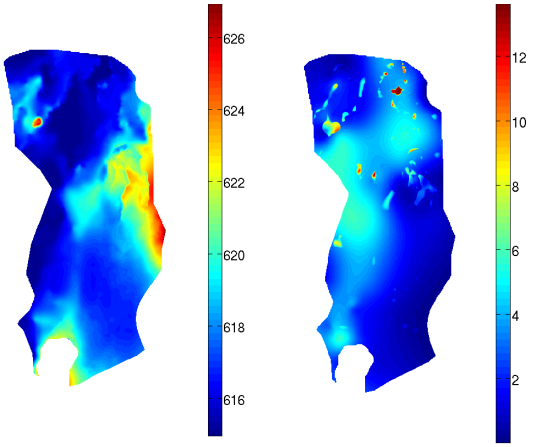


Figure 3.27: Estimated retaining capacity of the Statfjord aquifer. *Left*: Table showing estimated retaining capacity broken down by trapping mechanism. *Middle*: CO_2 density at caprock level (kg/m^3). *Right*: Trapping capacity as function of lateral coordinate (tonnes/m^2).

in Paper II and Paper VII.

Applying this algorithm for six wells to the Statfjord example produces the injection locations shown in the upper middle plot of Figure 3.29. As the basic procedure does not take boundaries into account, the position of well 4 has been chosen so close to the boundary that a large amount of CO_2 is likely to spill right out of the intended domain during injection. For this illustrative example, the problem is identified by the user, who decides to move the well to a different promising location instead[15], as shown on

[15] A more refined version of the site selection algorithm could certainly take the risk of leakage across domain boundaries into account, for instance by imposing requirements on minimal "safe" distances. The simple example presented here mainly serves to illustrate the general point that human input can be considered to be part of a workflow.

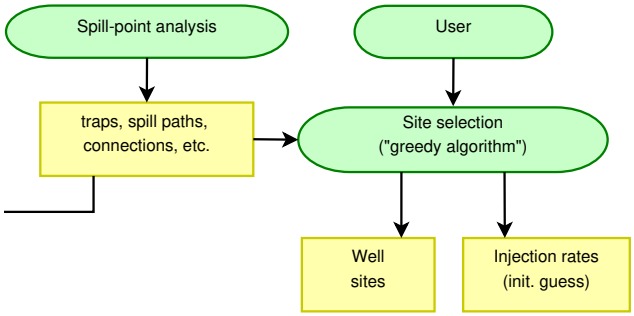
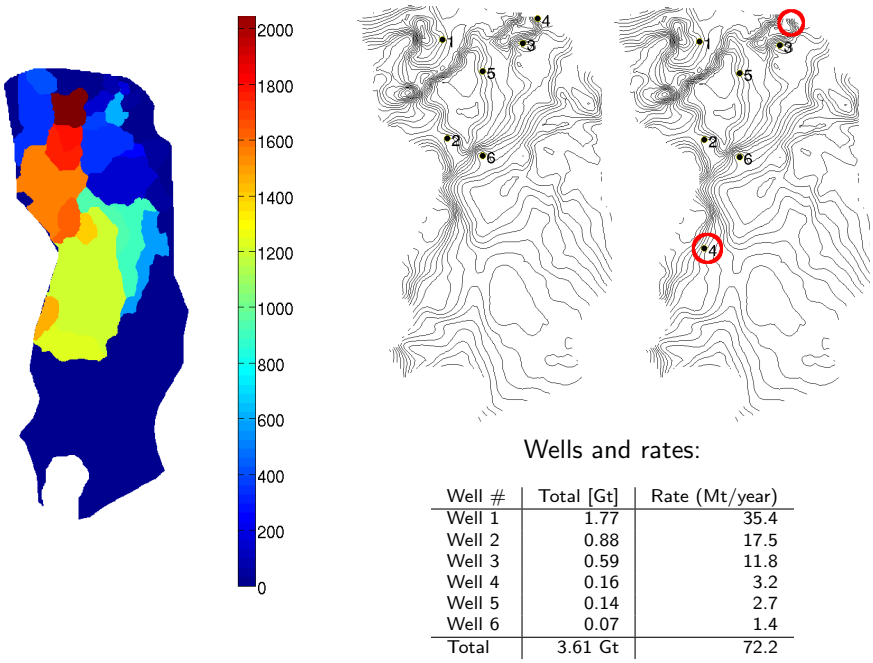


Figure 3.28: Part of the Statfjord workflow related to selection of well sites.



Well #	Total [Gt]	Rate (Mt/year)
Well 1	1.77	35.4
Well 2	0.88	17.5
Well 3	0.59	11.8
Well 4	0.16	3.2
Well 5	0.14	2.7
Well 6	0.07	1.4
Total	3.61 Gt	72.2

Figure 3.29: Information produced during the selection of well sites. *Left*: Reachable structural capacity (megatonnes). *Top middle*: Well positions suggested by greedy algorithm. *Top right*: New position of well 4, based on human input. *Bottom*: Well rates suggested by greedy algorithm.

the upper right plot of Figure 3.29.

Preliminary VE-simulation

After injection sites and tentative injection rates have been chosen, the next step of the Statfjord workflow is to run an actual simulation. On Figure 3.30 the components for

VE simulation and inventory computation have been added to the part of the workflow diagram discussed so far. The VE simulator components takes its inputs from the site selection algorithm and the fluid object, and produces results that are subsequently used to generate inventory plots similar to those in Figure 3.19 or 3.23. The outcome of the simulation serves to assess how much of the targeted structural traps are filled with CO_2, how much CO_2 ends up being trapped by other mechanisms, and the degree to which CO_2 ends up in regions not considered by the well placement algorithm. Since the well placement algorithm only takes structural trapping into account, it is to be expected that many of the initially suggested injection rates will be too conservative. Moreover, the flow of CO_2 will differ somewhat from the path predicted by spill-point analysis (which only takes gravity-driven flow into account), and this will also have an impact on what optimal injection rates should be.

A scenario with 50 years of injection and 2900 years of migration is considered. The simulation result is shown in Figure 3.31. After the end of the migration period, many of the traps in the northern region have been reached by migrating CO_2. However, several traps have not been reached, and significant migration out of the domain has occurred. The next step will thus be to seek an improved outcome by optimizing the injection rates.

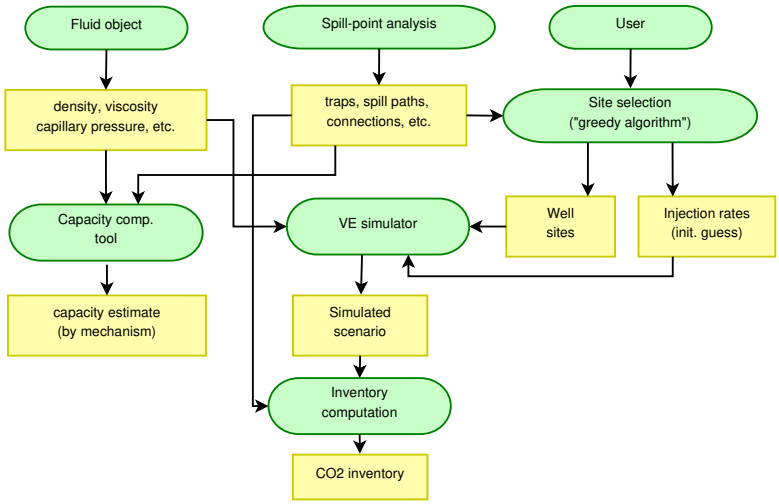


Figure 3.30: VE simulation included in the Statfjord workflow discussed so far.

Rate optimization

In Figure 3.32, the optimization component has been added to the workflow diagram. It takes input in form of initially suggested injection rates, and passes information both ways between the VE simulation component and itself, as previously laid out in Section 3.4.3. The outcome is a set of optimized injection rates, which are again passed to the VE simulator to compute the outcome of this optimized scenario. Again, the trapping inventory can be computed, and the result compared with the retaining capacity estimated in the first step of the workflow. The simulation result using optimized rates

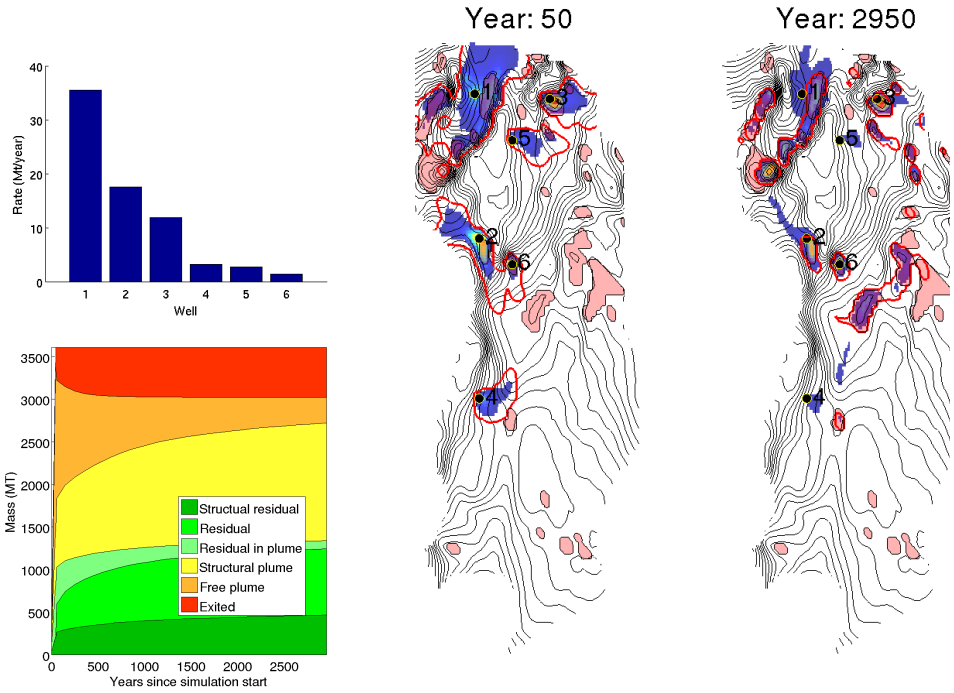


Figure 3.31: Results from VE simulation on Statfjord with injection rates obtained from well selection algorithm. *Top left*: Injection rates. *Bottom left*: Trapping inventory. *Middle*: CO_2 distribution after end of injection. Structural traps are indicated in pink. Outline of plume is traced in red. Regions with significant amounts of CO_2 (including residual CO_2) are colored in blue. *Right*: CO_2 distribution after 2900 years of migration.

is presented in Figure 3.33. A significant improvement can be seen[16], compared with the outcome from using the initial rates (Figure 3.31).

Further possibilities

The description of the Statfjord workflow example ends here, but further steps could be added. For instance, very high injection rates are considered, which would cause substantial levels of regional overpressure, but the above workflow does not include any assessment of pressure. The workflow could be extended by including pressure restrictions in the optimization procedure, as has been done in [8]. Injection sites or other locations where critical levels of overpressure are in risk of being reached can be made subject to more detailed studies. Such studies might be based on full 3D simulations at the local level, coupled with a geomechanical model. For local studies around an injection well, the CO_2 rate could be further adjusted in order to minimize the risk of damage to the confining layers. If gradient-based optimization is used for this

[16] It should be noted that both the initial and the optimized rates are here very high, and could cause overpressure issues. This illustrative workflow example does not include pressure considerations. The topic is touched upon under "**Further possibilities**" below.

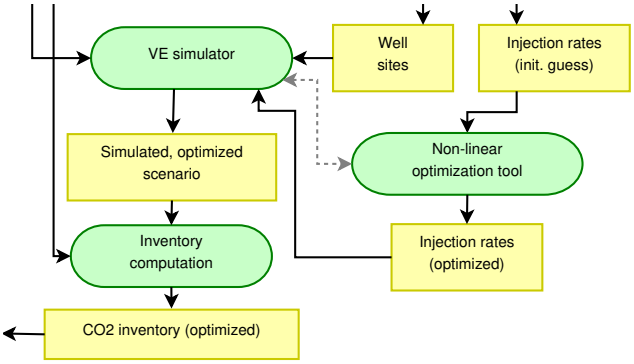


Figure 3.32: Optimization part of the Statfjord workflow.

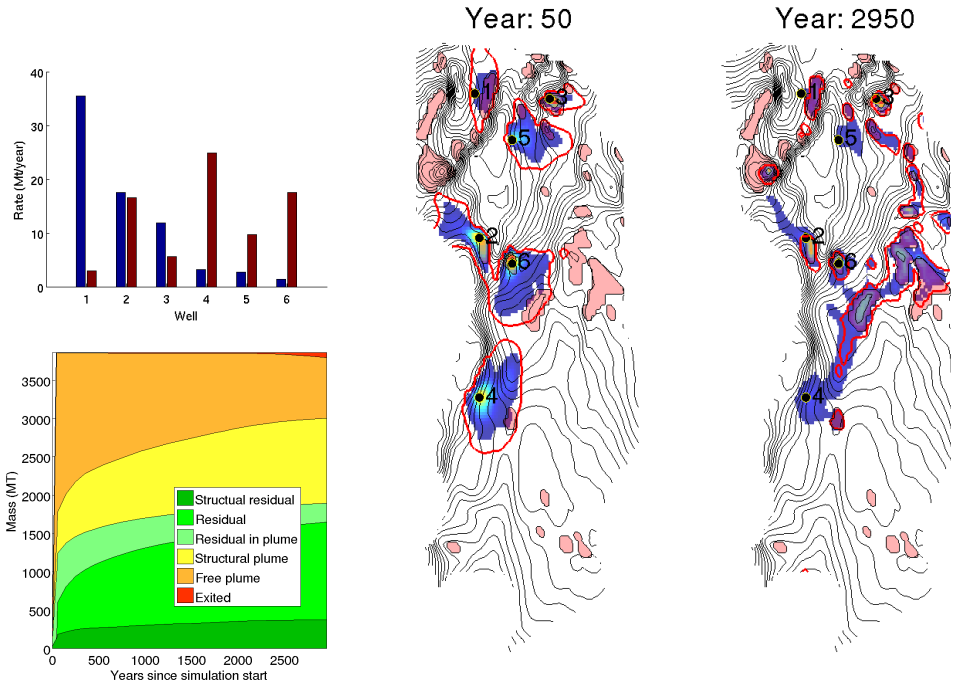


Figure 3.33: Results from VE simulation on Statfjord using optimized injection rates. *Top left*: Initial (blue) and optimized (red) injection rates. *Bottom left*: Trapping inventory. *Middle*: CO_2 distribution after end of injection. Structural traps are indicated in pink. Outline of plume is traces in red. Regions with significant amounts of CO_2 (including residual CO_2) are colored in blue. *Right*: CO_2 distribution after 2900 years of migration.

purpose, the simplified geomechanical model presented in Section 3.3 could serve to keep computational requirements manageable. To mitigate regional pressure buildup, brine-producing wells could also be introduced, and the optimal placement of such

wells would need its own strategy.

Other approaches could also be chosen for placing the injection wells. In Paper VI, a workflow applied on the Sandnes formation in the North Sea is described, where wells are placed based on a regular array. This strategy, which also involves a modified objective function, was chosen as a response to the very large structural trap volumes combined with relatively poor injectivity of the aquifer. This serves to illustrate the philosophy that there is not a one-size-fits-all approach to study CO_2 storage in geological formations, but that workflows must be adapted to the particular circumstances, constraints and questions asked.

It should also be pointed out that the concept of "workflow" supported by simplified models is here not limited to the optimization of usage scenarios. For instance, tools from statistical analysis could be applied to large number of simulations within a stochastic framework, in order to explore impact of uncertainty of specific parameters, or compute likelihoods of different outcomes. Another application is inverse modeling and value-of-information assessment, where the use of vertical equilibrium models and spill-point analysis have already been demonstrated in a study of the Sleipner storage site in [75].

Chapter 4

Introduction to the included papers

This dissertation includes a total of ten published or submitted research papers, all relating to the development and use of simplified mathematical models to study CO_2 storage issues. Although a number of examples are presented where the methods are applied on synthetic and real aquifer models, the research focus remains on the models themselves: their implementation and performance, impact of modeling choices, inclusion of new capabilities, and how they can fit together in a larger framework.

In Part II, the included articles are presented in approximately chronological order. Section 4.1 provides a general overview where articles are classified, related to each other, and have their main points described.

The author of this dissertation is the *primary* author for Paper I, Paper VII, Paper VIII, Paper IX and Paper X. The main ideas and mathematical developments, computer implementation, testing, interpretation of results and paper writing was here conducted by the author, in cooperation with supervisors and other coauthors. Moreover, the author of this dissertation is a *contributing* author to Paper II, Paper III, Paper IV, Paper V and Paper VI. For the work underlying these articles, the author contributed in all aspects of the process, including ideas and conceptual work, computer implementation, interpretation of results and paper writing.

4.1 Review of the papers

The articles included in this dissertation can be roughly sorted into three different categories, which do not always coincide with their chronological order: (1) Work on extending simplified models with new capabilities; (2) Study, implementation and testing of individual modeling approaches; and (3) Applications and workflows. Below, each of these categories is described and the contributions of the associated papers outlined.

Extending models with new capabilities (Paper I, Paper IX and Paper X)

Work in this category relates to the development and testing of new modeling capabilities for vertical equilibrium (VE) models.

Paper I, *Vertically Averaged Equations with Variable Density for CO_2 Flow in Porous Media*, focuses on the development of a mathematically consistent model of full CO_2 compressibility within the VE framework. Previously, vertical equilibrium

models for CO_2 storage have assumed constant fluid densities, or used simple approximations to model this effect. Density variations in the lateral directions are relatively easy to incorporate, whereas density variations in the vertical direction enter the integrals used to upscale the equations, and generally need to be determined by integrating an ordinary differential equation. The paper develops a mathematically correct VE formulation capable of describing both vertical and lateral density dependence on pressure and temperature, and studies the associated impacts using a few simulated examples. The work of this paper has led to a consistent model of variable CO_2 density in a VE setting. Nevertheless, the paper concludes than in most practical situations, a semi-compressible model where only lateral density variations are included will suffice to adequately model aquifer flow.

Paper IX, *Modeling geomechanical impact of CO2 injection*, proposes a simplified approach for modeling the full impacts of geomechanics on aquifer flow without having to solve a fully coupled poromechanical system. The proposed method, which in this paper is limited to the case of single-phase flow, is based on the observation that the flow equation is only influenced by the mechanics system through volumetric strain, as well as the assumption that the mechanics equations are linear. For any given, fixed set of boundary conditions and body forces, there is a linear, one-to-one relationship between fluid pressure and volumetric strain in the aquifer. Therefore, by computing the volumetric strain responses for a (possibly reduced) set of pressure basis functions, the volumetric field corresponding to any given pressure field can be approximately reconstructed. This allows for an efficient way of solving the flow equation while preserving the impact of the mechanical system, without having to couple with the mechanics equations when running a simulation. Numerical experiments presented in the paper show that the method reproduces the fully coupled solution to a good degree of accuracy, although this is often the case also for simpler "local" methods when multipliers for rock compressibility are appropriately chosen. A numerical procedure for computing these multipliers is also presented.

Paper X, *Vertical equilibrium flow models with fully coupled geomechanics for CO_2 storage modeling, using precomputed mechanical response functions*, extends the method proposed in Paper IX to two-phase flow, includes an additional coupling between mechanics and permeability, and applies the method in combination with a VE model to 2D and 3D CO_2 injection examples. Results are compared to similar results obtained from a fully-coupled poroelastic model, as well as from simple models based on local multipliers for rock compressibility. The results from the proposed model are very close to the corresponding results from the fully-coupled model. However, models using local multipliers also perform well, except for early times in the vicinity of the well. While the proposed model works as intended, the apparently good performance of simple, multiplier-based models for the considered problems suggests that a fully-coupled poroelastic approach might not always be needed to adequately account for the geomechanical impact of flow when modeling CO_2 injection scenarios. The paper therefore tentatively concludes that geomechanics may often be adequately represented using a simple multiplier model at the spatial and temporal scales for which VE modeling is useful, provided the multiplier is chosen correctly. A numerical procedure for obtaining this "optimal" multiplier is explained. On the other hand, the proposed method based on precomputed mechanical responses may be more useful when considering very local scales and short time-frames. In such situations, a full 3D model

for flow will generally be more appropriate than a VE formulation. Identifying situations where a fully coupled geomechanical model is indispensable to correctly model aquifer flow remains a topic of future investigation.

Individual modeling approaches (Paper II, Paper III and Paper IV)

Work in this category relates to the practical implementation, testing and discussion of vertical-equilibrium and spill-point models for CO_2 storage purposes. The models are implemented within a common software framework (`MRST-co2lab`), and the formulation used for the flow equations is based on the black-oil equations, similar to the practice of established commercial and academic 3D reservoir simulators.

Paper II, *Spill-Point Analysis and Structural Trapping Capacity in Saline Aquifers Using MRST-co2lab*,

This paper discusses spill-point analysis (c.f. Section 3.2), its implementation and its application to the study of long-term CO_2 migration. In previous literature, similar methods have been applied for the study of primary migration of hydrocarbons. The goal of the paper is to demonstrate and evaluate the method's usefulness in the context CO_2 storage modeling. In particular, this includes the prediction of long-term migration paths, identification of structural traps, bounds on storage capacity, and placement of injection wells. Such applications are demonstrated with practical examples based on real aquifer models, including the estimation of structural capacities for a number of North Sea formations. The article further explains how the method can be practically implemented, and discusses two different variants, based on different interpretation of the discrete grid geometry. The uncertainty of the method is assessed by comparing the different variants on the same datasets, by comparing outcomes using different grid resolutions, and by performing simple statistical analysis on results obtained from a large number of synthetically generated top surfaces. A greedy algorithm for choosing "optimal" injection points is also discussed.

The two papers Paper III, *Robust Simulation of Sharp-Interface Models for Fast Estimation of CO_2 Trapping Capacity in Large-Scale Aquifer Systems*, and Paper IV, *Fully-Implicit Simulation of Vertical-Equilibrium Models with Hysteresis and Capillary Fringe* should be considered as two parts of a whole. Together, these papers lay out a family of vertical equilibrium models that combine most of the physical effects previously published on such models in the context of CO_2 storage. Moreover, they describe an implementation of these models in a robust, fully-implicit framework, standard for commercial simulation software.

Paper III discusses sharp-interface models (i.e. neglecting fine-scale capillary pressure effects) that combine the effects of residual trapping, subscale caprock undulations, fully general functions for fluid densities and viscosities, and instant or rate-driven dissolution, based on a black-oil formulation of the equations. The paper discusses the relative merits of the *s*-formulation (upscaled saturation as one primary unknown) and the *h*-formulation (plume thickness as primary unknown). It also demonstrates the effects of the different trapping mechanisms, and compares computational performance, using examples based on real aquifer models.

Paper IV extends the models discussed in Paper III by introducing the effects of fine-scale capillarity (capillary fringe), and how it impacts and complicates the modeling of other effective quantities, including upscaled saturation, relative permeabilities,

capillary pressure and hysteretic behavior. Several practical models for the capillary fringe and hysteretic effects in upscaled relative permeability are proposed and compared, using both synthetic and real aquifer models.

Applications and workflows (Paper V, Paper VI, Paper VII and Paper VIII)

The common theme of the articles in this category is how the simplified models discussed in this thesis can be applied in isolation or in larger workflows for the investigation of questions related CO_2 storage.

Paper V, *Analysis of CO_2 Trapping Capacities and Long-Term Migration for Geological Formations in the Norwegian North Sea using MRST-co2lab*, employs the simplified models introduced in Paper II – IV to investigate issues related to CO_2 storage for real formations in the Norwegian North Sea, using simulation grid models constructed from [51]. The goal is to demonstrate how simplified analysis and modeling tools can play useful roles in addressing CO_2 storage questions.

The paper consist of three independent parts. The first part studies the impact of model resolution on the assessment of structural trapping for the Johansen and Utsira formations. The second part investigates a injection and long-term migration scenario for a hypothetical, upscaled storage scenario at the Sleipner site, also including a study of the potential impact of dissolution trapping. The third part of the papers demonstrates a workflow integrating nonlinear optimization, VE simulation and spill-point analysis in the search for an optimal configuration of a large-scale, aquifer-wide injection operation into the Utsira formation.

Paper VI, *A Simulation Workflow for Large-Scale CO_2 Storage in the Norwegian North Sea*, provides two examples of adapted workflows to estimate storage capacities and optimize utilization of two North Sea formations (Skade and Sandnes) with very different characteristics. The goal of the paper is to demonstrate the idea of workflows using simplified tools to study CO_2 storage, as well as how these must be adapted to the problem at hand.

The Skade formation is a shallow aquifer, with high permeability and very limited structural trapping. The Sandnes formation is a relatively thin aquifer at medium depths, with low permeability and a complex top surface offering large amounts of structural trapping. Both formations are considered as open in this study. The CO_2 retaining capacity of each aquifer, in term of structural, residual and dissolution trapping, is estimated using the methodology discussed in detail in Paper VII.

To investigate the practical utilization of the Skade aquifer, twenty well locations are chosen based on the goal of reaching as much structural trapping as possible (of which there is relatively little). The corresponding, optimal injection rates are obtained through adjoint-based nonlinear optimization with the goal of maximizing stored CO_2 while minimizing migration across aquifer boundaries.

It is shown that a similar methodology applied to the Sandnes formation would quickly lead to unacceptable levels of overpressure, caused both by its low permeability and its very large amount of structural trapping. As a consequence, a modified workflow is chosen, where wells are distributed according to a uniform pattern, and the objective function of the optimization routine is set up to discount CO_2 that ends up outside the catchment area of any structural trap. In addition, the optimization is carried out using a time-varying rate, and the simulated migration period is tweaked so

that most of the anticipated leakage will occur before simulation has ended.

Paper VII, *Reexamining CO_2 Storage Capacity and Utilization of the Utsira Formation*, presents a case-study of the Utsira formation, where the simplified methods available in `MRST-co2lab` are used to perform capacity estimation and to search for injection scenarios that exploits as much as possible of available storage.

The paper discusses various earlier capacity estimates for the Utsira aquifer, and points out the large variation in the resulting figures. It further introduces the concept of *retaining capacity* for open saline aquifers. The retaining capacity is intended to measure how much CO_2 an open aquifer could retain indefinitely through various trapping mechanisms (structural, residual and dissolution trapping), assuming that free-flowing CO_2 is free to leak across aquifer boundaries and should thus be discounted. As part of the process of computing the retaining capacity, a "capacity map" is also constructed, which indicates how much trapping capacity is associated with different parts of the aquifer.

Three different scenarios are considered for optimal utilization: a single-well scenario, a ten-well scenario and a scenario with multiple wells arranged in an array. Boundaries are considered fully open. The workflow for the two first scenarios are very similar to that of the Skade aquifer described in Paper VI, whereas the third scenario (well array) uses an elimination principle where wells that perform poorly after optimization are eliminated. This last approach manages to inject most CO_2, but also requires a larger number of wells than the other scenarios.

An important observation of the paper is that dynamical storage capacity (i.e., the storage capacity as determined by an "optimized" numerical simulation) will depend both on the time horizon considered and the total amount of leakage tolerated.

Paper VIII, *An Open-Source Toolchain for Simulation and Optimization of Aquifer-Wide CO_2 Storage*, is a relatively brief paper intended to present an overview of the current modeling capabilities of `MRST-co2lab`. It advocates the utility of flexible toolchains of simplified models in the context of studying CO_2 storage, basing its argument on the modeling challenges associated with the large spatial and temporal scales, wide range of possible aquifer characteristics and large data uncertainty.

The paper explains the principles behind and use of spill-point analysis and vertical equilibrium modeling. It further showcases some of the different modeling capabilities of the VE models in `MRST-co2lab`, and outlines what an integrated toolchain can look like.

This paper is the first publication that (briefly) presents results from VE models capable of modeling thermal effects within the `MRST-co2lab` framework. A simple energy conservation model that includes heat exchange with the aquifer over/underburden has been developed and preliminary results are shown in this paper for a scenario of hot CO_2 of magmatic origin invading a saline aquifer. At the time of writing this dissertation, thermal modeling in `MRST-co2lab` still remains work in progress, and further publications on this topic is anticipated.

Chapter 5

Concluding remarks and future work

Numerical simulation of CO_2 storage (or subsurface gas storage in general) comes with its own set of challenges compared to other common applications of reservoir modeling. These include the large spatial and temporal scales involved, the limited availability of data, the high density and viscosity ratio between the phases involved, and the limited ability to validate numerical experiments with measured data at time scales beyond the duration of the injection operation.

For the investigation of CO_2 storage problems, simplified models constitute one important set of tools, as they permit efficient exploration of possible long-term scenarios and model parameter sensitivities, at a low computational cost. Spill-point analysis provides a very rapid way of predicting long-term migration patterns over large distances, as well as providing information on structural trapping capacity and guidance on well placement. Vertical equilibrium models offer full numerical simulation capabilities with significantly reduced computational requirements, and whose results may rival those of 3D simulations as long as the underlying assumption of vertical equilibrium remains approximately valid. The robust implementation of VE models within a black-oil framework, that combines most of the modeling capabilities previously published for such models over the last decade, allows the study of complex scenarios that includes all the major trapping mechanisms and realistic fluid behavior. The potential of using simplified models within adaptive workflows has been demonstrated in several papers, using examples based on real aquifer models. An important part of the contribution of the work presented in this thesis consists of making the code and examples freely available to the research community in form of the open-software module `MRST-co2lab`. This has been done with the explicit intention to contribute to reproducible science and hopefully accelerate development on CO_2 storage modeling by allowing other researchers to rapidly test their ideas. Indeed, examples of this has already happened in practice [86].

The development and implementation of a simplified geomechanical model enables a flow simulator to include the full impact of mechanical deformation on flow, without coupling to a separate mechanics simulator. While this model has not yet been tested in larger workflows, it has been demonstrated to work correctly. In addition to eliminating the need for coupling separate simulators, the method can potentially reduce the computational cost significantly compared with a fully coupled geomechanical model.

Regarding possible future directions, significant grounds have already been covered on the topic of VE models, but possibilities for future research still remain. One ex-

ample is the inclusion of geochemical effects, in particular the possible impact of long-term mineral trapping. While the effect of mineral trapping is frequently neglected due to the long timescales involved, it is not clear whether the impact could be ignored in the context of long-term CO_2 migration studies. Also, the importance and speed of geochemical reactions is highly variable and site-dependent. Further effort could also be directed towards thermal effects in the context of VE models, in particular regarding the modeling of heat exchange between the target aquifer and its surroundings. There is also more work that could be done to facilitate the use of VE modeling for more general situations and modeling questions, for instance support of coupled multi-layer models, or for hybrid models that combine full 3D modeling in critical areas (around wells, faults, etc.) with dimensionally reduced VE modeling on the rest of the domain in some optimal manner. Automatic generation of such multi-layer or hybrid grids from realistic models could also be imagined.

For modeling based on spill-point analysis approach, as discussed in Section 3.2, the inherent geometric instability of the method could be addressed by further investigating these models in a probabilistic framework that explicitly incorporate uncertainty. In such a setting, long-term migration predictions could be presented in the form of a range of possible outcomes with associated probabilities, based on the geometric nature of the spill paths involved (long "valleys" versus flat "plains"), the complexity of the spill trees, and estimated level of uncertainty in the input data.

There are also interesting questions to look into regarding the practical use of simplified models in the context of larger workflows. One example of recent work in this direction include optimization of injection scenarios taking both pressure build-up, long-term leakage potential and economic constraints into account [8]. A very different application of simplified models is proposed in [75] where VE modeling is used as a key component in a workflow for estimating simulation model parameters from input data in the form of time-lapse seismic data, applied to the Sleipner injection site.

Finally, the storage of gases in the subsurface is not limited to CO_2 sequestration, but also include applications where gases are stored for later use (e.g., methane or hydrogen for energy purposes). Regarding CO_2, dual-purpose CO_2 storage has been proposed as a solution for the North Sea where the use of a geological formation for permanent storage of CO_2 is combined with shorter-term injection and extraction of CO_2 to provide buffer storage to support enhanced oil recovery operations. The role, benefit and possible adaptation of simplified computational models to this or other combined gas storage/utilization contexts constitutes another possible direction for future research.

Appendix A

Automatic differentiation in MRST

Motivation

The numerical derivative of an expression is frequently needed in computational applications, in particular for function optimization or root-finding problems where minima or roots are sought using iterative procedures. An important example is the use of the Newton-Raphson method to solve a system of n nonlinear equations in n unknowns: $f_i(\mathbf{x}) = 0,\ i = 1...n;\ \ \mathbf{x} \in \mathbb{R}^n$. The solution is here sought as the convergent limit of a sequence:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - J_F^{-1}(\mathbf{x}_k)F(\mathbf{x}_k), \tag{A.1}$$

where $\mathbf{x}_0$ is an initial guess, $F = \{f_i\}, i = 1...n$ and $J_F(\mathbf{x}) = \frac{\partial f_i(\mathbf{x})}{\partial x_j}$ is the Jacobian of F evaluated at the point $\mathbf{x}$. A prerequisite to use this method is the ability to compute all the elements of the Jacobian for any given $\mathbf{x}$.

One way of obtaining partial derivatives in the general case is to use numerical differentiation, i.e., for a small value ε compute:

$$\frac{\partial f_i(\mathbf{x})}{\partial x_j} \approx \frac{f_i([x_1, ...x_j + \varepsilon, ...x_N]) - f_i([x_1, ...x_j, ...x_N])}{\varepsilon}.$$

However, this approach introduces approximation error and can be highly numerically unstable. Moreover, each iteration of (A.1) will need another $2n^2$ function evaluations to obtain the Jacobian.

When each expression f_i can be explicitly expressed in terms of x, algebraic relations and functions with known derivatives, the analytic expression of the Jacobian can be deduced from standard differentiation rules. The resulting expression can then be implemented in code and provided as a separate function with no introduction of additional approximation error. However, this function can be complicated, cumbersome to derive, and time-consuming and error-prone to implement in numerical code.

Automatic differentiation is a technique that bypasses the need of deriving the symbolic expression of the derivative, while still providing results correct to machine precision. The general idea is to keep track of numerical derivatives throughout the series of basic operations needed to evaluate an expression.

Principle

Here, the basic principle of automatic differentiation (AD) is outlined in its forward form. Further discussion, including extension to higher-order derivatives, can be found in [74]. The underlying idea is not new (see [108] for an early, brief introduction). Examples of current reservoir simulation software implementing AD include AD-GPRS [115] and MRST.

AD employs the chain rule to keep track of derivatives as one evaluates a function $\mathbf{y} = F(\mathbf{x})$. Here, $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{x} \in \mathbb{R}^n$, and the evaluation of F consists of a finite set of elementary unitary or binary operations, $\{\xi^k(\cdot)\}_{k=1\ldots q}$, for which derivatives are trivial to obtain. Examples of ξ^k are basic arithmetic operations as well as simple mathematical functions such as $\sin(\cdot)$ and $\log(\cdot)$.

The computation of $\mathbf{y}$ from $\mathbf{x}$ can be seen as a series of operations involving a (possibly large) number of intermediate variables $\mathbf{z} \in \mathbb{R}^q$, which here is defined to include $\mathbf{y}$ so that $z_{q-m+1}\ldots z_q = \mathbf{y}$. Each intermediary value z_k only depend on $\mathbf{x}$ or on earlier indermediary values $z_{i\,(i<k)}$. By noting $Z_0 = \{x_i\}_{i=1\ldots n}$ and $Z_k = Z_{k-1} \cup \{z_k\}$ for $k > 0$, one can in other words write:

$$z_k = \begin{cases} \xi^k(u) & \text{for some } u \in Z_{k-1} \quad \text{if } \xi^k \text{ is unary} \\ \xi^k(u,v) & \text{for some } u,v \in Z_{k-1} \quad \text{if } \xi^k \text{ is binary.} \end{cases}$$

The partial derivative of z_k with respect to x_l is then obtained by:

$$\frac{\partial z_k}{\partial x_l} = \begin{cases} \frac{\mathrm{d}\xi^k}{\mathrm{d}u}\frac{\partial u}{\partial x_l} & \text{if } \xi^k \text{ is unary} \\ \frac{\partial \xi^k}{\partial u}\frac{\partial u}{\partial x_l} + \frac{\partial \xi^k}{\partial v}\frac{\partial v}{\partial x_l} & \text{if } \xi^k \text{ is binary.} \end{cases}$$

If at each step k the values of $\frac{\partial \xi^k}{\partial x_l}$ is stored, then one can immediately compute the derivatives of the new intermediate value z^k since the derivatives of ξ^k are trivial to obtain. The process culminates by computing the value and partial derivatives of $\mathbf{y}$ with respect to $\mathbf{x}$. A necessary starting point is of course to define $\frac{\partial x_i}{\partial x_j} = \delta_{ij}$, where δ_{ij} is the Kronecker delta symbol, equal to 1 for $i = j$ and zero otherwise.

In other words, for each intermediate variable z_k, it is necessary to keep track of its value *and* its partial derivatives with respect to $\mathbf{x}$. As an illustration, consider a simple example where $y \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^2$, and $y = F(\mathbf{x}) = (\log(x_1 + 2x_2))^2$. Further, consider that F and its partial derivatives $\nabla F = [\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}]$ is to be evaluated at point $\mathbf{x} = [3,1]$. In terms of intermediate variables, their values and partial derivatives, the sequence of computations is outlined in Table A.1.

Finally, one obtains the value of y (≈ 2.59) and its partial derivatives with respect to x_1 and x_2 (≈ 0.64 and ≈ 1.29, respectively). Note that the result was computed with no need for numerical differentiation nor explicit knowledge of the analytic derivative of the final expression.

Object orientation and implementation in MRST

In object-oriented programming languages that allow redefinition of basic computational operations, sequences such as described by Table A.1 can be implemented in a

Table A.1: Detailed sample AD calculation for expression $y = F(\mathbf{x}) = (\log(x_1 + 2x_2))^2$.

variable	value	$\nabla(.)$	$\xi(u)$ or $\xi(u,v)$	differentiation rule
x_1	3	[1, 0]		Kronecker delta
x_2	1	[0, 1]		Kronecker delta
$z_1 = 2x_2$	2	[0, 2]	$2u$	$2\nabla u$
$z_2 = x_1 + z_1$	5	[1, 2]	$u+v$	$\nabla u + \nabla v$
$z_3 = \log(z_2)$	≈ 1.61	[0.2, 0.4]	$\log(u)$	$u^{-1}\nabla u$
$z_4 = (z_3)^2 = y$	≈ 2.59	$\approx$ [0.64, 1.29]	u^2	$2u\nabla u$

compact and natural form using classes and operator overloading. The idea is to define a new numeric class to represent numbers when carrying out computations. An instance of this new class has two internal data fields: one to represent its value, and another to represent its partial derivatives with respect to some predefined set of m variables. Such an instance is here referred to as an AD-variable. Each basic computational operation and function is then redefined for AD-variables, to ensure that the outcome of the operation, a new AD-variable, has both the correct value *and* the correct derivative information. For example, if an AD-variable is represented using the pair $< v, v_x >$, where v denotes its value and v_x its partial derivatives, then the basic arithmetic operators +, -, $\times$ and / operators will be redefined as:

$$
\begin{aligned}
&< v, v_x > + < w, w_x > = < v + w, v_x + w_x > && \text{(addition)} \\
&< v, v_x > - < w, w_x > = < v + w, v_x - w_x > && \text{(subtraction)} \\
&< v, v_x > \times < w, w_x > = < v \times w, w \times v_x + v \times w_x > && \text{(multiplication)} \\
&< v, v_x > / < w, w_x > = < v/w, (w * v_x - v * w_x)/w^2 > && \text{(division).}
\end{aligned}
$$

The value of an AD-variable can be a scalar or a vector. For scalar AD-variables, the partial derivatives will be represented by a m-component vector. For vector-valued AD-variables, partial derivatives will be stored as the n by m sized Jacobian matrix, where n is the number of components in the value vector. To support vector-valued AD-variables, the redefinition of basic functions and operations must be implemented component-wise.

Once the new AD-class is defined and basic operations and functions redefined, computing the value and partial derivatives of expressions such as in the previous example becomes trivial, as will now be demonstrated using the `MRST` implementation of automatic differentiation. In `MRST`, a vector can be redefined as an AD variable using the function `initVariablesADI`. To implement the previous example, one can therefore specify the vector $X = [3, 1]^T$ and redefine it as an AD variable as follows:

```
X = [3,1]';              % define a vector of values
X = initVariablesADI(X) % redefine X as an AD-variable
```

```
X =
  ADI with properties:
    val: [2x1 double]
```

```
    jac: {[2x2 double]}
```

This redefined X has two fields: `val`, which contains the value, and `jac`, which contains the Jacobian. Its contents is verified as follows:

```
X.val  % display the value of X
```

```
ans =
     3
     1
```

```
full(X.jac{1})  % display the Jacobian of X as a full matrix
```

```
ans =
     1     0
     0     1
```

Initially, the partial derivatives of the two components of `X` are defined by the Kronecker delta. To compute the value and partial derivatives of $y = (\log(x_1 + 2x_2))^2$, one can simply type:

```
Y = log(X(1) + 2 * X(2)).^2
```

```
Y =
  ADI with properties:
    val: 2.5903
    jac: {[0.6438 1.2876]}
```

The same result is produced as was obtained from the explicit sequence of operations in Table A.1. Of course, this basic example is only scratching the surface. In general, the outcome of computations involving AD-variables will be vector-valued. `MRST` also supports grouping of AD-variables so that, if `X1` and `X2` are two numerical vectors, then the following example produces AD-variables with a common set of mutual partial derivatives:

```
X1 = [3 1]';
X2 = [1 2]';
[X1, X2] = initVariablesADI(X1, X2)
```

```
X1 =
  ADI with properties:
    val: [2x1 double]
    jac: {[2x2 double]  [2x2 double]}


X2 =
  ADI with properties:
```

```
val: [2x1 double]
jac: {[2x2 double]  [2x2 double]}
```

Each of the resulting AD-variables contain Jacobians with two separate blocks. The first block contains partial derivatives with respect to the components of `X1`, the second with respect to components in `X2`. The ability to group variables in this manner simplifies the implementation of discretized partial differential equations for a physical simulation. For instance, in a reservoir simulation setting the unknown variables represent discretized state variables such as total pressure or phase saturation, whose components conceptually belong together. In this way, the resulting expressions will contains separate blocks for derivatives with respect to the components of each involved state variable.

Appendix B

The Biot-Willis coefficient

The discussion below show how the Biot-Willis coefficient α in linear poroelasticity is linked both to the partial derivative of fluid content with respect to volumetric strain, and to the poroelastic definition of effective stress.

In Biot's poroelastic theory, four scalar quantities are linked by two constitutive equations [110]. The scalar quantities are: (1) volumetric strain $\varepsilon = \mathrm{tr}(\boldsymbol{\varepsilon})$; (2) mean normal stress $\sigma = \frac{1}{3}\mathrm{tr}(\boldsymbol{\sigma})$; (3) (increment of) fluid content ζ; and (4) fluid pressure p. The constitutive relations are:

$$\varepsilon = \frac{1}{K}\sigma + \frac{1}{H}p \tag{B.1}$$

$$\zeta = \frac{1}{H_1}\sigma + \frac{1}{R}p. \tag{B.2}$$

where $\frac{1}{K} = \frac{\partial \varepsilon}{\partial \sigma}$, $\frac{1}{H} = \frac{\partial \varepsilon}{\partial p}$, $\frac{1}{H_1} = \frac{\partial \zeta}{\partial \sigma}$ and $\frac{1}{R} = \frac{\partial \zeta}{\partial p}$. The assumption of the existence of a potential energy density $U = \frac{1}{2}(\sigma\varepsilon + p\zeta)$ entails that $H = H_1$. By simple algebraic manipulation of (B.1) and (B.2), one can then show that:

$$\zeta = \frac{K}{H}\varepsilon + \left(\frac{1}{R} - \frac{K}{H^2}\right)p. \tag{B.3}$$

Here, $\frac{K}{H} = \frac{\partial \zeta}{\partial \varepsilon}$ represents the change in fluid content per change in volumetric strain at constant pressure, and is known as the *Biot-Willis coefficient*, denoted α. The coefficient $\left(\frac{1}{R} - \frac{K}{H^2}\right) = \frac{\partial \zeta}{\partial p}$ represents the change in fluid content per change in pressure at constant strain, and is known as the *specific storage coefficient at constant strain*. It is denoted S_ε.

A different algebraic manipulation of (B.1) and (B.2) yields:

$$\varepsilon = \frac{1}{K}\left(\sigma + \frac{K}{H}p\right) = \frac{1}{K}\left(\sigma + \alpha p\right). \tag{B.4}$$

This relation shows that volumetric strain is proportional to the term $\sigma + \alpha p$, which is referred to as *effective stress* and denoted σ'. One can write:

$$\sigma' = \sigma + \alpha p, \tag{B.5}$$

with $\sigma' = K\varepsilon$.

For a full stress tensor, mean stress only depends on volumetric strain and deviatoric stress only depends on deviatoric strain. It is therefore possible to extend the definition of effective stress in (B.5) to the tensorial case by adding deviatoric components $\mathrm{dev}(\boldsymbol{\sigma'}) = \mathrm{dev}(\boldsymbol{\sigma}) = 2G\,\mathrm{dev}(\boldsymbol{\varepsilon})$:

$$\begin{aligned}\boldsymbol{\sigma'} &= \sigma'\mathbf{I} + \mathrm{dev}(\boldsymbol{\sigma'}) \\ &= (\sigma + \alpha p)\mathbf{I} + \mathrm{dev}(\boldsymbol{\sigma}) \\ &= \boldsymbol{\sigma} + \alpha p\mathbf{I}.\end{aligned}$$

Moreover, since $\sigma' = K\varepsilon$ and $\mathrm{dev}(\boldsymbol{\sigma}) = 2G\,\mathrm{dev}(\boldsymbol{\varepsilon})$, one has:

$$\boldsymbol{\sigma'} = K\varepsilon\,\mathbf{I} + 2G\,\mathrm{dev}(\boldsymbol{\varepsilon}). \tag{B.6}$$

Appendix C

Installing MRST-co2lab and running interactive examples

C.1 Downloading and setting up MRST and MRST-co2lab

To run the examples presented in Chapter 2 and Chapter 3, a MATLAB installation is required, and it is necessary to have downloaded and installed the MATLAB Reservoir Simulation Toolbox, release 2016b or later. MRST is an open source software collection developed and maintained by the Computational Geosciences group of SINTEF ICT, Department of Applied Mathematics. New revisions are released under the GNU General Public License (GPL) two times per year.

C.1.1 Downloading MRST and MRST-co2lab

The latest version of MRST can be obtained from:

https://www.sintef.no/projectweb/mrst/downloadable-resources/

MRST-co2lab is bundled as part of this release, so does not require a separate download. After downloading the zipped file, extract it where you want your MRST installation to be. To use it, first launch MATLAB, change the current path to the MRST root directory (named mrst-XXXXX, where XXXXX refers to the release number) and enter the following commands to activate MRST and load the MRST-co2lab module:

```
startup               % activate MRST
mrstModule add co2lab % tell MRST to load the co2lab-module
```

C.1.2 Downloading datasets

The next step is to download the sample datasets. Provided that MRST-co2lab has already been loaded as explained above, this can be done by running the following command:

```
downloadDataSets
```

You will be queried which datasets you want. To run the examples of Chapter 2 and Chapter 3, as well as the interactive examples briefly presented in the following Appendix, you should make sure to at least download the "Johansen" and "CO2 Atlas" datasets. Once the datasets have been downloaded, you should be able to run all the code examples from Chapter 2 and Chapter 3, as well as the interactive examples presented below.

The names of the aquifer models available from the CO2 Atlas dataset can be obtained in list form as follows:

```
getAtlasGrid()
```

```
Available grids are:
        - Brentgrp
        - Brynefm
        - Cookfm
        - Dunlingp
        - Fensfjordfm
        - Gassumfm
        - Huginfm
        - Johansenfm
        - Jurassic
        - Krossfjordfm
        - Paleocene
        - Pliocenesand
        - Sandnesfm
        - Skadefm
        - Sleipnerfm
        - Sognefjordfm
        - Statfjordfm
        - Ulafm
        - Utsirafm
```

C.2 Exploring storage capacity with `exploreCapacity`

The program `exploreCapacity` allows the user to interactively explore the impact of various parameters on estimated trapping capacity of the aquifers provided by the "CO2 Atlas" dataset. "Capacity" here refers to the maximum retaining capacity as described in Paper VII. The application is launched by the following command:

```
exploreCapacity
```

Using the graphical interface, shown in Figure C.1, the user can choose among a selection of geological formations, set some key parameters using the sliders, and evaluate the impact on total trapping capacity, broken down by trapping mechanism. The user can also choose to visualize different quantities, including reachable structural capacity (c.f. Paper VII), structural trap capacity, total capacity, CO_2 density, topography,

caprock pressure and temperature, formation thickness and formation depth.

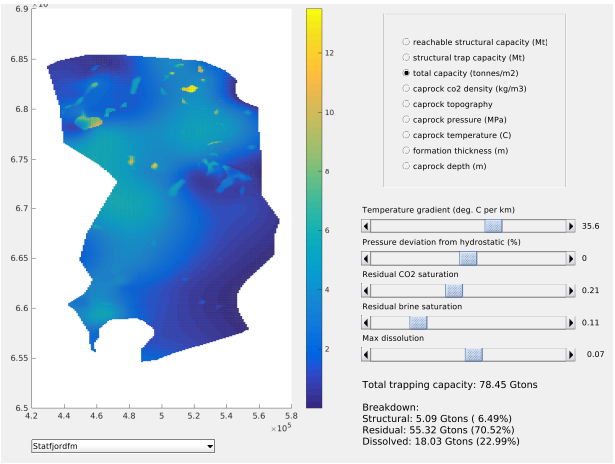


Figure C.1: `exploreCapacity` graphical user interface.

Additional parameters can be specified by launching the application while specifying additional options, on the form:

```
exploreCapacity('option1', value1, 'option2', value2, ...)
```

Some relevant options are:

- `grid_coarsening` - the level of down-sampling of the grid. A value of 1 means that the highest level of resolution is used. Higher values yield progressively coarser grids. Default value is 2.
- `default_formation` - name of the formation to load at startup. Default is 'Utsirafm'. (Other possible names can be found in the application's drop-down menu).
- `water_density` - Density of brine, in kg/m^3. Default is 1000. Used in the computation of hydrostatic pressure.
- `seafloor_depth` - Depth of the sea floor, in meters. Default is 100 m. Used in combination with `seafloor_temp` and `temp_gradient` when computing temperature at caprock level.
- `seafloor_temp` - Temperature at sea floor, in °C. Default is 7°. Used in the computation of temperature at caprock level.
- `temp_gradient` - Temperature gradient, in °C/km depth. Default is 35.6°C/km. Used in the computation of temperature at caprock level.

C.3 Setting up and running simulations with `exploreSimulation`

`exploreSimulation` is an interactive script for rapidly setting up and running vertical equilibrium simulations and inspecting the results, using simulation grids of the North Sea aquifers from the "CO2 Atlas" dataset. It is launched using the command:

```
exploreSimulation
```

When the application is started, the user is presented with a graphical interface (Figure C.2) for specifying the main details of a simulation scenario. This includes selection of target aquifer, specification of injection well positions, rates, boundary conditions, the duration of injection and migration periods, and the number of timesteps. In addition, the user can specify whether or not to include capillarity, subscale trapping and/or dissolution in the simulation. When all is set up, the "launch" button will start the simulation, which will usually require a few minutes to complete.

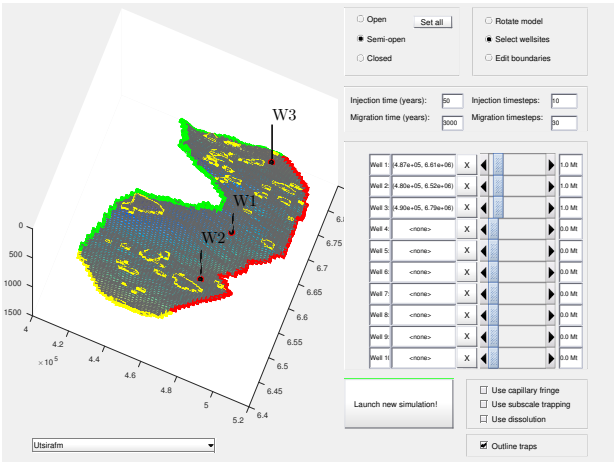


Figure C.2: `exploreSimulation` - graphical interface for setting up and launching a simulation.

After the simulation has completed, the result can be visually inspected using a second graphical interface (Figure C.3). This interface allows the user to specify which timestep and which state variable (saturation, pressure, etc.) to visualize. The result is plotted on top of a 3D grid, which is possible to rotate in any direction. The trapping inventory[1] associated with the simulation is shown in a separate window (Figure C.4).

[1] A trapping inventory is a graph tracking the CO_2 trapping state as a function of time.

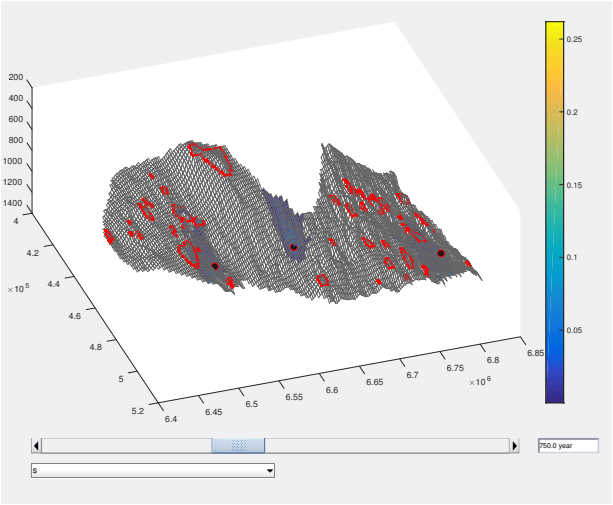


Figure C.3: `exploreSimulation` - graphical interface for inspecting the result of a simulation.

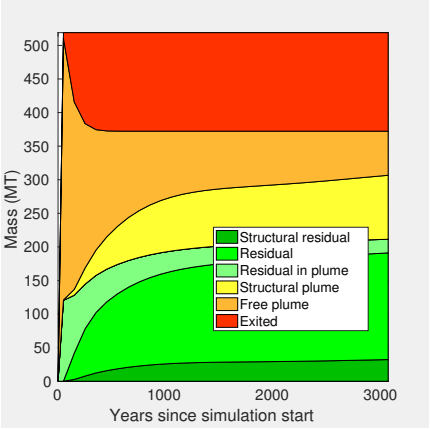


Figure C.4: exploreSimulation - trapping inventory display.

Additional parameters can be specified at the command line when launching the application, on the form:

```
exploreSimulation('option1', value1, 'option2', value2, ...)
```

Some relevant options are:

- `grid_coarsening` - the level of down-sampling of the grid. A value of 1 means that the highest level of resolution is used. Higher values yield progressively coarser grids. Default value is 2.

- `seafloor_depth` - Depth of the sea floor, in meters. Default is 100 m. Used in combination with `seafloor_temp` and `temp_gradient` when computing temperature at caprock level.
- `seafloor_temp` - Temperature at sea floor, in °C. Default is 7°. Used in the computation of temperature at caprock level.
- `temp_gradient` - Temperature gradient, in °C/km depth. Default is 35.6°C/km. Used in the computation of temperature at caprock level.
- `water_density` - Density of brine, in kg/m^3. Default is 1000.
- `dis_max` - Maximum volume of CO_2 that can be dissolved per volume of brine, measured at reference conditions. Default is 0.07.
- `max_num_wells` - maximum number of injector sites allowed. Default value is 10.
- `max_rate` - Maximum allowed injection rate, in terms of volume of CO_2 per second, measured at reference conditions.
- `water_compr_val` - Compressibility coefficient for water. Default value is 4.3e-10 Pa^{-1}.
- `pvMult` - Pore volume compressibility coefficient. Default value is 1e-10 Pa^{-1}.
- `water_residual` - Residual saturation of brine, as fracture of pore space. Default value is 0.11.
- `co2_residual` - Residual saturation of CO2, as fracture of pore space. Default value is 0.21.
- `well_radius` - Radius of the well-bore. Default is 0.3 meters.
- `C` - Scaling factor for a Brooks-Corey capillary pressure curve (only relevant when the capillary fringe option is utilized). Default value is 0.1.
- `outside_distance` - A fictive distance used to compute the flow resistance across semi-open boundaries. Default is 100 kilometers.
- `savefile` - If specified, the simulation result will be saved to the provided file name.

C.4 Exploring structural traps with `interactiveTrapping`

The `interactiveTrapping` program allows the user to explore structural traps and spill-paths on North Sea aquifer models provided by the "CO2 Atlas" dataset. The program is invoked as follows:

```
interactiveTrapping(formation_name)
```

where `formation_name` is the name of the target aquifer model. (A list of model names can be obtained by running the `getAtlasGrid` script without parameters).

The user is presented with a top view of the aquifer (Figure C.5). By clicking on a point in the aquifer using the *left* mouse button, the spill path (including traps) from that point upwards is highlighted. In other words, the traps reachable by gravity-driven migration from injection at the selected point will be highlighted. By clicking on a point in the aquifer using the *middle* mouse button, the part of the spill tree upslope of the trap associated with that point is highlighted. In either case, additional information is shown at the right side of the graphical window, including the individual volumes of all highlighted traps, as well as the percentage of total aquifer structural trap capacity they collectively cover. Clicking on a trap using the *right* mouse button brings up a separate graphical window where the trap in question is visualized up close (Figure C.6, left).

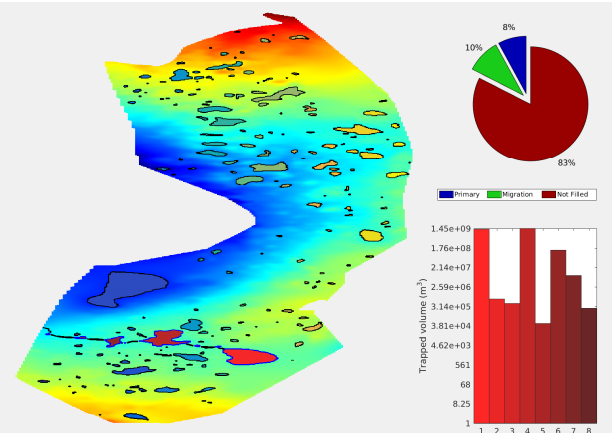


Figure C.5: `interactiveTrapping` exploration window. On the left side is a top view of the formation, with structural traps outlined.

Once an injection point has been selected (using the left mouse button), the user has the option of running a simple numerical simulation of an injection and migration scenario. The simulation is based on a very simplified, incompressible vertical-equilibrium description. This is done by clicking the button labeled `Sim`. At this point, the user is presented with an additional interface where various simulation parameters can be specified (Figure C.6, right). The simulation is then launched by clicking the `Simulate` button. During and after simulation, the CO_2 distribution in the aquifer for the current timestep is visualized in various ways in a separate panel (Figure C.7), which also indicates the distribution of CO_2 over various trapping states.

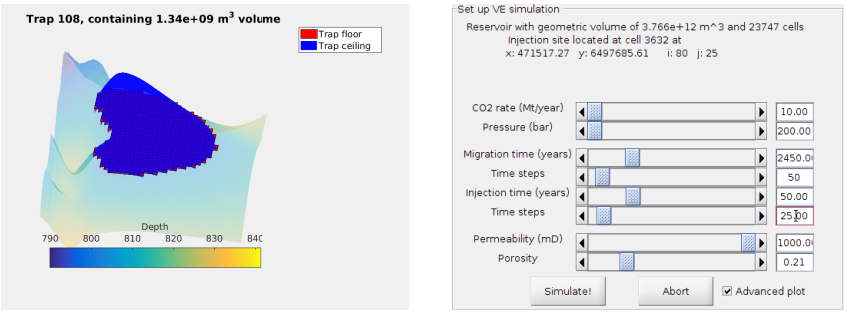


Figure C.6: *Left:* Plot of an individual trap, produced by right-clicking in the exploration window shown in Figure C.5. *Right:* Dialog box for specifying simulation parameters.

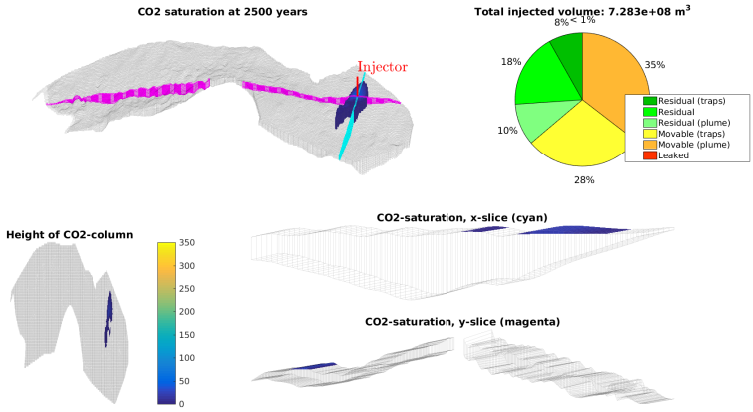


Figure C.7: Visualization of a simulation outcome using `interactiveTrapping`.

Additional parameters can be specified when the application is launched, on the following form:

```
interactiveTrapping(name, 'option1', value1, 'option2', value2, ...)
```

Some relevant options are:

- `coarsening` - the level of down-sampling of the grid. A value of 1 means that the highest level of resolution is used (default). Higher values yield progressively coarser grids, which speeds up the simulation.
- `spillregions` - if set to true, spill regions will be indicated along with their associated traps. Default is false.
- `method` - chooses between two different implementations of the spill-point analysis algorithm: one cell-based ('cell') and one node-based ('node'). Default is 'cell'.

Bibliography

[1] CO_2 emissions from fuel combustion highlights 2015. International Energy Agency (IEA), 2015. http://www.iea.org/publications/freepublications/publication/co2-emissions-from-fuel-combustion-highlights-2015.html. 1.1

[2] The global status of CCS 2015, summary report. Global CCS Institute, 2015. https://www.globalccsinstitute.com/publications/global-status-ccs-2015-summary-report. 1.1

[3] The Quest for less CO_2: learning from CCS implementation in Canada. A case study on Shell's Quest CCS project. Shell, 2015. http://www.globalccsinstitute.com/publications/shell-quest-case-study. 1.1

[4] Energy technology perspectives 2016 - executive summary. International Energy Agency (IEA), 2016. http://www.iea.org/publications/freepublications/publication/energy-technology-perspectives-2016---executive-summary---english-version.html. 1.1

[5] tNavigator. RFD - Rock Flow Dynamics, 2016. http://www.rfdyn.com/technology/. 2.2.4

[6] AAVATSMARK, I. An introduction to multipoint flux approximations for quadrilateral grids. *Computational Geosciences 6*, 3-4 (2002), 405–432. 2.2.4

[7] AAVATSMARK, I., KOMETA, B. K., GASDA, S. E., SANDVE, T. H., AND NILSEN, H. M. A generalized cubic equation of state with application to pure CO_2 injection in aquifers. *Computational Geosciences 20*, 3 (2016), 623–635, doi: 10.1007/s10596-015-9537-0. 2.1.3

[8] ALLEN, R. D., NILSEN, H. M., ANDERSEN, O. A., AND LIE, K.-A. On obtaining optimal well rates and placement for CO_2 storage. In *Proceedings of ECMOR XV–15th European Conference on the Mathematics of Oil Recovery* (Amsterdam, Netherlands, 29 August–1 September 2016), EAGE. 3.4, 3.4.5, 5

[9] ARUFFO, C., RODRIGUEZ-HERRERA, A., TENTHOREY, E., KRZIKALLA, F., MINTON, J., AND HENK, A. Geomechanical modelling to assess fault integrity at the CO2CRC Otway Project, Australia. *Australian Journal of Earth Sciences 61*, 7 (2014), 987–1001. 2.2.5

[10] BACHU, S. Screening and ranking of sedimentary basins for sequestration of CO_2 in geological media in response to climate change. *Environmental Geology 44*, 3 (2003), 277–289, doi: 10.1007/s00254-003-0762-9. 1.2

[11] BEAR, J. *Dynamics of Fluids in Porous Media*. Dover, 1988. 2.1.1, 3.1

[12] BELL, I. H., WRONSKI, J., QUOILIN, S., AND LEMORT, V. Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp. *Ind. Eng. Chem. Res. 53*, 6 (2014), 2498–2508, doi: 10.1021/ie4033999. 2.1.3

[13] BENNION, D. B., BACHU, S., ET AL. Dependence on temperature, pressure, and salinity of the IFT and relative permeability displacement characteristics of CO_2 injected in deep saline aquifers. In *SPE Annual Technical Conference and Exhibition* (2006), Society of Petroleum Engineers. 2.2.2, 2.2.2

[14] BISSELL, R., VASCO, D., ATBI, M., HAMDANI, M., OKWELEGBE, M., AND GOLDWATER, M. A full field simulation of the In Salah gas production and CO_2 storage project using a coupled geo-mechanical and thermal fluid flow simulator. *Energy Procedia 4* (2011), 3290–3297. 2.2.5

[15] BJØRNARÅ, T. I., NORDBOTTEN, J. M., AND PARK, J. Vertically integrated models for coupled two-phase flow and geomechanics in porous media. *Water Resources Research 52*, 2 (2016), 1398–1417, doi: 10.1002/2015WR017290. 3.3

[16] BRENIER, Y., AND JAFFRÉ, J. Upstream differencing for multiphase flow in reservoir simulation. *SIAM J. Numer. Anal. 28*, 3 (1991), 685–696, doi: 10.1137/0728036. 2.2.4

[17] BREZZI, F., AND FORTIN, M. *Mixed and Hybrid Finite Element Methods*, vol. 15 of *Springer Series in Computational Mathematics*. Springer-Verlag, New York, 1991. 2.2.4

[18] BROOKS, R. H., AND COREY, A. T. Hydraulic properties of porous media. Tech. rep., Colorado State University, 1964. 2.2.2

[19] BURNSIDE, N., AND NAYLOR, M. Review and implications of relative permeability of CO_2/brine systems and residual trapping of CO_2. *International Journal of Greenhouse Gas Control 23* (2014), 1 – 11, doi: http://dx.doi.org/10.1016/j.ijggc.2014.01.013. 2.2.2

[20] CAVANAGH, A. J., AND HASZELDINE, R. S. The Sleipner storage site: Capillary flow modeling of a layered CO_2 plume requires fractured shale barriers within the Utsira Formation. *International Journal of Greenhouse Gas Control 21* (2014), 101 – 112, doi: http://dx.doi.org/10.1016/j.ijggc.2013.11.017. 3.2.1, 3.2.1

[21] CAVANAGH, A. J., HASZELDINE, R. S., AND NAZARIAN, B. The Sleipner CO_2 storage site: using a basin model to understand reservoir simulations of plume dynamics. *First Break 33*, June (2015), 61–68. 3.1, 3.2.1

[22] CELIA, M. A., BACHU, S., NORDBOTTEN, J. M., AND BANDILLA, K. W. Status of CO2 storage in deep saline aquifers with emphasis on modeling approaches and practical simulations. *Water Resources Research 51*, 9 (2015), doi: 10.1002/2015WR017609. 1.1

[23] CLASS, H., EBIGBO, A., HELMIG, R., DAHLE, H. K., NORDBOTTEN, J. M., CELIA, M. A., AUDIGANE, P., DARCIS, M., ENNIS-KING, J., FAN, Y., FLEMISCH, B., GASDA, S. E., JIN, M., KRUG, S., LABREGERE, D., BENI, A. N., PAWAR, R. J., SBAI, A., THOMAS, S. G., TRENTY, L., AND WEI, L. A benchmark study on problems related to CO_2 storage in geologic formations. *Comput. Geosci. 13*, 4 (2009), 409–434, doi: 10.1007/s10596-009-9146-x. 3.1

[24] CMG. *User's Guide GEM*. Computer Modeling Group, Ltd., 2009. 1.1

[25] COATS, K., DEMPSEY, J., AND HENDERSON, J. The use of vertical equilibrium in two-dimensional simulation of three-dimensional reservoir performance. *SPE Journal 11-1* (1971), 63 – 71, doi: 10.2118/2797-PA. 3.1

[26] COREY, A. T. The interrelation between gas and oil relative permeabilities. *Producers monthly 19*, 1 (1954), 38–41. 2.2.2

[27] COURT, B., BANDILLA, K. W., CELIA, M. A., JANZEN, A., DOBOSSY, M., AND NORDBOTTEN, J. M. Applicability of vertical-equilibrium and sharp-interface assumptions in CO2 sequestration modeling. *International Journal of Greenhouse Gas Control 10* (2012), 134–147, doi: 10.1016/j.ijggc.2012.04.015. 3.1, 3.1

[28] DA VEIGA, L., LIPNIKOV, K., AND MANZINI, G. *The Mimetic Finite Difference Method for Elliptic Problems*. MS&A. Springer International Publishing, 2013. 2.2.4

[29] DAVIES, J., DAVIES, D., ET AL. Stress-dependent permeability: characterization and modeling. In *SPE Annual Technical Conference and Exhibition* (1999), Society of Petroleum Engineers. 2.2.5

[30] DEAN, R. H., GAI, X., STONE, C. M., MINKOFF, S. E., ET AL. A comparison of techniques for coupling porous flow and geomechanics. *SPE Journal 11*, 01 (2006), 132–140. 2.2.5, 3.3

[31] DESA. *Energy Statistics Yearbook*. United Nations, 2013. 1.1

[32] DIETZ, D. A theoretical approach to the problem of encroaching and by-passing edgewater. *Proceedings Akademie van Wetenschappen 56-B* (1953), 83 – 94. 3.1

[33] DIXON, T., HERZOG, H., TWINNING, S., CAVANAGH, A., AND NAZARIAN, B. A new and extended Sleipner benchmark model for CO_2 storage simulations in the Utsira Formation. *Energy Procedia 63* (2014), 2831 – 2835, doi: http://dx.doi.org/10.1016/j.egypro.2014.11.305. 3.1

[34] DOSTER, F., NORDBOTTEN, J., AND CELIA, M. Impact of capillary hysteresis and trapping on vertically integrated models for CO_2 storage. *Advances in Water Resources 62, Part C* (2013), 465 – 474, doi: http://dx.doi.org/10.1016/j.advwatres.2013.09.005. Computational Methods in Geologic CO_2 Sequestration. 3.1

[35] DU PLESSIS, E., NORDBOTTEN, J. M., GASDA, S. E., AND DAHLE, H. K. Influence of capillary pressure and trapping hysteresis on large-scale CO_2 migration. *Journal of Coupled Systems and Multiscale Dynamics 1*, 4 (2013), 442–458. 3.1

[36] EIGESTAD, G., DAHLE, H., HELLEVANG, B., RIIS, F., JOHANSEN, W., AND ØIAN, E. Geological modeling and simulation of CO_2 injection in the Johansen formation. *Comput. Geosci. 13*, 4 (2009), 435–450, doi: 10.1007/s10596-009-9153-y. 2, 2.1.1, 2.1.4

[37] EIKEN, O., 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* (2011), 5541–5548, doi: 10.1016/j.egypro.2011.02.541. 10th International Conference on Greenhouse Gas Control Technologies. 3.1

[38] ENNIS-KING, J., AND PATERSON, L. Role of convective mixing in the long-term storage of carbon dioxide in deep saline formations. *SPE Journal 10*, 03 (2005), 349–356. 1.2, 2.2.3, 3.1

[39] FLEMISCH, B., FRITZ, J., HELMIG, R., NIESSNER, J., AND WOHLMUTH, B. Dumux: a multi-scale multi-physics toolbox for flow and transport processes in porous media. In *ECCOMAS Thematic Conference on Multiscale Computational Methods for Solids and Fluids* (2007). 1.1

[40] GASDA, S., GRAY, W., AND DAHLE, H. *Vertically integrated models with coupled thermal processes*. European Association of Geoscientists and Engineers, EAGE, 2014. 3.1

[41] GASDA, S., NORDBOTTEN, J., AND CELIA, M. Vertical equilibrium with subscale analytical methods for geological CO_2 sequestration. *Computational Geosciences 13*, 4 (2009), 469–481. 3.1, 3.1, 3.1

[42] GASDA, S. E., NILSEN, H. M., AND DAHLE, H. K. Impact of structural heterogeneity on upscaled models for large-scale CO_2 migration and trapping in saline aquifers. *Advances in Water Resources* (2013), doi: 10.1016/j.advwatres.2013.05.003. 3.1

[43] GASDA, S. E., NILSEN, H. M., DAHLE, H. K., AND GRAY, W. G. Effective models for CO_2 migration in geological systems with varying topography. *Water Resources Research 48*, 10 (2012), doi: 10.1029/2012WR012264. 3.1

[44] GASDA, S. E., NORDBOTTEN, J. M., AND CELIA, M. A. The impact of local-scale processes on large-scale CO_2 migration and immobilization. *Energy Procedia 4*, 0 (2011), 3896 – 3903, doi: 10.1016/j.egypro.2011.02.327. 10th International Conference on Greenhouse Gas Control Technologies. 3.1

[45] GASDA, S. E., NORDBOTTEN, J. M., AND CELIA, M. A. Vertically averaged approaches for CO_2 migration with solubility trapping. *Water Resources Research 47*, 5 (2011), doi: 10.1029/2010WR009075. 3.1

[46] GASDA, S. E., NORDBOTTEN, J. M., AND CELIA, M. A. Application of simplified models to CO_2 migration and immobilization in large-scale geological systems. *Int. J. Greenh. Gas Control 9* (2012), 72–84, doi: 10.1016/j.ijggc.2012.03.001. 3.1

[47] GIRAULT, V., KUMAR, K., AND WHEELER, M. F. Convergence of iterative coupling of geomechanics with flow in a fractured poroelastic medium. Tech. rep., Technical Report ICES REPORT 15-05, The Institute for Computational Engineering and Sciences The University of Texas at Austin, Austin, Texas 78712, 2015. 2.2.5, 3.3

[48] GUO, B., BANDILLA, K. W., DOSTER, F., KEILEGAVLEN, E., AND CELIA, M. A. A vertically integrated model with vertical dynamics for CO_2 storage. *Water Resources Research 50*, 8 (2014), 6269–6284, doi: 10.1002/2013WR015215. 3.1

[49] GUO, B., BANDILLA, K. W., KEILEGAVLEN, E., DOSTER, F., AND CELIA, M. A. Application of vertically-integrated models with subscale vertical dynamics to field sites for CO_2 sequestration. *Energy Procedia 63* (2014), 3523–3531, doi: 10.1016/j.egypro.2014.11.381. 3.1

[50] GUO, B., BANDILLA, K. W., NORDBOTTEN, J. M., CELIA, M. A., KEILEGAVLEN, E., AND DOSTER, F. A multiscale multilayer vertically integrated model with vertical dynamics for CO_2 sequestration in layered geological formations. *Water Resources Research*, doi: 10.1002/2016WR018714. 3.1, 7

[51] HALLAND, E. K., JOHANSEN, W. T., AND RIIS, F., Eds. *CO_2 Storage Atlas: Norwegian North Sea*. Norwegian Petroleum Directorate, P. O. Box 600, NO–4003 Stavanger, Norway, Dec. 2011. 1.2, 3.2.3, 4.1

[52] HESSE, M. A., ORR, F. M., AND TCHELEPI, H. A. Gravity currents with residual trapping. *J. Fluid. Mech. 611* (2008), 35–60. 1.1

[53] HOLT, T., AND LINDEBERG, E. Thermal power — without greenhouse gases and with improved oil recovery. *Energy Conversion and Management 33*, 5–8 (1992), 595 – 602, doi: 10.1016/0196-8904(92)90061-Z. Proceedings of the First International Conference on Carbon Doxide Removal. 1.1

[54] HUPPERT, H. E., AND NEUFELD, J. A. The fluid mechanics of carbon dioxide sequestration. *Annual Review of Fluid Mechanics 46* (2014), 255–272. 1.2, 3.1

[55] IPCC. *Assessing Transformation Pathways. Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. IPCC, Cambridge University Press, 2014. 1.1

[56] IPCC. *IPCC, 2014: Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. IPCC, Geneva, Switzerland, 151 pp., 2014. 1.1, 1.4, 1.2

[57] JANSEN, J. D. Adjoint-based optimization of multi-phase flow through porous media – a review. *Computers & Fluids 46*, 1, SI (JUL 2011), 40–51, doi: 10.1016/j.compfluid.2010.09.039. 3.4.3

[58] JUANES, R., SPITERI, E. J., ORR JR., F. M., AND BLUNT, M. J. Impact of relative permeability hysteresis on geological CO_2 storage. *Water Resour. Res. 42*, W12418 (2006), doi: 10.1029/2005WR004806. 2.2.2

[59] KIM, J. *Sequential methods for coupled geomechanics and multiphase flow*. PhD thesis, Stanford University, 2010. 2.2.5, 3.3

[60] KOIDE, H., TAZAKI, Y., NOGUCHI, Y., NAKAYAMA, S., IIJIMA, M., ITO, K., AND SHINDO, Y. Subterranean containment and long-term storage of carbon dioxide in unused aquifers and in depleted natural gas reservoirs. *Energy Conversion and Management 33*, 5–8 (1992), 619 – 626, doi: 10.1016/0196-8904(92)90064-4. Proceedings of the First International Conference on Carbon Doxide Removal. 1.1

[61] KORBØL, R., AND KADDOUR, A. Sleipner vest CO_2 disposal - injection of removed CO_2 into the Utsira formation. *Energy Conversion and Management 36*, 6–9 (1995), 509 – 512, doi: 10.1016/0196-8904(95)00055-I. Proceedings of the Second International Conference on Carbon Dioxide Removal. 1.1

[62] KREVOR, S., BLUNT, M. J., BENSON, S. M., PENTLAND, C. H., REYNOLDS, C., AL-MENHALI, A., AND NIU, B. Capillary trapping for geologic carbon dioxide storage – from pore scale physics to field scale implications. *International Journal of Greenhouse Gas Control 40* (2015), 221 – 237, doi: 10.1016/j.ijggc.2015.04.006. Special Issue commemorating the 10th year anniversary of the publication of the Intergovernmental Panel on Climate Change Special Report on CO_2 Capture and Storage. 1.2, 2.2.2, 2.2.2

[63] LICHTNER, P. C., HAMMOND, G. E., LU, C., KARRA, S., BISHT, G., ANDRE, B., MILLS, R. T., AND KUMAR, J. PFLOTRAN user manual. Tech. rep., 2013. 2.2.4

[64] LIE, K.-A. *An Introduction to Reservoir Simulation Using MATLAB: User guide for the Matlab Reservoir Simulation Toolbox (MRST)*, 2nd ed. SINTEF ICT, `http://www.sintef.no/Projectweb/MRST/publications`, December 2015. 1.3, 2.1.1, 2.1.1

[65] LIGAARDEN, I. S., AND NILSEN, H. M. Numerical aspects of using vertical equilibrium models for simulating CO_2 sequestration. In *Proceedings of ECMOR XII–12th European Conference on the Mathematics of Oil Recovery* (Oxford, UK, 6–9 September 2010), EAGE. 3.1

[66] LONGUEMARE, P., MAINGUY, M., LEMONNIER, P., ONAISI, A., GÉRARD, C., AND KOUTSABELOULIS, N. Geomechanics in reservoir simulation: overview of coupling methods and field case study. *Oil & Gas Science and Technology 57*, 5 (2002), 471–483. 2.2.5, 3.3

[67] LOTHE, A., EMMEL, B., GRØVER, A., AND BERGMO, P. CO_2 storage modelling and capacity estimation for the Trøndelag Platform, offshore Norway - using a basin modelling approach. *Energy Procedia 63* (2014), 3648 – 3657, doi: http://dx.doi.org/10.1016/j.egypro.2014.11.394. 3.2.1, 3.2.3

[68] MALDAL, T., AND TAPPEL, I. CO_2 underground storage for Snøhvit gas field development. *Energy 29*, 9–10 (2004), 1403 – 1411, doi: 10.1016/j.energy.2004.03.074. 6th International Conference on Greenhouse Gas Control Technologies. 1.1

[69] MARTIN, J. C. Partial integration of equations of multiphase flow. *Soc. Pet. Eng. J. Dec* (1968), 370–380. 3.1

[70] MIKELIĆ, A., AND WHEELER, M. F. Convergence of iterative coupling for coupled flow and geomechanics. *Computational Geosciences 17*, 3 (2013), 455–461. 2.2.5, 3.3

[71] MIRI, R., AND HELLEVANG, H. Salt precipitation during CO_2 storage — a review. *International Journal of Greenhouse Gas Control 51* (2016), 136 – 147, doi: 10.1016/j.ijggc.2016.05.015. 1.2

[72] The MATLAB Reservoir Simulation Toolbox, version 2015b, 12 2015. `http://www.sintef.no/MRST/`. 1.3

[73] MUKHOPADHYAY, S., DOUGHTY, C., BACON, D., LI, J., WEI, L., YAMAMOTO, H., GASDA, S., HOSSEINI, S. A., NICOT, J.-P., AND BIRKHOLZER, J. T. The Sim-SEQ project: Comparison of selected flow models for the S-3 site. *Transport in Porous Media 108*, 1 (2015), 207–231, doi: 10.1007/s11242-014-0361-0. 3.1

[74] NEIDINGER, R. Introduction to automatic differentiation and MATLAB object-oriented programming. *SIAM Review 52*, 3 (2010), 545–563, doi: 10.1137/080743627. A

[75] NILSEN, H., KROGSTAD, S., ANDERSEN, ., ALLEN, R., AND LIE, K.-A. Using sensitivities and vertical-equilibrium models for parameter estimation of CO_2 injection models with application to the sleipner injection to measured data. *Submitted to Energy Procedia* (2016). 3.4, 3.4.5, 5

[76] NILSEN, H. M., HERRERA, P. A., ASHRAF, M., LIGAARDEN, I., IDING, M., HERMANRUD, C., LIE, K.-A., NORDBOTTEN, J. M., DAHLE, H. K., AND KEILEGAVLEN, E. Field-case simulation of CO_2-plume migration using vertical-equilibrium models. *Energy Procedia 4*, 0 (2011), 3801–3808, doi: 10.1016/j.egypro.2011.02.315. 3.1

[77] NILSEN, H. M., SYVERSVEEN, A. R., LIE, K.-A., TVERANGER, J., AND NORDBOTTEN, J. M. Impact of top-surface morphology on CO_2 storage capacity. *Int. J. Greenh. Gas Control 11*, 0 (2012), 221–235, doi: 10.1016/j.ijggc.2012.08.012. 3.2.1

[78] NOCEDAL, J., AND WRIGHT, S. J. *Numerical Optimization*, 2nd ed. Springer, New York, 2006. 3.4.3

[79] NORDBOTTEN, J., CELIA, M., AND BACHU, S. Injection and storage of CO2 in deep saline aquifers: Analytical solution for CO2 plume evolution during injection. *Transp. Porous Media 58*, 3 (2005), 339–360, doi: 10.1007/s11242-004-0670-9. 3.1

[80] NORDBOTTEN, J. M., AND CELIA, M. A. *Geological Storage of CO_2: Modeling Approaches for Large-Scale Simulation*. John Wiley & Sons, Hoboken, New Jersey, 2012. 3.1, 3.1

[81] NORDBOTTEN, J. M., AND DAHLE, H. K. Impact of the capillary fringe in vertically integrated models for CO_2 storage. *Water Resources Research 47*, 2 (2011), doi: 10.1029/2009WR008958. 1.2, 3.1, 3.1

[82] NORDBOTTEN, J. M., FLEMISCH, B., GASDA, S. E., NILSEN, H. M., FAN, Y., PICKUP, G. E., WIESE, B., CELIA, M. A., DAHLE, H. K., EIGESTAD, G. T., AND PRUESS, K. Uncertainties in practical simulation of CO_2 storage. *Int. J. Greenh. Gas Control 9*, 0 (2012), 234–242, doi: 10.1016/j.ijggc.2012.03.007. 1.1, 3.1

[83] OLIVELLA, S., GENS, A., CARRERA, J., AND ALONSO, E. Numerical formulation for a simulator (code_bright) for the coupled analysis of saline media. *Engineering Computations 13*, 7 (1996), 87–112. 1.1

[84] ONSAGER, L. Reciprocal relations in irreversible processes. i. *Phys. Rev. 37* (Feb 1931), 405–426, doi: 10.1103/PhysRev.37.405. 2.1.1

[85] PEACEMAN, D. W. *Fundamentals of Numerical Reservoir Simulation*. Elsevier Science Inc., New York, NY, USA, 1991. 2.2.4

[86] PEROZZI, L., GLOAGUEN, E., GIROUX, B., AND HOLLIGER, K. A stochastic inversion workflow for monitoring the distribution of CO_2 injected into deep saline aquifers. *Computational Geosciences 20*, 6 (2016), 1287–1300, doi: 10.1007/s10596-016-9590-3. 5

[87] PLESSIX, R.-E. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International 167*, 2 (2006), 495–503, doi: 10.1111/j.1365-246X.2006.02978.x. 3.4.3, 3.4.3

[88] PREISIG, M., AND PRÉVOST, J. H. Coupled multi-phase thermo-poromechanical effects. case study: CO_2 injection at In Salah, Algeria. *International Journal of Greenhouse Gas Control 5*, 4 (2011), 1055–1064. 2.2.5

[89] PRUESS, K. The tough codes - a family of simulation tools for multiphase flowand transport processes in permeable media. *Vadose Zone J. 3* (2004), 738–746, doi: 10.2136/vzj2004.0738. 1.1, 2.2.4

[90] ROCHELLE, C. A., CZERNICHOWSKI-LAURIOL, I., AND MILODOWSKI, A. E. The impact of chemical reactions on CO2 storage in geological formations: a brief review. *Geological Society of London Special Publications 233* (Jan. 2004), 87–106, doi: 10.1144/GSL.SP.2004.233.01.07. 1.2

[91] RUTQVIST, J. The geomechanics of CO_2 storage in deep sedimentary formations. *Geotechnical and Geological Engineering 30*, 3 (2012), 525–551, doi: 10.1007/s10706-011-9491-0. 2.2.5

[92] RUTQVIST, J., BIRKHOLZER, J., CAPPA, F., AND TSANG, C.-F. Estimating maximum sustainable injection pressure during geological sequestration of CO_2 using coupled fluid flow and geomechanical fault-slip analysis. *Energy Conversion and Management 48*, 6 (2007), 1798–1807. 2.2.5

[93] RUTQVIST, J., VASCO, D. W., AND MYER, L. Coupled reservoir-geomechanical analysis of CO_2 injection and ground deformations at In Salah, Algeria. *International Journal of Greenhouse Gas Control 4*, 2 (2010), 225–230. 2.2.5

[94] SANDVE, G. K., NEKRUTENKO, A., TAYLOR, J., AND HOVIG, E. Ten simple rules for reproducible computational research. *PLoS Comput Biol 9*, 10 (10 2013), 1–4, doi: 10.1371/journal.pcbi.1003285. (document)

[95] SATHAYE, K. J., HESSE, M. A., CASSIDY, M., AND STOCKLI, D. F. Constraints on the magnitude and rate of CO_2 dissolution at Bravo Dome natural gas field. *Proceedings of the National Academy of Sciences 111*, 43 (2014), 15332–15337. 3.1

[96] SCHLUMBERGER. *ECLIPSE Technical Description*. Schlumberger, 2010. 1.1, 2.1.1, 2.2.4

[97] SETTARI, ANTONIN; WALTERS, D. Advances in coupled geomechanical and reservoir modeling with applications to reservoir compaction. *SPE J. 6* (2001), doi: 10.2118/74142-PA. 2.2.5, 3.3

[98] SETTARI, ANTONIN; MOURITS, F. A coupled reservoir and geomechanical simulation system. *SPE J. 3* (1998), doi: 10.2118/50939-PA. 2.2.5, 3.3

[99] SINGH, V., CAVANAGH, A., HANSEN, H., NAZARIAN, B., IDING, M., AND RINGROSE, P. Reservoir modeling of CO_2 plume behavior calibrated against monitoring data from Sleipner, Norway. In *SPE Annual Technical Conference and Exhibition, 19-22 September 2010, Florence, Italy* (2010). SPE 134891-MS. 3.2.1

[100] SINTEF ICT. The MATLAB Reservoir Simulation Toolbox: Numerical CO_2 laboratory, 12 2015. http://www.sintef.no/co2lab. 1.3

[101] SPAN, R., AND WAGNER, W. A new equation of state for carbon dioxide covering the fluid region from triple-point temperature to 1100 K at pressures up to 800 MPa. *J. Phys. Chem. Ref. Data 25*, 6 (1996), 1509–1597. 2.1.3

[102] SUNDAL, A., HELLEVANG, H., MIRI, R., DYPVIK, H., NYSTUEN, J. P., AND AAGAARD, P. Variations in mineralization potential for CO_2 related to sedimentary facies and burial depth – a comparative study from the North Sea. *Energy Procedia 63* (2014), 5063 – 5070, doi: 10.1016/j.egypro.2014.11.536. 1.2

[103] SUNDAL, A., MIRI, R., RAVN, T., AND AAGAARD, P. Modelling CO_2 migration in aquifers; considering 3D seismic property data and the effect of site-typical depositional heterogeneities. *International Journal of Greenhouse Gas Control 39* (2015), 349 – 365, doi: 10.1016/j.ijggc.2015.05.021. 1.2

[104] SYLTA, Ø. Modelling of secondary migration and entrapment of a multicomponent hydrocarbon mixture using equation of state and ray-tracing modelling techniques. *Geological Society, London, Special Publications 59*, 1 (1991), 111–122, doi: 10.1144/gsl.sp.1991.059.01.07. 3.2.1

[105] SYLTA, Ø. *Hydrocarbon migration modelling and exploration risk*. PhD thesis, Norwegian University of Science and Technology, Faculty of Engineering Science and Technology, Department of Geology and Mineral Resources Engineering, 2004. 3.2.1

[106] TERWEL, B. W., TER MORS, E., AND DAAMEN, D. D. It's not only about safety: Beliefs and attitudes of 811 local residents regarding a CCS project in Barendrecht. *International Journal of Greenhouse Gas Control 9* (2012), 41 – 51, doi: 10.1016/j.ijggc.2012.02.017. 1.1

[107] VAN DER MEER, L. Investigations regarding the storage of carbon dioxide in aquifers in the Netherlands. *Energy Conversion and Management 33*, 5–8 (1992), 611 – 618, doi: 10.1016/0196-8904(92)90063-3. Proceedings of the First International Conference on Carbon Doxide Removal. 1.1

[108] VERMA, A. An introduction to automatic differentiation. *CURRENT SCIENCE-BANGALORE- 78*, 7 (2000), 804–807. A

[109] VILARRASA, V., SILVA, O., CARRERA, J., AND OLIVELLA, S. Liquid CO_2 injection for geological storage in deep saline aquifers. *International Journal of Greenhouse Gas Control 14* (2013), 84 – 96, doi: http://dx.doi.org/10.1016/j.ijggc.2013.01.015. 1.2, 3.1

[110] WANG, H. F. Theory of linear poroelasticity. *Princeton Series in Geophysics, Princeton University Press, Princeton, NJ* (2000). 2.2.5, 2.2.5, 3.3.1, B

[111] WHEELER, J., WHEELER, M., ET AL. Integrated parallel and accurate reservoir simulator. Tech. rep., TICAM01-25, CSM, University of Texas at Austin, 2001. 1.1

[112] WILKINSON, M., HASZELDINE, R. S., FALLICK, A. E., ODLING, N., STOKER, S. J., AND GATLIFF, R. W. CO_2–mineral reaction in a natural analogue for CO_2 storage—implications for modeling. *Journal of Sedimentary Research 79*, 7 (2009), 486–494, doi: 10.2110/jsr.2009.052. 1.2

[113] YAMAMOTO, H., NAKAJIMA, K., ZHANG, K., AND NANAI, S. Performance of parallel simulators on peta-scale platforms for coupled multi-physics modelling of CO_2 geologic sequestration. *Energy Procedia 63* (2014), 3795 – 3804, doi: 10.1016/j.egypro.2014.11.408. 1.2

[114] YORTSOS, Y. A theoretical analysis of vertical flow equilibrium. *Transp. Porous Media 18*, 2 (1995), 107–129, doi: 10.1007/BF01064674. 3.1

[115] ZHOU, Y., TCHELEPI, H. A., AND MALLISON, B. T. Automatic differentiation framework for compositional simulation on unstructured grids with multi-point discretization schemes. In *SPE Reservoir Simulation Symposium, 21-23 February, The Woodlands, Texas* (2011). SPE 141592-MS. A

Part II

Included papers

Paper I

Vertically Averaged Equations with Variable Density for CO_2 Flow in Porous Media

Odd A. Andersen, Sarah E. Gasda, Halvor M. Nilsen
Transport in Porous Media, **107**, (2015)

Vertically Averaged Equations with Variable Density for CO_2 Flow in Porous Media

O. Andersen · S. E. Gasda · H. M. Nilsen

Received: 20 December 2013 / Accepted: 26 November 2014 / Published online: 6 December 2014

Abstract Carbon capture and storage has been proposed as a viable option to reduce CO_2 emissions. Geological storage of CO_2 where the gas is injected into geological formations for practically indefinite storage, is an integral part of this strategy. Mathematical models and numerical simulations are important tools to better understand the processes taking place underground during and after injection. Due to the very large spatial and temporal scales involved, commercial 3D-based simulators for the petroleum industry quickly become impractical for answering questions related to the long-term fate of injected CO_2. There is an interest in developing simplified modeling tools that are effective for this type of problem. One approach investigated in recent years is the use of upscaled models based on the assumption of vertical equilibrium (VE). Under this assumption, the simulation problem is essentially reduced from 3D to 2D, allowing much larger models to be considered at the same computational cost. So far, most work on VE models for CO_2 storage has either assumed incompressible CO_2 or only permitted lateral variations in CO_2 density (semi-compressible). In the present work, we propose a way to fully include variable CO_2 density within the VE framework, making it possible to also model vertical density changes. We derive the fine-scale and upscaled equations involved and investigate the resulting effects. In addition, we compare incompressible, semi-compressible, and fully compressible CO_2 flow for some model scenarios, using an in-house, fully-implicit numerical code based on automatic differentiation, implemented using the MATLAB reservoir simulation toolkit.

Keywords Vertical equilibrium · CO_2 sequestration · CO_2 properties · Compressibility · CCS

O. Andersen (✉)
Department of Mathematics, University of Bergen, Bergen, Norway
e-mail: odd.andersen@sintef.no

O. Andersen · H. M. Nilsen
Department of Applied Mathematics, SINTEF ICT, PO Box 124, Blindern, Oslo 0314, Norway

S. E. Gasda
Uni Research CIPR, PO Box 7810, 5020 Bergen, Norway

1 Introduction

Carbon capture and storage (CCS) is considered an attractive emission reduction strategy as it is compatible with current energy infrastructure and much of the required experience already exists. The potential for underground storage (primarily in saline aquifers or depleted oil and gas fields) is significant, and IPCC estimates that CCS could contribute 15–55 % of cumulative mitigation efforts worldwide until 2100 (according to most scenarios) (IPCC 2005). It is hoped that CCS as a technology could provide a temporary bridge between a fossil fuel-based economy and a future economy based mainly on renewable energy. Mitigation of risk is one focus of current research, with leakage being the main concern. Current experience and models suggest that appropriately selected and managed reservoirs will retain a large percentage of injected CO_2 (IPCC 2005). However, the injected volumes in a meaningful emissions reduction scenario would be on an unprecedented scale.

Numerical simulators provide a key tool for better understanding the properties of potential storage sites, assessing sensitivities to unknown parameters, interpreting observed evolution over time, and providing long-term projections of the ultimate fate of injected CO_2. Present-day commercial simulators (CMG 2009; Schlumberger 2010) developed for the oil and gas industry can be adapted to model CO_2 geological storage scenarios, but require prohibitively long computational times when applied to many problems relevant to long-term storage of CO_2, due to the large spatial and temporal scales involved (Nordbotten and Celia 2012). Simulators developed by universities and institutes, such as CODE_BRIGHT (Olivella et al. 1996), DUMUX (Flemisch et al. 2007), TOUGH2 (Pruess 2004), IPARS (Wheeler et al. 2001), and others, also face this problem. This has led to recent efforts in developing fast new tools designed to address the very large scales associated with CO_2 storage modeling, while respecting the relevant physical and chemical aspects of the system.

As part of this ongoing effort, models based on the assumption of *vertical equilibrium* (VE) have regained attention in the context of CO_2 storage (Nordbotten and Celia 2012). Under this assumption, the injected CO_2 and resident brine are considered to separate quick enough into distinct layers that the process can be considered instantaneous at the time scales relevant for the study of lateral migration. Mathematical models based on this assumption have a long history and were initially investigated for purposes such as oil extraction and water management (Coats et al. 1971; Dietz 1953). The assumption of vertical equilibrium is often applicable given the extreme aspect ratio of a typical reservoir, where the horizontal extent is measured in tens or even hundreds of kilometers whereas the vertical extent is typically no more than a couple of hundred meters. A formal analysis carried out in (Yortsos 1995), expresses the validity of the VE assumption in terms of the geometrical aspect ratio and ratio between horizontal and vertical permeability. The assumption of vertical equilibrium effectively reduces the model from three dimensions (3D) to two dimensions (2D). Together with selective inclusion of physical effects (e.g., hysteresis, capillary fringe, rock heterogeneity), this represents a significant reduction in computational requirements for similar complexity solved with a 3D code.

Multiple comparison studies have shown that results provided by VE modeling compare well with those obtained from full 3D simulations in many scenarios (Class et al. 2009; Ligaarden and Nilsen 2010; Nilsen et al. 2011; Nordbotten et al. 2012). In recent years, VE models for CO_2 storage have been a subject of active research. For simplified cases, analytic solutions have been proposed (Dentz and Tartakovsky 2009; Nordbotten et al. 2005). Numerical solutions can be obtained for more general cases, allowing for the inclusion of, e.g., reservoir heterogeneity in the model. The VE-modeling framework has been extended to include several physical phenomena that affect two-phase flow, such as capillarity (Nord-

botten and Dahle 2011), dissolution and convective mixing (Gasda et al. 2011a, b), leakage through caprock (Nordbotten and Celia 2012), and caprock rugosity (Gasda et al. 2013, 2012). In situations with significant vertical flow, such as in the vicinity of an active or leaky well, the assumption of vertical equilibrium is no longer valid and VE models cannot be applied by themselves. However, by combining local analytic solutions with a global numeric model, the effect of multiple leaky wells has also been handled in a VE setting (Gasda et al. 2009).

Whereas most work on VE models for CO_2 storage so far has considered constant CO_2 density, there has also been some work on the inclusion of compressibility effects at the large scale (Vilarrasa et al. 2010) and the wellbore scale (Mijic et al. 2014). Villarasa et al. proposed a method to include variable CO_2 density (and viscosity) in an analytic VE model (Vilarrasa et al. 2010). Under this approach, the density of injected CO_2 is allowed to vary in time, while remaining constant in space. In (Vilarrasa et al. 2013a), a semi-analytical model is proposed for vertical CO_2 injection, in which CO_2 density varies both in space and time. This model addresses the injection phase on an idealized domain, with several additional simplifying assumptions. The model proposed in (Gasda et al. 2009) includes horizontal variations in density, but does not take into account vertical variability. The inclusion of horizontal variations makes it possible to model the significant changes in CO_2 density arising from the large variations in pressure during CO_2 injection, as well as the gradual changes in the hydrostatic pressure field that occur in a large, sloping reservoir. On the other hand, density differences due to pressure and temperature changes in the vertical direction are not included. Although such changes are assumed to be small in most cases, it is not clear whether they could always be neglected.

In this paper, we develop a mathematically consistent model for variable CO_2 density in VE models. We derive the VE equations for two-phase flow with full compressibility and show how the CO_2 density dependence on depth can be separately factored out as scalar functions in the resulting equations. We discuss how these functions can be practically computed or estimated for the purposes of numerical simulation. Moreover, we investigate the range of pressures and temperature gradients that are likely to occur in CO_2 storage sites, and try to identify conditions for which vertical density changes may have significant impacts. Finally, we compare a fully compressible, a semi-compressible (i.e., horizontally compressible), and an incompressible VE model on some simulated test cases designed to demonstrate typical scenarios for CO_2 storage in which density variation could be important.

2 Model Description

2.1 Physical System

We consider a scenario with CO_2 injected into a saline aquifer. The aquifer is confined on top and below by formations of very low permeability. The confining formation on top is called the caprock. While the caprock can have significant local topographical variation, it is still assumed to be reasonably flat on the large scale. The lower confining formation is simply referred to as the bottom.

The pore space of the aquifer is initially filled with brine. This brine is gradually displaced by CO_2 during and after injection. Although pressures and temperatures in the model may vary, the site would normally be chosen for conditions that allows injected CO_2 to remain in a dense phase. In any case, except in very rare circumstances (Bachu 2003), the density of injected CO_2 would be significantly lower than the density of brine, resulting in strong buoyancy forces acting on the injected CO_2 plume. Gravity and viscous forces will drive the

CO_2 plume to spread out and move upward where possible. If the aquifer is sloping, this would lead the plume to slowly migrate uphill, constrained by the local shape of the aquifer. Given enough time, this migration can cover a large range in depth and potentially lead to significant density changes within the plume.

2.1.1 VE Assumptions and Upscaling

The aim of a VE model is to represent a 3D aquifer of large horizontal extent on a 2D plane by solving integrated 2D governing equations. The development of the integrated or upscaled model relies on the underlying assumption of vertical equilibrium, i.e., zero flow perpendicular to the aquifer plane, which must hold at the spatial and temporal scales of the system of interest. Vertical equilibrium is established under two related conditions, a large aspect ratio and gravity–capillary equilibrium. Previous work has investigated the validity of these assumptions with regards to the spatial (Yortsos 1995) and temporal (Court et al. 2012) scales appropriate for CO_2 storage. The assumption of vertical equilibrium can be considered a special case of the more general *Dupuit assumption*, which considers a pressure field that leads to no flow across the confining (top, bottom) boundaries (Bear 1988).

The aspect ratio of the system is related to the horizontal versus vertical extent of the aquifer of interest. Whereas the horizontal extent of the aquifer can be very large (tens to hundreds of kilometers), the thickness is usually limited to a couple of hundred meters or less. We can therefore think of its global shape as a thin and flat sheet, which may be horizontal or inclined. Since the lateral extent of aquifers considered for CO_2 storage is very large compared to the thickness (Nordbotten and Celia 2012), we expect any vertical movement to be negligibly small compared to the lateral flow velocities.

The gravity segregation process also should occur on very short time scales compared to the characteristic time of the analysis in order for vertical equilibrium to hold. Supercritical CO_2 and brine will tend to separate quickly due to buoyancy, with the CO_2 rising upward (van der Meer 1993). Capillary forces will also act on the plume, creating a two-phase transition zone at equilibrium, known as a capillary fringe. If gravity forces are balanced with capillary forces, then the CO_2 phase can be represented as a plume collected at the top of the aquifer, held in place by the caprock (Nordbotten and Celia 2012). In the absence of strong capillarity, an interface can then be defined between the part of the aquifer occupied by brine and the part occupied by CO_2 referred to as the brine–CO_2 interface.

Given the validity of the above assumptions, the vertical dimension can be eliminated by integration and all upscaled variables either represent (weighted) averages across the vertical direction (densities, porosities, saturation), or values at some reference depth (pressure). The resulting, upscaled model is called a VE model.

2.1.2 Other Simplifying Assumptions

The VE framework is capable of modeling various physical effects such as mutual solubility, capillary fringe, hysteretic behavior, and diffuse leakage, among others. However, it should be noted that our primary goal in this paper is to develop a VE formulation that includes variable CO_2 density in a manner fully consistent with the fine scale and to compare the resulting model with simpler approaches. Thus, we wish to isolate the effect of CO_2 density separate from other mechanisms, which facilitates presentation and analysis of the model. In doing so, we disregard a number of physical effects that are nevertheless important for a complete understanding of CO_2 migration. The following are the main simplifications: the

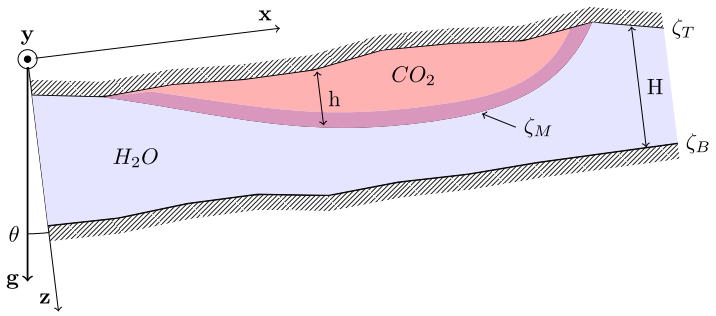


Fig. 1 A vertical section of a part of an aquifer containing a plume of injected CO_2. ζ_{T} and ζ_{B} represent the shape of the *top*- and *bottom*-confining layers, while ζ_{M} describes the shape of the brine–CO_2 interface. H and h denote the local height of the aquifer and the CO_2 plume, while θ is the angle between the z coordinate axis and the gravity vector $\mathbf{g}$

top and bottom boundaries of the aquifer are impermeable; the rock matrix and brine phase are incompressible; negligible capillary fringe, viscosity of both fluids is constant; no mutual solubility; zero residual saturations; and thermal equilibrium between fluids and rock matrix at all times.

Notably, CO_2 viscosity is kept constant in the model development and all test examples, even though we are well aware that viscosity changes with temperature and pressure are as important as density changes in real systems. The main purpose of ignoring viscosity variations is to focus on handling of vertical variation of CO_2 density when upscaling the 3D compressible model, a process that has some important implications for the vertical pressure profile and subsequent integration since pressure affects density and vice versa (as discussed in the model derivation in Sect. 2.2). This type of coupling with pressure does not occur with viscosity variation, which is therefore of less interest mathematically. To be consistent, we ignore viscosity variation throughout the subsequent test examples, despite the fact that this may lead to unrealistic combinations of fluid properties.

We emphasize that the simplifications made here can be easily included, along with variable CO_2 density and viscosity, in a more general VE model. Some such models have recently been implemented using the CO_2 module of the open-source MATLAB Reservoir Simulation Toolbox (Lie et al. 2012; MRST 2014; SINTEF ICT 2014), as demonstrated in several recent publications (Andersen et al. 2014; Nilsen et al. 2014a, b, c).

2.1.3 Coordinate System

We define a coordinate system associated with the aquifer, where the x and y coordinate axes lie in the aquifer plane aligned with the principal flow direction, and the z axis is perpendicular to it, directed from the top (caprock) toward the bottom. The shapes of the caprock and bottom surfaces are described, respectively, by the functions ζ_{T} and ζ_{B}, which each associate a depth value (z) to each pair of (x, y) values. Likewise, the shape of the brine–CO_2 interface is described by the function ζ_{M}. Since we allow the aquifer to be sloped, the axes of our coordinate system form an angle, θ, with the true horizontal and vertical directions (c.f. Fig. 1). The unity vectors along the axes are written $\mathbf{e}_x$, $\mathbf{e}_y$, and $\mathbf{e}_z$. We write the coordinate tuple $\mathbf{x} = (x, y, z)$, and its first two components $\mathbf{x}_{||} = (x, y)$. Moreover, we define the unity vector $\mathbf{e}_{||}$, which lies in the (x, y) plane and indicates the normalized projection of the true vertical direction onto this plane. As such, the gravity vector, $\mathbf{g}$, and

the thermal gradient vector, $\mathbf{G}$, both of which are directed true vertically downward, can be written in our coordinate system as follows:

$$\begin{aligned} \mathbf{g} &= g_{||}\,\mathbf{e}_{||} + g_{\perp}\,\mathbf{e}_z \\ \mathbf{G} &= G_{||}\,\mathbf{e}_{||} + G_{\perp}\,\mathbf{e}_z \end{aligned} \tag{1}$$

where $g_{||} = ||\mathbf{g}||\sin\theta$, $g_{\perp} = ||\mathbf{g}||\cos\theta$, and similar for $G_{||}$ and $G_{\perp}$ with respect to $\mathbf{G}$. From now on, we will use the word *vertical* to refer to the direction specified by $\mathbf{e}_z$, and *lateral* for directions in the (x, y) plane. As the temperature gradient G may have both a vertical and lateral component in our inclined coordinate system, temperature at a given point will depend both on depth z and lateral position (x, y). Given a reference temperature T_0 at some reference point $\mathbf{x}_0 = (x_0, y_0, z_0)$, the expression for temperature T at any other point $\mathbf{x}$ becomes

$$\begin{aligned} T(\mathbf{x}) &= T_0 + \mathbf{G}\cdot(\mathbf{x} - \mathbf{x}_0) \\ &= T_0 + G_{||}\mathbf{e}_{||}\cdot(\mathbf{x}_{||} - \mathbf{x}_{||0}) + G_{\perp}(z - z_0). \end{aligned} \tag{2}$$

2.2 Derivation of the Full Variable Density Model

In this section, we derive the equations for the vertically averaged model of two-phase flow in porous media, extended to allow for temporal and spatial variations in density. We start by presenting the derivation and general form of the upscaled equations with variable density. Using the general form as a basis, we proceed in the following section by showing what specific forms these equations take when adding the assumptions of a sharp interface and vertically homogeneous rock properties.

For multiphase immiscible flow, the mass balance equation for phase α (where α denotes either CO_2 or brine) on the fine scale can be written (Nordbotten and Celia 2012):

$$\frac{\partial(\rho_\alpha \phi s_\alpha)}{\partial t} + \nabla(\rho_\alpha \mathbf{u}_\alpha) = \psi_\alpha. \tag{3}$$

In this equation, ρ represents fluid density (ML^{-3}), ϕ the porosity of the medium (–), s the saturation (–), ψ the mass source term ($\mathrm{ML}^{-3}\mathrm{T}^{-1}$), and $\mathbf{u}$ the volumetric flux vector (LT^{-1}). We relate $\mathbf{u}_\alpha$ to phase pressure using Darcy's law for multiphase flow:

$$\mathbf{u}_\alpha = -\lambda_\alpha \mathbf{k}\,(\nabla p_\alpha - \rho_\alpha \mathbf{g}) \tag{4}$$

Here, $\mathbf{k}$ is the permeability tensor (L^2), p_α the phase pressure ($\mathrm{ML}^{-1}\mathrm{T}^{-2}$), $\mathbf{g}$ the gravity acceleration vector (LT^{-2}), and λ_α the phase mobility ($\mathrm{M}^{-1}\mathrm{LT}$) defined as the relative permeability of the phase divided by phase viscosity: $\lambda_\alpha := k_{r,\alpha}(s_\alpha)/\mu_\alpha$.

By integrating the fine-scale equation (3) across the thickness of the aquifer with respect to z, we obtain an upscaled equation whose variables only depend on two spatial variables (x, y) in addition to time t. The equation can be expressed on the following form:

$$H\Phi\frac{\partial}{\partial t}(R_\alpha S_\alpha) + \nabla_{||}\mathbf{F}_\alpha = \Psi_\alpha \tag{5}$$

The upscaled variables in this equation are H (aquifer thickness, as a function of x and y), S (saturation), $\mathbf{F}_\alpha$ (mass flux), Φ (porosity), and R_α (density). $\nabla_{||}$ represents the 2D gradient operator in x and y. While we assume impermeable top and boundaries, the general case would see additional source and sink terms appearing as a result of the integration, corresponding to mass flow across boundaries. Details on the integration have been published in previous literature, and a thorough explanation can be found in (Nordbotten and Celia 2012).

With H given, the formal definitions of the upscaled variables follow from the upscaled terms of (3):

$$H\Phi := \int_{\zeta_T}^{\zeta_B} \phi \, \mathrm{d}z \tag{6}$$

$$\mathbf{F}_\alpha := \int_{\zeta_T}^{\zeta_B} \rho_\alpha \mathbf{u}_{||\alpha} \, \mathrm{d}z \tag{7}$$

$$H\Phi S_\alpha := \int_{\zeta_T}^{\zeta_B} \phi s_\alpha \, \mathrm{d}z \tag{8}$$

$$\Psi_\alpha := \int_{\zeta_T}^{\zeta_B} \psi_\alpha \, \mathrm{d}z \tag{9}$$

$$H\Phi S_\alpha R_\alpha := \int_{\zeta_T}^{\zeta_B} \rho_\alpha \phi s_\alpha \, \mathrm{d}z \tag{10}$$

The variable $\mathbf{u}_{||\alpha}$ in the definition of $\mathbf{F}_\alpha$ above refers to the lateral components of the fine-scale volumetric flux vector $\mathbf{u}_\alpha$. Using Leibniz' rule, it can be shown that for impermeable top and bottom boundaries

$$\int_{\zeta_T}^{\zeta_B} \nabla(\rho_\alpha \mathbf{u}_\alpha) \, \mathrm{d}z = \nabla_{||} \int_{\zeta_T}^{\zeta_B} \rho_\alpha \mathbf{u}_{||\alpha} \, \mathrm{d}z$$

which justifies the use of $\mathbf{u}_{||\alpha}$ in the equations above (Nordbotten and Celia 2012). By assuming lateral and vertical components of tensor $\mathbf{k}$ to be independent, $\mathbf{u}_{||}$ only depends on the lateral component of the pressure gradient $\nabla_{||} p$. We can combine (7) and (4) to obtain an expression of upscaled flux in terms of pressure:

$$\mathbf{F}_\alpha = -\int_{\zeta_\mathrm{T}}^{\zeta_\mathrm{B}} \rho_\alpha \mathbf{k}_{||} \lambda_\alpha (\nabla_{||} p_\alpha - \rho_\alpha g_{||} \mathbf{e}_{||}) \, \mathrm{d}z \tag{11}$$

In order to compute integrals (6)–(11), we need knowledge of the involved fine-scale quantities. Some of these (ϕ, $\mathbf{k}_{||}$ and ψ_α) are considered known inputs, whereas the others (s_α, ρ_α, λ_α and $\nabla_{||} p_\alpha$) will be reconstructed from reference quantities based on specific assumptions. Several models exist for the reconstruction of fine-scale saturations s_α and mobilities λ_α based on the upscaled saturation S_α. The appropriate model to choose depends on the approximations valid for a given scenario. For the present paper we discuss a sharp-interface model, which is valid when capillary pressure effects are negligible in the 3D model. As for ρ_α and $\nabla_{||} p$, we must establish how these can be represented in terms of reference values of density and pressure, with the dependence on z separately identified.

The assumptions of hydrostatic pressure (a consequence of VE) and constant geothermal gradient make it possible to express fine-scale pressure and density directly as functions of depth. This allows us to relate the integrals above with pressure and density values at some chosen reference surface ζ_R. To simplify notation, we will omit the subscript α from the presented equations from this point forward, keeping in mind that the form is the same for both phases.

We choose to express fine-scale pressure in terms of its value at the reference surface $\zeta_R = \zeta_R(\mathbf{x}_{||})$. The vertical (depth) distance from this surface to a point $\mathbf{x} = (x, y, z)$ is thus $z - \zeta_R(x, y)$. We then have the following expression for pressure as a function of z:

$$p(z) = p_R + g_\perp \int_{\zeta_R}^{z} \rho(z')\,\mathrm{d}z'. \tag{12}$$

Here, p_R represents the phase pressure at the corresponding point on the reference surface ζ_R for a given vertical column. For the numerical examples in this paper, we chose the CO_2–water interface as reference, i.e., $\zeta_R = \zeta_M$. This choice has an advantage when using the semi-compressible VE model, as discussed in Sect. 2.4.1. Other choices of reference surface are, however, also valid, with a similar degree of accuracy.

From (12), we see that the vertical pressure profile depends on the corresponding vertical density profile. Since density is a function of pressure and temperature, this profile can be obtained from the corresponding equation of state by function composition:

$$\rho(z) = \rho(p(z), T(z)). \tag{13}$$

We here encounter a complication that arise when including variable vertical density in the VE framework: vertical pressure and density profiles depend on each other. In order to proceed, we compute the derivative of $\rho(z)$

$$\frac{\mathrm{d}}{\mathrm{d}z}\rho(z) = \frac{\partial\rho}{\partial p}\frac{\partial p}{\partial z} + \frac{\partial\rho}{\partial T}\frac{\partial T}{\partial z}. \tag{14}$$

From (12), we see that $\frac{\partial}{\partial z}p = g_\perp\rho$, and from (2), that $\frac{\partial}{\partial z}T = G_\perp$. The other partial derivatives are given by the following functions of state:

$$\beta = \frac{1}{\rho}\frac{\partial\rho}{\partial p} \qquad \gamma = -\frac{1}{\rho}\frac{\partial\rho}{\partial T}. \tag{15}$$

β is called the isothermal compressibility coefficient for the given substance, whereas γ is called the isobaric coefficient of thermal expansion. (Note the negative sign in front of γ). Using the above, we can now further develop expression (14) into

$$\frac{\mathrm{d}}{\mathrm{d}z}\rho(z) = \rho(\beta g_\perp\rho - \gamma G_\perp) \tag{16}$$

We now have ρ expressed in the form of an ordinary differential equation (ODE) that can be solved numerically. Note that both β and γ are continuous functions of temperature and pressure as long as one does not cross the vapor–liquid boundary.

Having established an equation to obtain $\rho(z)$ in terms of reference values, we now turn our attention to $\nabla_{||}p$. As we see from (11), the computation of upscaled mass flux requires knowledge of the fine-scale lateral pressure gradient $\nabla_{||}p$. In the incompressible setting, this quantity is independent of z and can simply be moved outside the integral. In the compressible setting, however, the lateral pressure gradient varies with depth and has

to remain in the integrand. In order to compute the integral, we therefore seek to obtain an analytical expression of the fine-scale gradient in terms of reference quantities and depth. This approach facilitates analysis and prevents the need of explicitly reconstructing fine-scale pressure at each time-step in order to compute the gradient numerically.

By combining (12) with (2), and writing out the result explicitly in terms of all three spatial coordinates $(x, y, z) = (\mathbf{x}_{||}, z)$, we get

$$p(\mathbf{x}_{||}, z) = p_R(\mathbf{x}_{||}) \tag{17}$$

$$+g_{\perp} \int_{\zeta_R(\mathbf{x}_{||})}^{z} \rho\left(p(\mathbf{x}_{||}, z'), T_0 + G_{||}\mathbf{e}_{||} \cdot (\mathbf{x}_{||} - \mathbf{x}_{||0}) + G_{\perp}(z' - z_0)\right) \mathrm{d}z'. \tag{18}$$

We note that the dependence of p on the spatial coordinates can be fully expressed in terms of T and the reference quantities p_R and ζ_R. Using the chain rule, $\nabla_{||} p$ can therefore be expressed solely in terms of the lateral gradients of these quantities. By noting that $\nabla_{||} T = \nabla_{||}(G_{||}\mathbf{e}_{||} \cdot \mathbf{x}_{||}) = G_{||}\mathbf{e}_{||}$, we can write

$$\nabla_{||} p = \nu_p \nabla_{||} p_R + \nu_\zeta \nabla_{||} \zeta_R + \nu_G G_{||}\mathbf{e}_{||} \tag{19}$$

with

$$\nu_p = \frac{\partial p}{\partial p_R} \tag{20}$$

$$\nu_\zeta = \frac{\partial p}{\partial \zeta_R} \tag{21}$$

$$\nu_G = \frac{\partial p}{\partial T}. \tag{22}$$

For each column, the coefficients ν_p, ν_ζ, and ν_G are functions of z only, described by ODEs. We can determine these ODEs by partial differentiation of hydrostatic pressure with respect to p, ζ_R, and T, respectively. In Appendix 1 we provide details on the calculation and further show that $\nu_\zeta(z) = -g_{\perp}\rho_R \nu_p(z)$, and $\nu_G = \frac{g}{G}(\rho - \rho_R \nu_p)$, allowing us to eliminate these functions and express the z-dependence of $\nabla_{||} p$ using only the two ODEs for ν_p and ρ:

$$\nabla_{||} p = \nu_p \left[\nabla_{||} p_R - \rho_R (g_{\perp} \nabla_{||} \zeta_R + g_{||}\mathbf{e}_{||})\right] + \rho g_{||}\mathbf{e}_{||} \tag{23}$$

We show the ODE defining ν_p to be

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}z}\nu_p = (g_{\perp}\rho\beta)\nu_p \\ \nu_p(\zeta_R) = 1. \end{cases} \tag{24}$$

We now have all we need to compute the upscaled flux. Combining (11) and (23), reorganizing terms and moving the z-independent part outside the integral, we get

$$\mathbf{F} = -\left(\int_{\zeta_T}^{\zeta_B} \nu_p \rho \lambda \mathbf{k}_{||} \, \mathrm{d}z\right) \left(\nabla_{||} p_R - \rho_R (g_{\perp} \nabla_{||} \zeta_R + g_{||}\mathbf{e}_{||})\right)$$

$$= -H\mathbf{N}\tilde{\mathbf{R}}\boldsymbol{\Lambda}\mathbf{K}\left(\nabla_{||} p_R - \rho_R (g_{\perp} \nabla_{||} \zeta_R + g_{||}\mathbf{e}_{||})\right) \tag{25}$$

with the following definitions of upscaled variables:

$$H\mathbf{K} := \int_{\zeta_T}^{\zeta_B} \mathbf{k}_{||}\, \mathrm{d}z \tag{26}$$

$$H\boldsymbol{\Lambda}\mathbf{K} := \int_{\zeta_T}^{\zeta_B} \lambda \mathbf{k}_{||}\, \mathrm{d}z \tag{27}$$

$$H\tilde{\mathbf{R}}\boldsymbol{\Lambda}\mathbf{K} := \int_{\zeta_T}^{\zeta_B} \rho\lambda \mathbf{k}_{||}\, \mathrm{d}z \tag{28}$$

$$H\mathbf{N}\tilde{\mathbf{R}}\boldsymbol{\Lambda}\mathbf{K} := \int_{\zeta_T}^{\zeta_B} \nu_p \rho\lambda \mathbf{k}_{||}\, \mathrm{d}z \tag{29}$$

Previously in (10), we defined R as a vertical average of the fine-scale density, weighted by porosity. In (28) above, a different density-related quantity $\tilde{\mathbf{R}}$ is defined, which represents the vertical average of density, weighted by *permeability*. As such, $\tilde{\mathbf{R}}$ is a tensor value, and not easily interpretable as a quantity separate from the context in which it figures—as an inherent component of the upscaled mass flux. For this reason, we choose to interpret the upscaled mass flux as a basic quantity in itself and will not formally factor it into a density and a volumetric flux part. (Moreover, the upscaled volumetric flux from such a factorization will not generally be equal to the upscaled volumetric flux directly obtained from vertical integration).

2.2.1 Complete Upscaled Equation Set

Using previous definitions, the final set of upscaled equations for each phase becomes

$$H\Phi \frac{\partial}{\partial t}(RS) + \nabla_{||} \cdot \mathbf{F} = \Psi \tag{30}$$

$$\mathbf{F} = -H\mathbf{N}\tilde{\mathbf{R}}\boldsymbol{\Lambda}\mathbf{K}\left(\nabla_{||} p_R - \rho_R (g_\perp \nabla_{||}\zeta_R + g_{||}\mathbf{e}_{||})\right). \tag{31}$$

The equation system can be solved for upscaled saturation, S_α, and reference pressure, p_R, for each phase. Each phase can be modeled either as compressible or incompressible. For the incompressible phase, the equations reduce to those of incompressible two-phase VE flow, c.f. Sect. 2.4.2.

In our inclined coordinate system, the expression for the gravity potential V can be established in a similar manner as for temperature in (2), namely

$$\begin{aligned} V(\mathbf{x}) &= V_0 - \mathbf{g}\cdot(\mathbf{x}-\mathbf{x}_0) \\ &= V_0 - g_{||}\mathbf{e}_{||}\cdot(\mathbf{x}_{||}-\mathbf{x}_{||0}) - g_\perp(z-z_0) \end{aligned} \tag{32}$$

On the chosen reference surface, we have $z = \zeta_R(\mathbf{x}_{||})$, so the gravity potential $V_R(\mathbf{x}_{||})$ and its lateral gradient $\nabla_{||}V_R(\mathbf{x}_{||})$ are written

$$V_R(\mathbf{x}_{||}) = V_0 - g_{||}\mathbf{e}_{||}\cdot(\mathbf{x}_{||}-\mathbf{x}_{||0}) - g_\perp(\zeta_R(\mathbf{x}_{||}) - z_0) \tag{33}$$

$$\nabla_{||}V_R(\mathbf{x}_{||}) = -g_{||}\mathbf{e}_{||} - g_\perp \nabla_{||}\zeta_R(\mathbf{x}_{||}). \tag{34}$$

Table 1 Upscaled variables for the homogeneous, sharp-interface system

$\Phi = \phi$	$\Lambda = \left(\frac{h}{H\mu}\right)$
$S = \frac{h}{H}$	$R = \mathscr{J}_1$
$\mathbf{K} = \mathbf{k}_{\|\|}$	$\Psi = H\psi$
$N = \frac{\mathscr{J}_2}{\mathscr{J}_1}$	$\tilde{R} = R$

We recognize $\nabla_{||} V_R$ inside the parenthesis of the upscaled flux expression (31). We further note that in (31), the upscaled flux $\mathbf{F}$ becomes zero when the pressure gradient equals the negative gravity potential gradient, as one would expect.

Although the upscaled quantities $\mathbf{N}$, $\tilde{\mathbf{R}}$, $\boldsymbol{\Lambda}$, and $\mathbf{K}$ can be defined as separate quantities according to Eqs. (26)–(29) above, for practical simulation only their product is needed, and we would not need to compute them separately. Moreover, their interpretation as separate quantities is less natural than for their fine-scale counterparts; they are linked by their dependence on the same fine-scale variables. For instance, changes in fine-scale $\mathbf{k}_{||}$ lead to changes not only in $\mathbf{K}$, but in $\boldsymbol{\Lambda}$, $\tilde{\mathbf{R}}$, and $\mathbf{N}$ as well.

2.3 Homogeneous, Sharp-Interface System

To simplify the definition of several upscaled variables and facilitate further analysis of the equations, we assume vertically homogeneous rock properties and a sharp CO_2–water interface. A sharp interface means that the saturation of phase α is always at endpoint saturation inside its region, and at residual saturation outside. Since we do not take residual saturation into account, the CO_2 saturation therefore equals one inside the plume and zero outside. Note that residual saturations could easily be added if needed. As a consequence, the mobility of phase α will be μ_α^{-1} inside the region it occupies and zero outside. Vertically homogeneous rock properties allow us to move porosity and permeability outside the integrals. The upscaled expressions for CO_2 and for brine are similar, but with integrals taken over different intervals (corresponding to the height of the respective phase domain). While we present the expressions for the CO_2 phase below, the corresponding expressions for the brine phase can be obtained by substituting h with $H - h$ and changing the bounds of the integral to be from ζ_{M} to ζ_{B}.

For the CO_2 phase, the new significantly simpler expressions for the previously defined upscaled variables are listed in Table 1, using the following short-hand for integrals:

$$\mathscr{J}_1 = \frac{1}{h} \int_{\zeta_{\mathrm{T}}}^{\zeta_{\mathrm{M}}} \rho \, \mathrm{d}z \tag{35}$$

$$\mathscr{J}_2 = \frac{1}{h} \int_{\zeta_{\mathrm{T}}}^{\zeta_{\mathrm{M}}} \rho v_p \, \mathrm{d}z. \tag{36}$$

Since the CO_2 mobility is zero outside the zone it occupies, we only need to integrate across the plume height, i.e., from ζ_{T} to ζ_{M}. Note that several previously defined tensorial quantities ($\tilde{\mathbf{R}}$, $\mathbf{N}$, $\boldsymbol{\Lambda}$) can be now be represented as simple scalar values.

The simplified equations for the CO_2 phase can be written out directly in terms of the $\mathscr{J}_i$:

$$\phi \frac{\partial}{\partial t}(h \mathscr{J}_1) + \nabla_{||} \cdot \mathbf{F} = H\psi \tag{37}$$

$$\mathbf{F} = -\frac{h}{\mu}\mathscr{J}_2\mathbf{k}_{||}\left(\nabla_{||}p_R - \rho_R(g_\perp\nabla_{||}\zeta_R + g_{||}\mathbf{e}_{||})\right). \tag{38}$$

By choosing $\zeta_R = \zeta_{\mathrm{M}}$, the equations of the two phases are described using a common reference pressure p_R, and the combined system can be solved in terms of p_R and h.

2.4 Special Cases

2.4.1 Special Case 1: Semi-compressible Model

It can often be reasonable to ignore vertical density changes while still allowing lateral variation. We refer to such a model as semi-compressible. Our equations reduce to a version of this model when we remove the effects of depth from the upscaled variables. For the sharp-interface model above, this means that both integrals $\mathscr{J}_1$ and $\mathscr{J}_2$ reduce to ρ_R, and equation (37) and (38) become

$$\phi\frac{\partial}{\partial t}(\rho_R h) + \nabla_{||}\cdot\mathbf{F} = H\psi \tag{39}$$

$$\mathbf{F} = -\rho_R\frac{h}{\mu}\mathbf{k}_{||}\left(\nabla_{||}p_R - \rho_R(g_\perp\nabla_{||}\zeta_R + g_{||}\mathbf{e}_{||})\right). \tag{40}$$

These equations describe a semi-compressible, sharp-interface system where $\rho(\mathbf{x}) := \rho_R(x, y)$. However, as these equations stand above, they are not entirely consistent with the implied fine-scale flux. In fact, the assumption of constant vertical density inevitably leads to non-physical distortions in the hydrostatic pressure field. To demonstrate this, we consider the reconstruction of the fine-scale hydrostatic pressure field under the assumption of constant vertical density (for simplicity, we consider a zero dip angle):

$$p(z) = p_R + \rho_R g(z - \zeta_R)$$

The lateral gradient at z then becomes

$$\nabla_{||}p(z) = \nabla_{||}p_R - \rho_R g\nabla_{||}\zeta_R + \nabla_{||}\rho_R g(z - \zeta_R). \tag{41}$$

We see that the fine-scale lateral pressure gradient includes a term that depends linearly on z. This vertical variation in the pressure field is non-physical and introduced by our assumption that density is free to vary in two spatial dimensions but not in the third.

We can obtain a new corrected expression for the upscaled mass flux using (41) as the expression of fine-scale pressure gradient and doing the vertical integration. We obtain

$$\mathbf{F}^* = -\rho_R\frac{h}{\mu}\mathbf{k}_{||}\left(\nabla_{||}p_R - \rho_R(g_\perp\nabla_{||}\zeta_R + g_{||}\mathbf{e}_{||}) + C^*\nabla_{||}\rho_R\right), \tag{42}$$

where

$$C^* = \frac{1}{2h}g_\perp((\zeta_M - \zeta_R)^2 - (\zeta_{\mathrm{T}} - \zeta_R)^2). \tag{43}$$

The mass flux expression in (42) is thus equal to the one in (40) with the additional term $C^*\nabla_{||}\rho_R$. This 'corrective' term is necessary to keep the coarse-scale mass-flux consistent with the fine scale one. For certain choices of ζ_R, the expression of C^* becomes simpler. For instance, for $\zeta_R = \zeta_M$, we get $C^* = -\frac{1}{2}hg_\perp$. If on the other hand ζ_R is taken to be $\frac{1}{2}(\zeta_M + \zeta_{\mathrm{T}})$, then C^* becomes zero.

It should also be noted that the term linear in z in (41) means that for situations where there is lateral variation in density, a set of horizontal, parallel isobars at the fine-scale cannot

be obtained. One consequence for VE simulation is that after complete equilibrium has been achieved in a model (system at rest), the CO_2–water interface will not be completely flat unless the reference surface ζ_R itself is. This problem manifests itself whether one uses (40) or (42) to compute upscaled flux, but is more pronounced for (40). Using the CO_2–water interface as reference surface ζ_R shifts the problem away from this interface, which will then be flat for systems in equilibrium.

2.4.2 Special Case 2: Incompressible Model

By considering ρ to be constant in time and space, the equations of (39)–(40) can be divided by density, and the system is reduced to the well-known equations for a sharp-interface, incompressible system, with the right hand side now representing a volumetric source term.

$$\phi \frac{\partial}{\partial t} h + \nabla_{||} U = H\psi \tag{44}$$

$$U = -\frac{h}{\mu} \mathbf{k}_{||} \left(\nabla_{||} p_R - \rho_R (g_\perp \nabla_{||} \zeta_R + g_{||} \mathbf{e}_{||}) \right) \tag{45}$$

2.5 Computation of ρ and ν_p

The computation of the integrals $\mathscr{J}_1$ and $\mathscr{J}_2$ involves evaluation of ρ and ν_p for given values of reference pressure, reference surface position, and depth. These values will generally vary from time-step to time-step in a simulation and have to be computed for each cell of the discretized grid. However, these functions are not defined by explicit formulas, but indirectly as solutions to ODEs.

There are several ways to obtain actual function values for the purposes of running a numerical simulation. The most obvious approach would be to compute the function values directly as needed using an ODE solver. This is however a computationally expensive solution, since these functions need to be evaluated in every grid-point at every time-step. Moreover, as they figure in the integrands of $\mathscr{J}_1$ and $\mathscr{J}_2$, the ODE solver needs to provide the function values across the whole integration interval, not just at the end points. On the other hand, it should be mentioned that the computation of ρ and ν_p by integration of ODEs is trivially parallelizable, allowing for efficient implementation on modern computing hardware.

Another option would be to precompute the integrals for a wide range of pressures, reference surface positions, and depths. The results could be stored in lookup tables from which specific values could be extracted or interpolated as needed during simulation. This approach requires much less computation than solving the relevant ODE directly every time. In order to provide adequate accuracy, the lookup tables would however have to be large, as they need to be sufficiently densely sampled in three independent variables (two, if reference surface is fixed).

A third option is to approximate the functions during runtime using a Taylor expansion centered on the known value of the function (i.e., at the reference surface). This approach only requires knowledge of the function derivatives around ζ_R. These can be computed formally for all three functions, in terms of derivatives of the function of state (β, γ, and higher derivatives of these). A second-order Taylor development of ρ and ν_p is written on the general form:

$$f(z) = f(\zeta_R) + f'(\zeta_R)(z - \zeta_R) + \frac{1}{2} f''(\zeta_R)(z - \zeta_R)^2 + (z - \zeta_R)^2 \epsilon(z - \zeta_R), \tag{46}$$

where $\epsilon \to 0$ as $z \to \zeta_R$. The formal expressions of values and derivatives for ρ and ν_p are provided in Appendix 2. For the sharp-interface, vertically homogeneous model, a convenient

feature of the Taylor expansion approach is that the numerical integration is no longer needed to compute the involved integrals, as the integrals themselves are easily approximated by the Taylor developments.

One limitation of this approach, however, is that it requires the involved functions to have continuous derivatives, which means it cannot be used for approximations that involves crossing the liquid–vapor boundary. Also, in regions close to the critical point, the approximation will only be reasonably accurate in a very limited depth range around ζ_R. This issue will be closely investigated in the following section.

3 Applicability

Vertical changes in CO_2 density within the aquifer are determined by changes in pressure and temperature, both increasing with depth. These changes exert opposite effects on CO_2 density, which may therefore increase, decrease, or remain roughly constant with depth. As seen in the previous section, the full inclusion of variable density adds significant complexity to the VE model. Depending on reservoir conditions and required accuracy, this additional complexity might not be needed. In this section, we examine a wide range of scenarios defined by constant temperature gradients and hydrostatic pressure. In each case, we examine the behavior of the CO_2 density profile and mass flux and determine what vertical correction (in terms of Taylor development) would be appropriate.

3.1 The Depth–Temperature Gradient Diagram

To analyze the CO_2 density changes at different depths and for temperature gradients, we use a 2D diagram whose horizontal axis corresponds to the temperature gradient, and vertical axis to depth. We refer to this diagram as the depth versus temperature gradient diagram (DTG diagram). Assuming hydrostatic pressure and constant water density, the region spanned by such a diagram can be seen as a nonlinear transformation of the pressure/temperature plane commonly used for the CO_2 phase diagram (Fig. 2b). A vertical line in this diagram corresponds to the locus of pressure/temperature values encountered when moving from the surface downward. The vertical axis of the diagram has been inverted to reflect this idea.

We will use this format to examine vertical differences in CO_2 density and mass flux observed for specified CO_2 plume thickness. We construct an example diagram for an offshore aquifer where we assume a constant water density of 1,000 kg/m^3, a sea depth of 100 m, and a sea floor temperature of 4 °C. We let the geothermal gradient vary from 15 K/km to 60 K/km.

Values for CO_2 density are here interpolated from a regularly sampled table based on the equation of state proposed by (Span and Wagner 1996). The table contains 2000 × 2000 values, with pressure ranging from 0.1 to 15 MPa and temperature from 270 to 350 K.

3.2 Vertical Differences in Density

We here examine the vertical variation in density for a CO_2 plume of a given thickness h, trapped under a confining formation, for varying depths and geothermal gradient values. We consider the relative difference between vertically averaged density R (cf. Table 1) and reference density ρ_R, using ζ_M as the reference surface. The relative density difference then has the following expression:

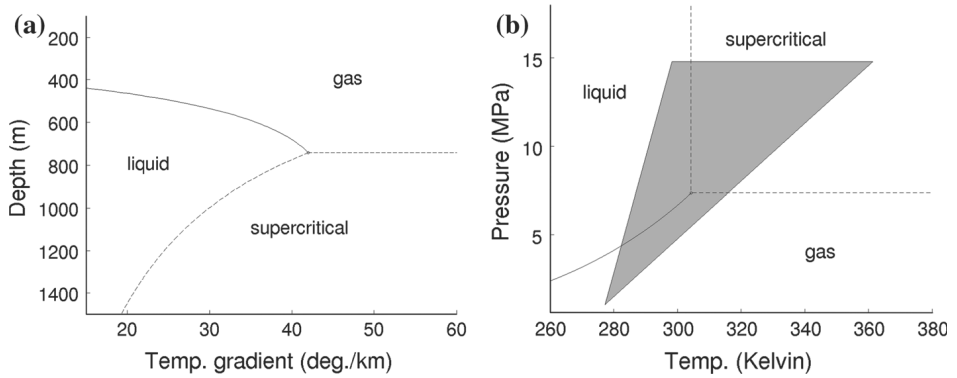


Fig. 2 *Left* A DTG diagram corresponding to a surface temperature of 4 °C at a sea floor depth of 100 m. The CO_2 phase transition lines and critical point are indicated. *Right* relevant part of the usual CO_2 phase diagram, with the domain covered by the DTG diagram indicated by the *gray triangular area*. The lowermost point of the triangle corresponds to the sea floor bottom

$$\frac{R-\rho_R}{\rho_R} = \frac{1}{\rho_R}\mathscr{J}_1(h) - 1. \tag{47}$$

The left column diagrams of Fig. 3a and b present the relative CO_2 density difference as contour plots on the DTG diagram (liquid–vapor boundary and supercritical region have been indicated for reference), for two different plume thicknesses. Positive values indicate that vertically averaged densities are higher than the reference, suggesting decreasing density with depth. The white region around the liquid–vapor boundary line indicates conditions where gas–liquid phase transition is encountered within the height of the plume. With the possibility of partial CO_2 evaporation/condensation, this region requires more care in the numerical reconstruction of the vertical density profile and is thus omitted here.

For the 20-m thick plume in (a), only a small egg-shaped region around the critical point, most of which in the supercritical region, shows a difference of more than 1 % for averaged density. For the 100-m thick plume in (b), the regions where correction is needed have significantly expanded, covering most of the supercritical region and parts of the gas and liquid regions as well.

The plots on the right indicate zones where different orders of Taylor development are needed to approximate the variation in CO_2 density in order to keep relative density error below a given tolerance threshold, which we here have set to 1 %. In the red region around the critical point, not even a second-order correction is sufficient to approximate density within the tolerance. In the white region, where a crossing of the liquid–vapor boundary will occur within the height of the plume, a Taylor approximation would not work at all. To construct these diagrams, the Taylor approximated values have been compared with values for density obtained through numerical integration of the corresponding ODE, using the MATLAB `ode23` solver.

As previous work has pointed out (Bachu 2003), CO_2 density profiles tend to remain constant or slightly increase with depth as long as the temperature gradient is high enough to avoid conditions where the liquid–vapor boundary is crossed moving downward. It might therefore seem surprising that the diagrams in Fig. 3 indicate large areas where CO_2 density at the bottom is lower than at the top. This can be explained by the fact that the corresponding density curves presented in (Bachu 2003) involve hydrostatic pressure based on the density of the surrounding brine, whereas the hydrostatic pressure inside a CO_2 plume is based

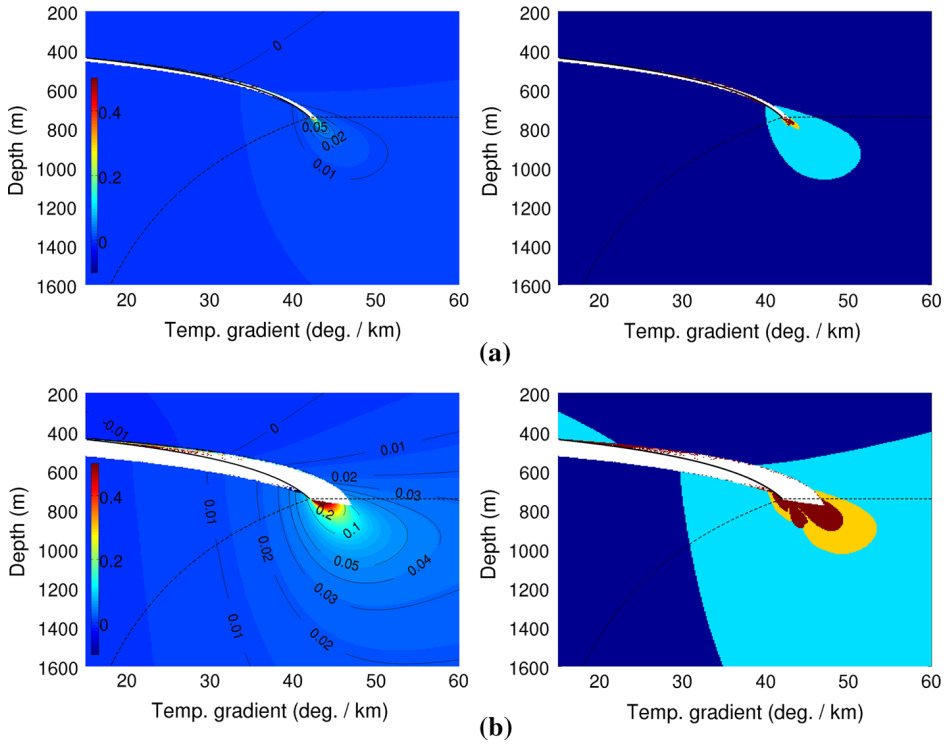


Fig. 3 Relative difference between vertically averaged and *bottom* (reference) CO_2 density for plume heights of **a** 20 m and **b** 100 m, superposed on the DTG diagram, with the vertical axis indicating the depth of ζ_M. Plots on the *left* display the density differences; plots on the *right* the corresponding zones of approximation for a selected tolerance of 1 %. In the *dark blue* regions, density differences are already smaller than this tolerance. In *light blue* regions, a first-order Taylor development is required to represent the variation in density within the tolerance. In *yellow regions*, a second-order development is required. In *red regions*, the second-order development is insufficient. The *white region* along the liquid–vapor boundary indicates the zone where the liquid–vapor boundary is located within the plume

on the density of the CO_2 itself, significantly lower than that of brine. Figure 4 shows a sample of some reconstructed density profiles under different conditions. It is possible that, in (likely) rare cases, this vertical density difference may cause gravitational instabilities within the plume. We have however not further investigated this possibility in the present work.

3.3 Vertical Differences in Mass Flux

We will now study the influence of vertical density variations on CO_2 mass flux. With the sharp-interface assumption, we see from (45) that the expression for vertically integrated CO_2 mass flux in the incompressible case is

$$F_{\text{incomp}} = -\rho_{c,R}\frac{h}{\mu_c}\mathbf{k}_{||}\left(\nabla_{||}p_R - \rho_{c,R}(g_\perp \nabla_{||}\zeta_R + g_{||}\mathbf{e}_{||})\right). \tag{48}$$

Similarly, from (38) we see that the corresponding expression in the fully compressible case is

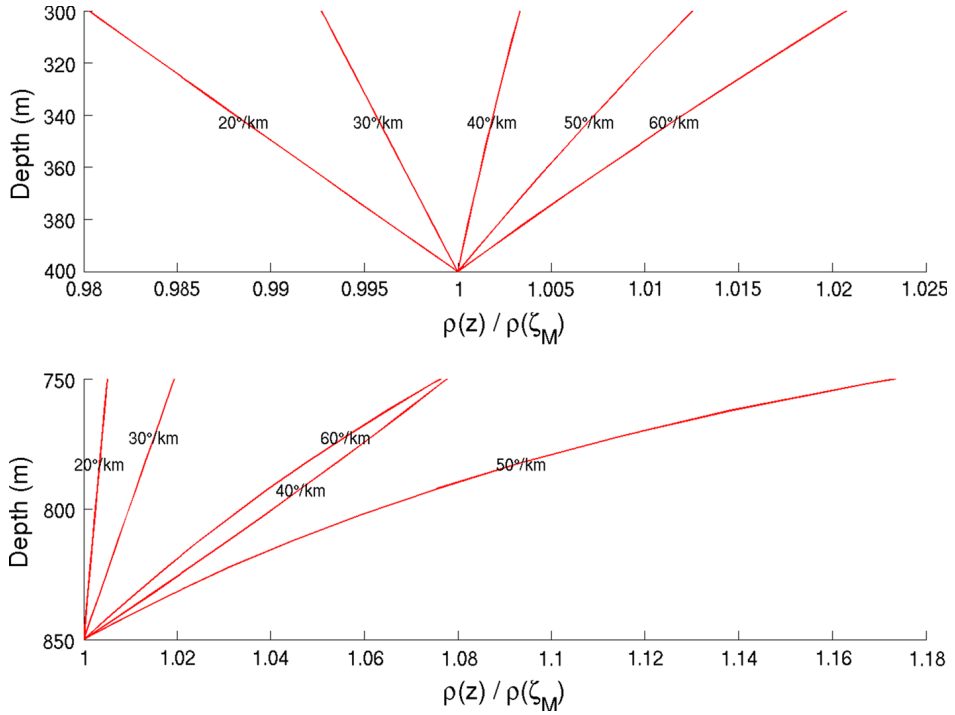


Fig. 4 Vertical CO_2 density profiles within a 100-m thick plume, for interface depth of 400 m (*top*) and 850 m (*bottom*), and with different temperature gradients. (Sea floor temperature considered to be 4 °C, at a depth of 100 m). In most situations, density decreases with depth within the plume

$$F_{\text{comp}} = -\mathscr{J}_2(h)\frac{h}{\mu_c}\mathbf{k}_{||}\left(\nabla_{||}p_R - \rho_{c,R}(g_{\perp}\nabla_{||}\zeta_R + g_{||}\mathbf{e}_{||})\right)$$
$$= \frac{\mathscr{J}_2(h)}{\rho_{c,R}}F_{\text{incomp}}. \tag{49}$$

The relative difference in magnitude is therefore

$$\frac{||F_{\text{comp}} - F_{\text{incomp}}||}{||F_{\text{incomp}}||} = \frac{1}{\rho_R}\mathscr{J}_2(h) - 1. \tag{50}$$

Figure 5 shows the resulting relative errors and approximation regions on the DTG diagram. Perhaps the most striking feature here is that the size and shape of the contour regions differ significantly from those in Fig. 3. A large part of the gas phase region has a 1 % or greater difference in mass flux, even while Fig. 3 shows that there is almost no corresponding difference in density. On the other hand, much of the supercritical region has a greater than 1 % difference in density, but not in mass flux.

It may seem surprising to find a noticeable difference in mass flux magnitude even at conditions where the density profile remains virtually constant in depth. However, at the fine scale, the mass flux is a product of a volumetric flux and a density. The density varies with depth according to ODE (16), which is governed by two terms that tend to cancel out (involving the coefficients of isothermal compressibility and of isobaric thermal expansion). On the other hand, the variation of the volumetric flux is scaled by the ν_p function, defined by ODE (24), involves the isobaric compressibility coefficient only. The function ν_p describes

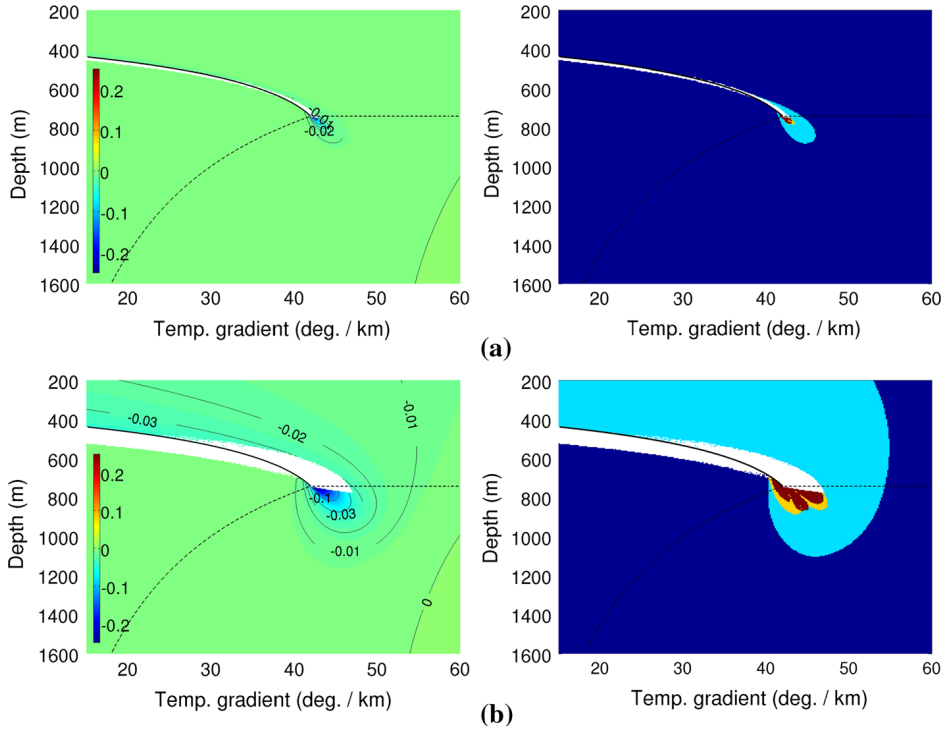


Fig. 5 *Left* Relative differences in vertically integrated CO_2 mass flux magnitude between the fully compressible and the incompressible model, for a plume height of **a** 20 m and **b** 100 m, assuming a horizontal flow plane. Negative values signify lower mass flux in the compressible model. *Right* plot of the corresponding zones of approximation, for a selected tolerance of 1 %. The coloring of zones has the same meaning as in Fig. 4

how, for a given depth, local pressure reacts to changes in reference pressure. At any given depth temperature is constant, and v_p thus describes a change in pressure that is *not* counteracted by first-order thermal effects. As such, even under conditions where density remains fairly constant in depth, the changes in volumetric flux can in some cases lead to noticeable differences in mass flux under the fully compressible model.

4 Simulated Test Cases

To assess the impact of variable density, we ran numerical simulations on three simple scenarios based on the sharp-interface model. In each scenario, we compared the incompressible, the semi-compressible, and the fully compressible VE model for the CO_2 phase (brine density assumed constant).

The simulator code we implemented employs a fully implicit numerical scheme using automatic differentiation (Neidinger 2010), within the framework of the MATLAB Reservoir Simulation Toolbox (Lie et al. 2010). Automatic differentiation is based on the idea that all simulation variables, as well as all quantities derived thereof, are represented by numeric types that also contain complete information about their partial derivatives with respect to each simulation variable. Any computation involving these types propagates and updates the derivative information using standard derivative rules. Since the system Jacobians are

obtained directly as a side effect of computing the expressions themselves, automatic differentiation allows for rapid and easy implementation of fully implicit model prototypes using a standard Newton-Raphson approach.

Values for CO_2 density are obtained using interpolation of sampled tables computed according to the equation of state proposed in (Span and Wagner 1996). The reference surface is set to the brine–CO_2 interface ζ_M. For the fully compressible model, ρ and ν_p are approximated using a second-order Taylor expansion from this reference surface. Although the accuracy of this approach degrades when approaching the critical point, we conclude that the results are accurate enough for the purpose of comparing the models in the scenarios investigated. This is supported by good comparisons with full 3D simulations (presented only for Scenario 1).

In order to obtain close-to-critical conditions for CO_2, all our scenarios consider relatively high values for the thermal gradient. For low thermal gradients, the depths corresponding to critical temperature and critical pressure, respectively, would be farther apart, and vertical variation in CO_2 density would be less important.

4.1 Scenario 1: CO_2 Injection into Horizontal Aquifer

In this scenario, we model injection of CO_2 into a horizontal aquifer with impermeable, flat caprock. We consider the case of injection along a horizontal well located at the bottom of the aquifer, aligned with the y-direction. Since we have symmetry along this direction, we model only an x–z slice of the domain, and by vertical integration the simulation is essentially reduced to a single dimension. The lateral extent of the resulting one-dimensional model is 80 km, which we discretize as a single row of cells with hydrostatic pressure boundary conditions at the two ends. The injection well is located in the center of this domain, 40 km from either boundary. Actual examples of horizontal wells for injection of CO_2 are the In Salah Gas Project in Algeria and the Weyburn Field in Canada (IPCC 2005).

The modeled aquifer is 150 m thick and located at a depth of 750 m, with a constant rock porosity of 0.2 and permeability of 400 mD, roughly comparable to, e.g., parts of the Carrizo-Wilcox aquifer (Hesse et al. 2008). We assume initial pressure to be hydrostatic, with a constant brine density of 1,050 kg/m^3. Temperature is given by a surface temperature of 279.15 K (6 °C) and a geothermal temperature gradient of 40 K/km.

We consider an injection at a rate of 3.33 megatons of CO_2 per year per unit (kilometer) of injection well for 19 years, followed by a 51-year migration period (total simulation time of 60 years). A separate simulation is run for each of the three density models. Since the effects of vertical density variations increase with plume height, we aim to obtain a plume that does not spread out too quickly. Therefore, we choose viscosity values for CO_2 and brine that give a ratio close to 1 (5.3×10^{-2} and 5.4×10^{-2}cP, respectively). For the incompressible model, CO_2 density is set to 315.5 kg/m^3, which corresponds to density at fluid-static pressure and thermal equilibrium. Although this choice will be inaccurate in the presence of overpressure during injection, it is the better choice for describing CO_2 density in the post-injection phase.

In Fig. 6 below, we present the simulation outcomes at the end of the injection (year 19), the first year after injection has ended (year 20), and the end of the simulation (year 60). Each column presents, for a particular year, the plots for interface depth, CO_2 pressure, density, and mass flux across the extent of the simulation domain. The depth plot has its second axis inverted for a more intuitive representation of the plume shape.

In the figure we note differences between the incompressible and compressible models both in terms of interface position, pressure, density, and mass flux. Given that all three plumes contain the same amount of mass, we note that the plume of incompressible CO_2 has

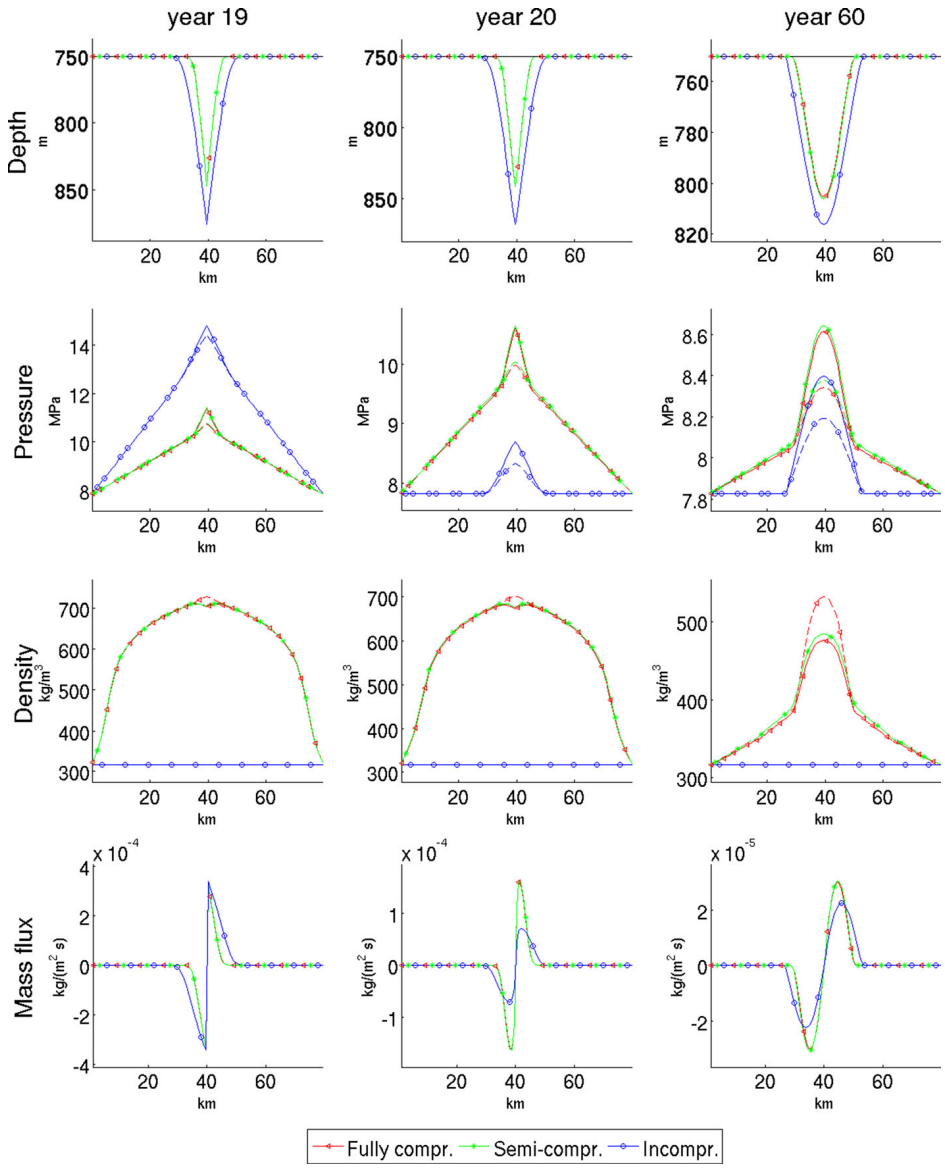


Fig. 6 Graphs representing CO_2 depth profile, pressure, density, and mass flux for selected years in Scenario 1. Graphs in *blue* (*marked with circles*) represent results from using the incompressible CO_2 model, the *green graphs* (*stars*) represent the semi-compressible model, and *red graphs* (*triangles*) the fully compressible model. For pressure and density, *solid lines* represent values at the CO_2–water interface, and *dashed lines* represent values at the top of the aquifer. Note that for the density plots, only the fully compressible model has separate lines for *top* and interface values, as the other models assume constant vertical density

lower density and hence a larger volume than the others. During injection, the overpressure in the incompressible model is significantly higher than for the other models, due to the larger volume of displaced fluid. After injection, the pressure drops instantly in the incompressible case, but only gradually in the compressible cases, where continuously expanding plumes push on the surrounding brine and drive flow toward the boundaries.

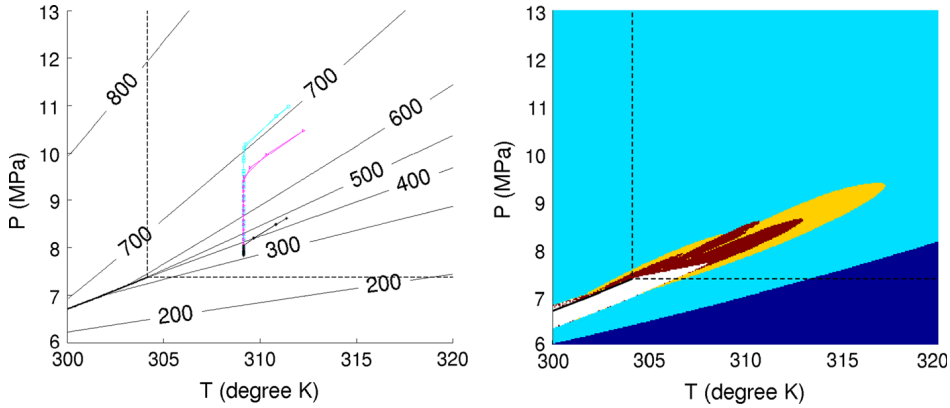


Fig. 7 *Left* Conditions at the CO_2–water interface for the fully compressible model in Scenario 1, plotted on the CO_2 phase diagram. The *cyan curve* (*squares*) represents year 19, the *magenta curve* (*triangles*) year 20, and the *black curve* (*stars*) year 60. For reference, the contour lines on the *left* plot represent isovalues for CO_2 density (in kg/m^3). *Right* Zones of approximation, defined in the same way as those for vertically averaged density in Fig. 3, computed for a plume height of 100 m and a tolerance of 1 %

Comparing the two compressible models, we find interface depths, pressures, and mass fluxes to be very similar. On the other hand, there are marked differences in terms of CO_2 density. Whereas the density at the interface in the fully compressible model is relatively close to, but slightly less than, that of the semi-compressible model, the density at the top is significantly higher. Despite this difference it should be noted that, when integrated across height, the total CO_2 masses in the two models for a given vertical column do not significantly differ. This can also be inferred from the respective depth plots, keeping in mind that the total plume masses are the same.

Figure 7 relates to the fully compressible model only. The left plot traces out the pressure and temperature values at the CO_2–water interface for year 19, 20, and 60 in the CO_2 phase diagram. The point where the three curves meet corresponds to the condition at the domain boundaries, whereas the opposite endpoints represent the deepest points of the respective plume interfaces. We note that the interface conditions at all times remain fully within the supercritical zone, with boundary conditions fixed relatively close to the critical point. The right plot shows the same part of the CO_2 phase diagram, this time with a color map representing zones of approximation similarly to the lower right plot of Fig. 3b (but using the phase diagram rather than the DTG diagram). Comparing the left and right plots of Fig. 7, we see that conditions at the plume interface enter the red approximation zone, meaning that a second-order Taylor development is insufficient to compute average density within 1 % of the exact value for a 100-m tall plume. As such, our simulation result (which is based on Taylor approximation) is not exact within this tolerance. If higher precision is needed, the use of finely sampled tables or direct integration would be necessary. On the other hand, we note that although we purposefully designed the scenario to present conditions with high-density variations, the difference in the simulation results for the semi-compressible and the fully compressible model remains small.

In order to assess the accuracy of using the VE assumption in this scenario, we compare the simulation outcome using the fully compressible VE model with that of a full 3D simulation. Similar to the VE simulations, the 3D simulation uses a fully implicit, finite-volume numerical scheme solved using automatic differentiation and Newton-Raphson. A uniform spatial resolution is used, with 100 cells along the x-axis and 90 along the y-axis. We use

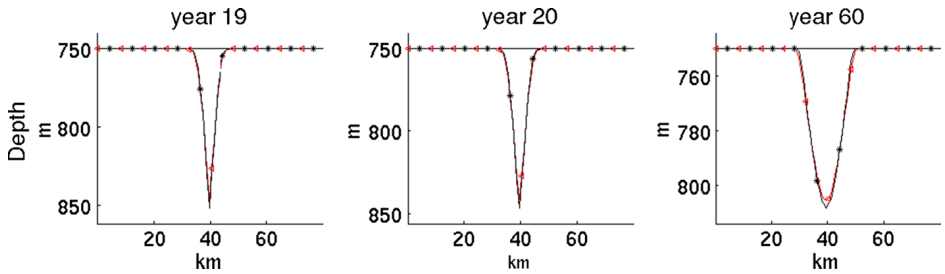


Fig. 8 Comparing the depth profile from a simulation using the compressible VE model with that obtained from a full 3D simulation. The *red curves* (*triangles*) represent the VE model and the *black curves* (*stars*) the full 3D model

linear relative permeabilities to limit differences caused by the vertical discretization. As a sharp interface is assumed, we use zero capillary pressure. The comparison is presented in Fig. 8, where we see a good correspondence between the resulting depth profiles. Similar correspondences were also seen for pressure and density (not shown here).

4.2 Scenario 2: Gravity-Driven Flow of CO_2 Along a Sloping Aquifer

We now consider gravity-driven flow of a CO_2 plume as it migrates upward along a gently sloping, open aquifer. The aquifer caprock is assumed flat, with a constant upward slope of one-half degree toward the right. We simulate flow along a 160 km stretch of this slope, the bottom of which is located at a depth of 1,397 m, and the top reaches the surface at 0 m. As in the previous scenario, we consider a situation symmetrical along the y-axis, and therefore model our domain in one dimension as a single row of cells. Rock porosity is 0.1, permeability 200 mD, and the aquifer has a constant thickness of 200 m. The geothermal gradient is 45 K/km, with a surface temperature of 9 °C, and brine density is set to 1,100 kg/m^3. Under these conditions, CO_2 will transition from supercritical to gas phase somewhere in the middle of the simulated domain, passing close to but avoiding the critical point. We impose hydrostatic pressure boundary conditions at the two end points, no flow conditions everywhere else, and set fluid viscosities to 5.36×10^{-2} cP for CO_2 and 6.5×10^{-1} cP for brine. We simulate the gravity-driven flow of a CO_2 plume of 30 megaton per kilometer in the y-direction, initially positioned at a depth of 877 m, as it slowly migrates upward for a total simulation time of 400 years, using 10-year timesteps. The initial plume has the shape of an inverted triangle with a base of 7.6 km and an maximal height of 164 m for the fully compressible model. Initial pressure at the CO_2–brine interface is hydrostatic, and the plume is in vertical equilibrium. For the incompressible model, CO_2 density is set to 433 kg/m3, which corresponds to the density at the CO_2–brine interface for the lower most point of the initial plume (according to the fully compressible model).

Figure 9 presents the simulation outcomes for years 40 and 400. We observe that although our simulation parameters were chosen to highlight density variation (as seen on the density plot for year 40), the semi-compressible and the fully compressible model have virtually identical plume profiles, pressures, and mass fluxes. On the other hand, we note an appreciable difference in the advancement of the plume between the compressible and incompressible models toward the end of the simulation period, where the compressible plumes have advanced about fifteen kilometers further than the incompressible one. The compressible plumes also end up being more voluminous than the incompressible one, due to the considerable decrease in CO_2 density as the plume is moving upward. We also notice differences

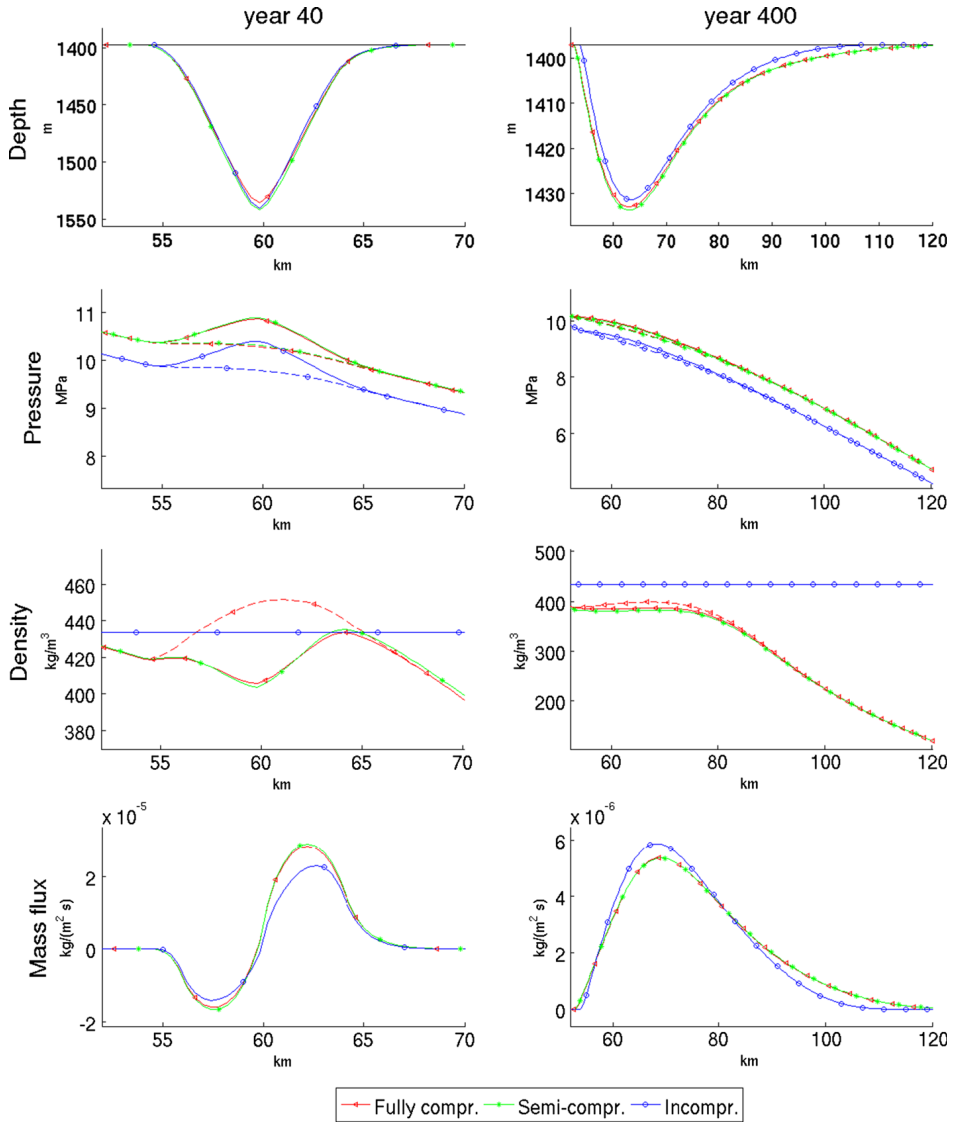


Fig. 9 Graphs representing CO_2 depth profile, pressure, density, and mass flux for year 40 and 400 in Scenario 2. Graphs in *blue* (*circles*) represent results from using the incompressible CO_2 model, the *green graphs* (*stars*) represent the semi-compressible model, and *red graphs* (*triangles*) the fully compressible model. For pressure and density, whole *lines* represent values at the CO_2–water interface, and *dashed lines* represent values at the *top* of the aquifer. Note that for the density plots, only the fully compressible model has separate lines for *top* and interface values, as the other models assume constant vertical density. Also note that only the local plume region has been plotted for each year, not the entire 160 km domain

between the incompressible and compressible models in terms of mass flux, which is overally higher for the compressible model at year 40. On the other hand, at year 400, the compressible plume has a lower mass flux than the incompressible one toward the deep (left) side of the model, whereas the situation is reversed toward the shallow end. Since we do not include capillary or dissolution trapping in our model, and since no CO_2 has left the domain during simulation, the total mass of each plume remains identical.

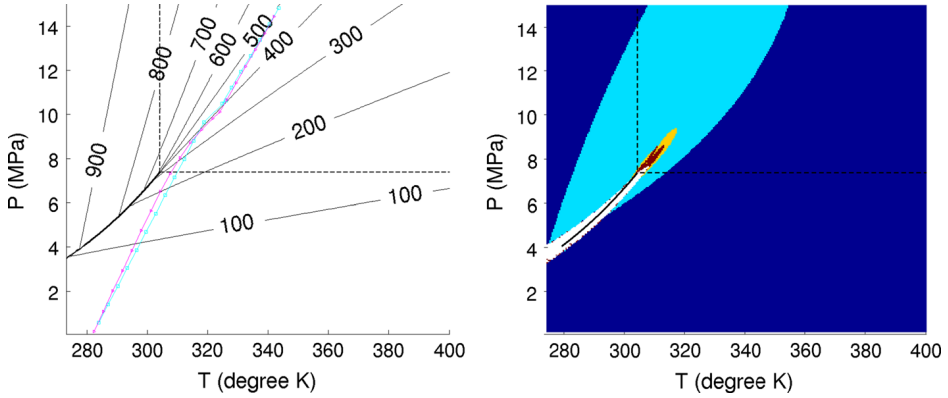


Fig. 10 *Left* Conditions at the CO_2–water interface for the fully compressible model in Scenario 2, plotted on the CO_2 phase diagram. The *cyan curve* (*squares*) represents year 40 and the *magenta curve* (*triangles*) year 400. For reference, the contour lines on the *left* plot represent isovalues for CO_2 density (in kg/m^3). *Right* Zones of approximation, defined in the same way as those for vertically averaged density in Fig. 3, computed for a plume height of 100 m and a tolerance of 1 %

Figure 10 presents the conditions on the CO_2–water interface of the fully compressible CO_2 plume, similarly to Fig. 7 for the previous scenario. On the left plot, we can see how these conditions gradually change from supercritical to gas phase for CO_2 as we move along the interface from the deep to the shallow end of the simulated aquifer domain. Comparing the left and right plots of the figure, we note that the interface conditions barely avoid the white region in the right plot. Passing through the white region would signify the crossing of the liquid–vapor boundary for a plume height of 100 m or more, thus implying the need of a more advanced model.

4.3 Scenario 3: CO_2 Injection into Structural Trap, at Conditions Close to Critical Point

In the last example, we study the injection and accumulation of CO_2 into a large dome. We here aim to obtain a plume of significant thickness and study it as it settles toward equilibrium. This time, we consider a bell-shaped domain of 10 × 10 km, measuring 160 m from top to bottom in the z direction. The apex is located at a depth of 700 m, and the spill point at 860 m. For the first 25 years, CO_2 is injected at a rate of 4 megatons per year (injection point located at the apex). Thereafter, we continue the simulation for another 175 years in order for the plume to settle within the confines of the dome. The injection rate of CO_2 has been chosen so that that no CO_2 spills out of the dome during the injection period for any of the models. The permeability is 1.1 Darcy, and the porosity 0.2. Brine density is 1,100 kg/m^3, and the temperature gradient is 40°/km, with a surface temperature of 6 °C. Water and CO_2 viscosities are 6.5×10^{-1} and 5.36×10^{-2} cP, respectively. As we expect the dome to fill up about half way, the CO_2 density for the incompressible model was chosen based on conditions at a depth of 775.4 m, close to the midpoint between apex and spill point depths, yielding a value of 503.4 6 kg/m^3. (If the apex had instead been chosen as reference, the density would have been significantly lower, at 330 kg/m^3). In order to reduce the influence of the constant-pressure boundary conditions, we extend the domain horizontally 10 km in each direction. The total extent of the simulated domain thus becomes 30 km × 30 km, discretized as a regular grid with 75 × 75 cells.

The simulation results for year 25, 26, and 200 are shown in Fig. 11. During the injection period, the injection pressure pushes CO_2 downward and to the sides. After injection ends,

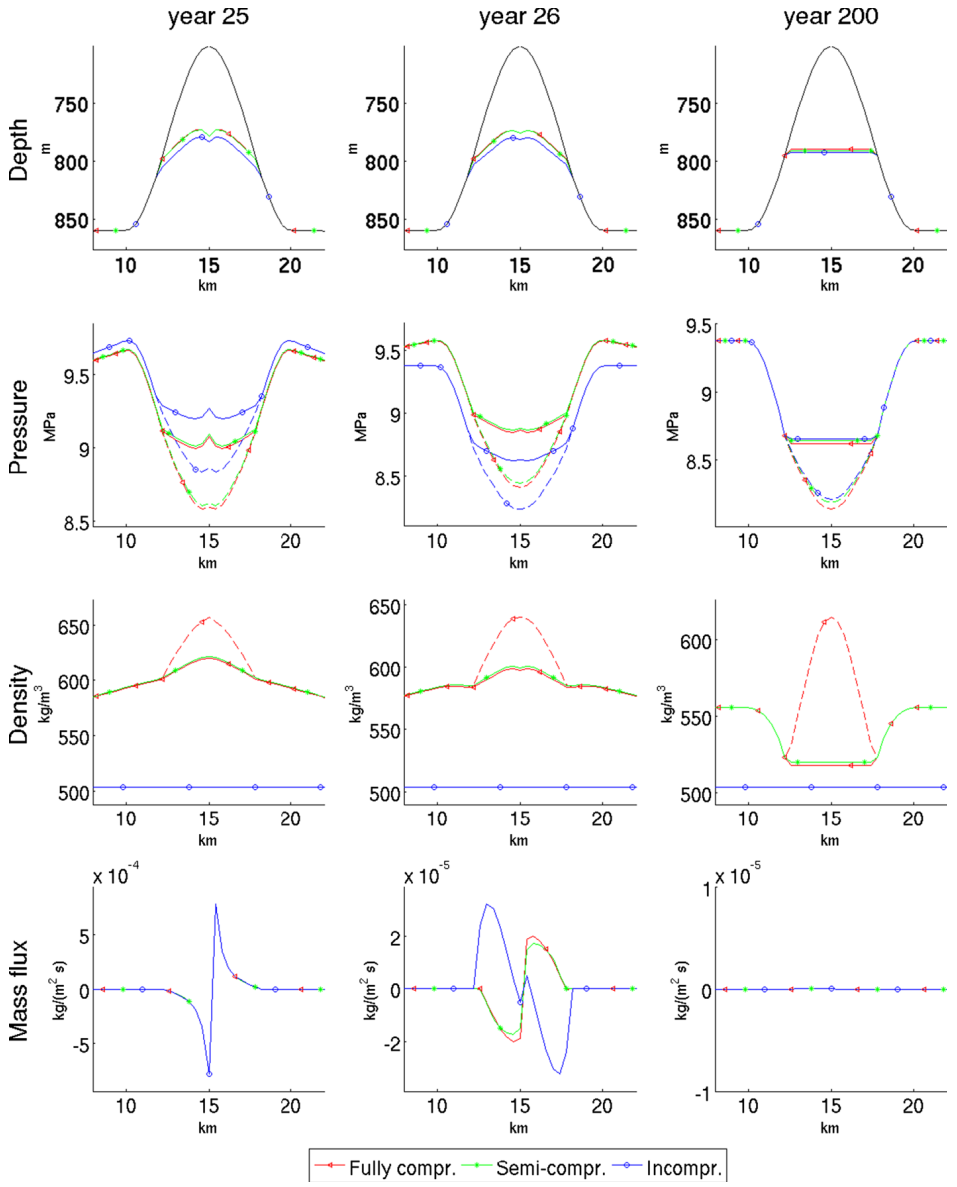


Fig. 11 Graphs representing CO_2 depth profile, pressure, density, and mass flux for selected years in Scenario 3. Graphs in *blue* (*circles*) represent results from using the incompressible CO_2 model, the *green graphs* (*stars*) represent the semi-compressible model, and *red graphs* (*triangles*) the fully compressible model. For pressure and density, whole lines represent values at the CO_2–water interface, and *dashed lines* represent values at the *top* of the aquifer. Note that for the density plots, only the fully compressible model has separate lines for *top* and interface values, as the other models assume constant vertical density. Note that only a cross-section of the local area around the dome is plotted

the pressure drops and the injected CO_2 slowly levels out toward an equilibrium state with a horizontal CO_2–brine interface. We note that the direction of the mass flux switches immediately in the incompressible case, whereas it changes only gradually for the compressible models.

At the last year of the simulation, equilibrium has almost been obtained, with a flat interface and negligible mass flux. The final interface positions for all three models closely match at this point. We also note fairly small pressure differences between the three models. The fully compressible model exhibits an intra-plume vertical density difference of 18.5 % (the top being heavier than the bottom). Note that this is an overestimation due to the use of Taylor expansion in a regime where the error term is large—the CO_2 density at caprock level obtained directly from the equation of state would yield a density difference closer to 15 %. In any case, the slightly different interface positions and densities help explain why all three plumes, all containing the same mass of CO_2, end up with almost identical thickness. Interface conditions remain firmly within the supercritical region and in relative proximity to the critical point at all times.

5 Summary and Conclusions

Models based on the assumption of vertical equilibrium have regained interest in recent years because of their potential for fast and accurate simulation of large-scale long-term CO_2 storage scenarios. The work presented in this paper is part of ongoing efforts to develop and adapt such models for more complex problems. To this end, we present a novel and mathematically consistent way of handling CO_2 density variations within the VE framework. The model formulation captures vertical variation in density through the definition and subsequent integration of two ordinary differential equations. We have also studied the impact of variable vertical density for a range of conditions and examined its influence in three specific scenarios. We have also made some additional observations, including the decreasing density with depth within the CO_2 plume, the interdependence of the upscaled quantities on the same set of fine-scale variables, and the need for a correction term in the semi-compressible model to remain consistent with the implied fine-scale flux.

While the presented model introduces a complete treatment of variable CO_2 density within a VE framework, it should be noted that we have made a number of simplifying assumptions in order to facilitate the model formulation and isolate the impact of CO_2 density variations. A more complete model would include physical effects such as capillarity, residual saturations, leakage of brine through caprock, and mixing of phases. As mentioned in the introduction, these effects have been modeled within the VE setting in previous work, and therefore may be included along with the vertical density model described herein for a more complete model. Moreover, whereas thermal equilibrium is assumed at all times, this would not be the case in the vicinity of the injection point. For instance, as discussed in (Vilarrasa et al. 2013b), the injection of liquid CO_2 would lead to smaller displaced volumes close to the well, resulting in lower induced overpressure overall.

The handling of vertical density changes in the vertical integration is not straightforward. Even if we approximate the involved differential equations using Taylor expansions, the inclusion of full vertical density variations involves computational overhead and introduces complexity to the simulation code. Although we have seen from the two previous sections that the model may predict considerable density differences between top and bottom of the plume, the impact on the actual mass movement is generally limited, and the much simpler semi-compressible model seems to produce very similar results in all scenarios tested so far. The scenarios presented in the previous section were tuned to model conditions where density changes would be important, but despite these choices, the differences in plume shape, pressures, and fluxes remain small. This remained the case when we later tried to re-run the scenarios while allowing for variable viscosity. However, it is possible that the

large differences in vertical density might in itself be interesting for some applications, such as interpretation of seismic or microgravimetric measurements for monitoring purposes.

A couple of observations may provide some intuition on the reason mass flux changes so little with depth even in the fully compressible model. First, increases in pressure and temperature have opposite effects on CO_2 density, and these effects are similar in magnitude. As we have seen, in almost all cases, the temperature effect is strongest, causing CO_2 to expand slightly with depth within the plume. On the other hand, the change in pressure gradient with depth, described by ν_p, is not directly influenced by the temperature gradient. As a consequence, the model predicts the Darcy volumetric flux to increase slightly with depth. Multiplied by a density that *decreases* slightly with depth, the combined impact of depth on the resulting mass flux tends to be small.

The effect of variable density becomes progressively stronger as one approaches the critical point. However, at the same time the accuracy of the Taylor approximation deteriorates, and ρ and ν_p would eventually need to be computed in a different way. Moreover, when the crossing of the liquid–vapor boundary occurs within the interior of the plume, a more careful approach coupled with a thermal model would be required.

The results of this study have several implications for the role of variable CO_2 density in realistic storage projects. First, a complete model is now available that allows for either full- or semi-compressibility of CO_2, which before was either approximated as a constant value or in some other approximate way. Second, a systematic approach is proposed to identify conditions where the full compressibility model is needed to satisfy the required accuracy. In doing so, we may conclude that the vast majority of long-term, large-scale storage systems require only a semi-compressible model, which substantially reduces the complexity of the computational implementation and execution. It remains the subject of future work to determine if the full model is needed more often in coupled systems in which a more accurate vertical description of density and pressure are required. In general, this work gives the needed flexibility to move between models of different complexity as the specific CO_2 storage problem dictates.

Acknowledgments This work was supported by the Mathematical Modeling for Risk Assessment (MatMoRA-II) project, financed by Statoil Petroleum AS and the Research Council of Norway through grant no. 215641 from the CLIMIT programe.

Appendix A: Deriving the ODEs for ν_p, ν_ζ, and ν_G

The coefficients ν_p, ν_ζ and ν_G are functions of z, defined as follows:

$$\nu_p = \frac{\partial p}{\partial p_R} \tag{51}$$

$$\nu_\zeta = \frac{\partial p}{\partial \zeta_R} \tag{52}$$

$$\nu_G = \frac{\partial p}{\partial T} \tag{53}$$

with p being the hydrostatic pressure defined in terms of pressure p_R at some reference level ζ_R, i.e.,

$$p = p_R + g_\perp \int_{\zeta_R}^{z} \rho(p, T)\, \mathrm{d}z'. \tag{54}$$

The coefficients ν_p, ν_ζ, and ν_G are defined by ordinary differential equations that can be derived by partial differentiation of (54) with respect to p_R, ζ_R, and T, respectively.

A.1 Deriving the ODE for ν_p

To find ν_p, we take the partial derivative of (54) with respect to p_R and get

$$\frac{\partial p}{\partial p_R} = \nu_p = \partial_{p_R} \left(p_R + g_\perp \int_{\zeta_R}^{z} \rho \, \mathrm{d}z' \right)$$

$$\nu_p = 1 + g_\perp \int_{\zeta_R}^{z} \frac{\partial \rho}{\partial p_R} \, \mathrm{d}z'$$

$$\nu_p = 1 + g_\perp \int_{\zeta_R}^{z} \left(\frac{\partial \rho}{\partial p} \frac{\partial p}{\partial p_R} + \frac{\partial \rho}{\partial T} \frac{\partial T}{\partial p_R} \right) \mathrm{d}z'$$

$$\nu_p = 1 + g_\perp \int_{\zeta_R}^{z} \rho \beta \frac{\partial p}{\partial p_R} \, \mathrm{d}z'$$

$$\nu_p = 1 + g_\perp \int_{\zeta_R}^{z} \rho \beta \nu_p \, \mathrm{d}z'.$$

Note that T is independent of pressure, which explains why the corresponding partial derivative term disappears in the integrand above. By taking the z-derivative of both sides, we obtain the following ODE:

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}z} \nu_p(z) = g_\perp \rho(z) \beta(z) \nu_p(z) \\ \nu_p(\zeta_R) = 1 \end{cases} \tag{55}$$

A.2 Deriving the ODE for ν_ζ

To find ν_ζ, we take the partial derivative of (54) with respect to ζ_R and get

$$\frac{\partial p}{\partial \zeta_R} = \nu_\zeta = \partial \zeta_R \left(p_R + g_\perp \int_{\zeta_R}^{z} \rho \, \mathrm{d}z' \right)$$

$$\nu_\zeta = g_\perp \left(\partial_{\zeta_R} \int_{\zeta_R}^{z} \rho \, \mathrm{d}z' \right)$$

$$v_\zeta = -g_\perp \rho_R + g_\perp \int_{\zeta_R}^{z} \left(\frac{\partial \rho}{\partial p} \frac{\partial p}{\partial \zeta_R} + \frac{\partial \rho}{\partial T} \frac{\partial T}{\partial \zeta_R} \right) \mathrm{d}z'$$

$$v_\zeta = -g_\perp \rho_R + g_\perp \int_{\zeta_R}^{z} \rho\beta \frac{\partial p}{\partial \zeta_R} \mathrm{d}z'$$

$$v_\zeta = -g_\perp \rho_R + g_\perp \int_{\zeta_R}^{z} \rho\beta v_\zeta \, \mathrm{d}z'.$$

Note that T is independent of the position of the interface ζ_R, so the corresponding partial derivative term disappears in the integrand above. By taking the z-derivative of both sides, we obtain the following ODE:

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}z} v_\zeta(z) = g_\perp \rho(z) \beta(z) v_\zeta(z) \\ v_\zeta(\zeta_R) = -g_\perp \rho_R \end{cases} \tag{56}$$

By comparing (55) and (56), we see that if v_p solves (55), then $(-g_\perp \rho_R v_p)$ solves (56). We therefore have

$$v_\zeta = -g_\perp \rho_R v_p. \tag{57}$$

A.3 Deriving the ODE for v_G

To find v_G, we take the partial derivative of (54) with respect to T, and get

$$\frac{\partial p}{\partial T} = v_G = \partial_\mathrm{T} \left(p_R + g_\perp \int_{\zeta_R}^{z} \rho \, \mathrm{d}z' \right)$$

$$v_G = g_\perp \int_{\zeta_R}^{z} \left(\frac{\partial \rho}{\partial p} \frac{\partial p}{\partial T} + \frac{\partial \rho}{\partial T} \right) \mathrm{d}z'$$

$$v_G = g_\perp \int_{\zeta_R}^{z} \rho \left(\beta \frac{\partial p}{\partial T} - \gamma \right) \mathrm{d}z'$$

$$v_G = g_\perp \int_{\zeta_R}^{z} \rho \left(\beta v_G - \gamma \right) \mathrm{d}z'.$$

Note that fine-scale hydrostatic pressure does depend on T (since temperature influences the density of the substance in the column between ζ_R and z, which again influences the hydrostatic pressure). Therefore, both partial derivatives of ρ are needed in the integrand above.

By taking the z-derivative of both sides, we obtain the following ODE:

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}z} v_G(z) = g_\perp \rho(z) \beta(z) v_G(z) - g_\perp \rho(z) \gamma(z) \\ v_G(\zeta_R) = 0 \end{cases} \tag{58}$$

However, v_G can be written directly in terms of v_p and ρ. To demonstrate this, it is sufficient to show that the functions v_G and $f(z) = \frac{g}{G}(\rho - \rho_R v_p)$ solve the same Cauchy problem. By inserting the derivatives of ρ and v_p, we obtain the following expression for the derivative of f:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}z}f(z) &= \frac{g}{G}\left(\frac{\mathrm{d}}{\mathrm{d}z}\rho(z) - \rho_R\frac{\mathrm{d}}{\mathrm{d}z}v_p(z)\right)\\ &= \frac{g}{G}\Big(\rho(z)\big(g_\perp\beta(z)\rho(z) - G_\perp\gamma(z)\big) - \rho_R g_\perp\rho(z)\beta(z)v_p(z)\Big)\\ &= \frac{g}{G}\Big(g_\perp\rho(z)\beta(z)\big(\rho(z) - \rho_R v_p(z)\big) - G_\perp\rho(z)\gamma(z)\Big)\\ &= g_\perp\rho(z)\beta(z)f(z) - g_\perp\rho(z)\gamma(z)\end{aligned}$$

Since also $f(\zeta_R) = 0$, B is the solution to

$$\begin{cases}\frac{\mathrm{d}}{\mathrm{d}z}f(z) = g_\perp\rho(z)\beta(z)f(z) - g_\perp\rho(z)\gamma(z)\\ f(\zeta_R) = 0\end{cases} \tag{59}$$

which is the same Cauchy problem as (58). We therefore have

$$v_G = f = \frac{g}{G}(\rho - \rho_R v_p). \tag{60}$$

Appendix B: Estimation of ρ and v_p by Taylor Expansion

When including full compressibility effects in a VE model, the functions $\rho(z)$ and $v_p(z)$ are used in the computation of upscaled variables. Here, the function ρ represents density as a function of z. Given a reference height ζ_R and the phase pressure p_R at this reference height, v_p is defined as

$$v_p(z) = \frac{\partial p(z)}{\partial p_R}.$$

The Taylor developments of ρ and v_p can be expressed in terms of the following functions defined from the equation of state:

$$\begin{aligned}\beta &= \frac{1}{\rho}\frac{\partial\rho}{\partial p}\\ \gamma &= -\frac{1}{\rho}\frac{\partial\rho}{\partial T}\\ \beta^{[2]} &= \frac{1}{\rho}\frac{\partial^2\rho}{\partial p^2}\\ \gamma^{[2]} &= \frac{1}{\rho}\frac{\partial^2\rho}{\partial T^2}\\ \chi &= \frac{1}{\rho}\frac{\partial^2\rho}{\partial p\partial T}\end{aligned}$$

In what follows, $g_\perp$ and $G_\perp$, respectively, denote the z-components of the gravity vector and the thermal gradient vector (the z axis is oriented at an angle θ with the true downward direction). Values of γ, β, etc., at the reference height ζ_R are identified by appending a subscript $_R$ to the variable name.

B.1 Taylor Expansion of ρ

We here consider ρ as a function of vertical coordinate z.

The second-order development of ρ around ζ_R is written as

$$\rho(z) = \rho_R + \rho'_R[z - \zeta_R] + \frac{1}{2}\rho''[z - \zeta_R]^2 + [z - \zeta_R]^2\epsilon(z - \zeta_R) \tag{61}$$

with the following expressions for the derivatives of ρ

$$\begin{aligned}\rho' &= \left(\rho g_\perp \beta - \gamma G_\perp\right)\rho \\ \rho'' &= \left((g_\perp\rho)^2(\beta^2 + \beta^{[2]}) + g_\perp\rho\tilde{G}(2\chi - \gamma\beta) + G_\perp^2\gamma^{[2]}\right)\rho.\end{aligned}$$

The values of ρ and its derivatives, evaluated at ζ_R, become

$$\begin{aligned}\rho(\zeta_R) &= \rho_R \\ \rho'(\zeta_R) &= \left(g_\perp\rho_R\beta_R - G_\perp\gamma_R\right)\rho_R \\ \rho''(\zeta_R) &= \left((g_\perp\rho_R)^2(\beta_R^2 + \beta^{[2]}) + g_\perp\tilde{G}\rho_R(2\chi_R - \gamma_R\beta_R) + G_\perp^2\gamma_R^{[2]}\right)\rho_R.\end{aligned}$$

B.2 Taylor Expansion of ν_p

We here consider ν_p as a function of vertical coordinate z.

The second-order development of ν_p in z around ζ_R is written as

$$\nu_p(z) = \nu_{pR} + \nu_{pR}{}'[z - \zeta_R] + \frac{1}{2}\nu_{pR}{}''[z - \zeta_R]^2 + [z - \zeta_R]^2\epsilon(z - \zeta_R) \tag{62}$$

with the following expressions for the derivatives of ν_p

$$\begin{aligned}\nu'_p &= g_\perp\rho\beta\nu_p \\ \nu''_p &= (g_\perp\rho)\Big((g_\perp\rho)(\beta^2 + \beta^{[2]}) + G_\perp\chi\Big)\nu_p.\end{aligned}$$

The values of ν_p and its derivatives, evaluated at ζ_R, become:

$$\begin{aligned}\nu_p(\zeta_R) &= 1 \\ \nu'_p(\zeta_R) &= g_\perp\rho_R\beta_R \\ \nu''_p(\zeta_R) &= (g_\perp\rho_R)\Big((g_\perp\rho_R)(\beta_R^2 + \beta_R^{[2]}) + G_\perp\chi_R\Big)\end{aligned}$$

References

Andersen, O., Nilsen, H.M., Lie, K.-A.: Reexamining CO_2 storage capacity and utilization of the Utsira Formation. In: ECMOR XIV - 14^{th} European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8–11 September 2014. EAGE. (2014)

Bachu, S.: Screening and ranking of sedimentary basins for sequestration of CO_2 in geological media in response to climate change. Environ. Geol. **44**(3), 277–289 (2003)

Bear, J.: Dynamics of Fluids in Porous Media. Dover Books on Physics and Chemistry. Dover Publications, Incorporated, Mineola (1988)

Class, H., Ebigbo, A., Helmig, R., Dahle, H., Nordbotten, J., Celia, M., Audigane, P., Darcis, M., Ennis-King, J., Fan, Y., Flemisch, B., Gasda, S., Jin, M., Krug, S., Labregere, D., Naderi Beni, A., Pawar, R., Sbai, A., Thomas, S., Trenty, L., Wei, L.: A benchmark study on problems related to CO_2 storage in geologic formations. Comput. Geosci. **13**(4), 409–434 (2009)

CMG: User's Guide GEM. Computer Modeling Group, Ltd., Calgary (2009)
Coats, K., Dempsey, J., Henderson, J.: The use of vertical equilibrium in two-dimensional simulation of three-dimensional reservoir performance. SPE J. **11–1**, 63–71 (1971)
Court, B., Bandilla, K.W., Celia, M.A., Janzen, A., Dobossy, M., Nordbotten, J.M.: Applicability of vertical-equilibrium and sharp-interface assumptions in CO_2 sequestration modeling. Int. J. Greenh. Gas Control **10**, 134–147 (2012)
Dentz, M., Tartakovsky, D.: Abrupt-interface solution for carbon dioxide injection into porous media. Transp. Porous Media **79**(1), 15–27 (2009)
Dietz, D.: A theoretical approach to the problem of encroaching and by-passing edgewater. Proc. Akad. van Wet. **56–B**, 83–94 (1953)
Flemisch, B., Fritz, J., Helmig, R., Niessner, J., Wohlmuth, B.: Dumux: a multi-scale multi-physics toolbox for flow and transport processes in porous media. In: ECCOMAS Thematic Conference on Multiscale Computational Methods for Solids and Fluids. (2007)
Gasda, S., Nordbotten, J., Celia, M.: Vertical equilibrium with sub-scale analytical methods for geological CO_2 sequestration. Comput. Geosci. **13**(4), 469–481 (2009)
Gasda, S.E., Nilsen, H.M., Dahle, H.K.: Impact of structural heterogeneity on upscaled models for large-scale CO_2 migration and trapping in saline aquifers. Adv. Water Resour. **62**(Part C), 520–532 (2013)
Gasda, S.E., Nilsen, H.M., Dahle, H.K., Gray, W.G.: Effective models for CO_2 migration in geological systems with varying topography. Water Resour. Res. **48**(10), 10546 (2012)
Gasda, S.E., Nordbotten, J.M., Celia, M.A.: The impact of local-scale processes on large-scale CO_2 migration and immobilization. Energy Procedia 4, 3896–3903. In: 10th International Conference on Greenhouse Gas Control Technologies, 2011a
Gasda, S.E., Nordbotten, J.M., Celia, M.A.: Vertically averaged approaches for CO_2 migration with solubility trapping. Water Resour. Res. **47**(5), 5528 (2011b)
Hesse, M.A., Orr, F.M., Tchelepi, H.A.: Gravity currents with residual trapping. J. Fluid Mech. **611**, 35–60 (2008)
IPCC: IPCC Special Report on Carbon Dioxide Capture and Storage. Prepared by Working Group III of the Intergovernmental Panel on Climate Change (2005)
Lie, K., Krogstad, S., Ligaarden, I., Natvig, J., Nilsen, H., Skaflestad, B.: Discretisation on complex grids-open source matlab implementation. In: Proceedings of ECMOR XII-12th European Conference on the Mathematics of Oil Recovery, EAGE, Oxford, UK (2010)
Lie, K.-A., Krogstad, S., Ligaarden, I.S., Natvig, J.R., Nilsen, H.M., Skaflestad, B.: Open source MATLAB implementation of consistent discretisations on complex grids. Comput. Geosci. **16**, 297–322 (2012)
Ligaarden, I., Nilsen, H.: Numerical aspects of using vertical equilibrium models for simulating CO_2 sequestration. In: 12th European Conference on the Mathematics of Oil Recovery. (2010)
Mijic, A., LaForce, T.C., Muggeridge, A.H.: Co_2 injectivity in saline aquifers: the impact of non-darcy flow, phase miscibility, and gas compressibility. Water Resour. Res. **50**(5), 4163–4185 (2014)
MRST.: The MATLAB Reservoir Simulation Toolbox, version 2014a. http://www.sintef.no/MRST/ (2014). Accessed 12 November 2014
Neidinger, R.: Introduction to automatic differentiation and matlab object-oriented programming. SIAM Rev. **52**(3), 545–563 (2010)
Nilsen, H.M., Herrera, P.A., Ashraf, M., Ligaarden, I., Iding, M., Hermanrud, C., Lie, K.-A., Nordbotten, J.M., Dahle, H.K., Keilegavlen, E.: Field-case simulation of CO_2-plume migration using vertical-equilibrium models. Energy Procedia 4, 3801–3808. In: 10th International Conference on Greenhouse Gas Control Technologies, 2011
Nilsen, H.M., Lie, K.-A., Andersen, O.: Analysis of trapping capacities in the Norwegian North Sea using mrst-co2lab. submitted. (2014a)
Nilsen, H.M., Lie, K.-A., Andersen, O.: Fully implicit simulation of vertical-equilibrium models with hysteresis and capillary fringe. submitted. (2014b)
Nilsen, H.M., Lie, K.-A., Andersen, O.: Robust simulation of sharp-interface models for fast estimation of CO_2 trapping capacity. submitted (2014c)
Nordbotten, J., Celia, M., Bachu, S.: Injection and storage of CO_2 in deep saline aquifers: analytical solution for CO_2 plume evolution during injection. Transp. Porous Media **58**(3), 339–360 (2005)
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 CO_2 storage. Int. J. Greenh. Gas Control **9**, 234–242 (2012)
Nordbotten, J.M., Celia, M.A.: Geological Storage of CO_2: Modeling Approaches for Large-Scale Simulation. Wiley, Hoboken (2012)
Nordbotten, J.M., Dahle, H.K.: Impact of the capillary fringe in vertically integrated models for CO_2 storage. Water Resour. Res. **47**(2), 2537 (2011)

Olivella, S., Gens, A., Carrera, J., Alonso, E.: Numerical formulation for a simulator (code_bright) for the coupled analysis of saline media. Eng. Comput. **13**(7), 87–112 (1996)

Pruess, K.: The tough codes—a family of simulation tools for multiphase flowand transport processes in permeable media. Vadose Zone J. **3**, 738–746 (2004)

Schlumberger: ECLIPSE Technical Description. Schlumberger (2010)

SINTEF ICT: The MATLAB Reservoir Simulation Toolbox: Numerical CO_2 laboratory. (2014)

Span, R., Wagner, W.: A new equation of state for carbon dioxide covering the fluid region from the triple-point temperature to 1100 k at pressures up to 800 mpa. J. Phys. Chem. Ref. Data **25**(6), 1509–1596 (1996)

van der Meer, L.: The conditions limiting CO_2 storage in aquifers. Energy Conversion Management 34(911), 959–966. In: Proceedings of the International Energy Agency Carbon Dioxide Disposal Symposium, 1993

Vilarrasa, V., Bolster, D., Dentz, M., Olivella, S., Carrera, J.: Effects of CO_2 compressibility on CO_2 storage in deep saline aquifers. Transp. Porous Media **85**(2), 619–639 (2010)

Vilarrasa, V., Carrera, J., Bolster, D., Dentz, M.: Semianalytical solution for CO_2 plume shape and pressure evolution during CO_2 injection in deep saline formations. Transp. Porous Media **97**(1), 43–65 (2013a)

Vilarrasa, V., Silva, O., Carrera, J., Olivella, S.: Liquid CO_2 injection for geological storage in deep saline aquifers. Int. J. Greenh. Gas Control **14**, 84–96 (2013b)

Wheeler, J., Wheeler, M., et al.: Integrated parallel and accurate reservoir simulator. Technical report, TICAM01-25, CSM, University of Texas at Austin. (2001)

Yortsos, Y.: A theoretical analysis of vertical flow equilibrium. Transp. Porous Media **18**(2), 107–129 (1995)

II

Paper II

Spill-Point Analysis and Structural Trapping Capacity in Saline Aquifers Using MRST-co2lab

Halvor M. Nilsen, Knut-Andreas Lie, Olav Møyner, Odd A. Andersen
Computers & Geosciences, **75**, (2015)

Computers & Geosciences 75 (2015) 33–43

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Spill-point analysis and structural trapping capacity in saline aquifers using MRST-co2lab

Halvor Møll Nilsen*, Knut-Andreas Lie, Olav Møyner, Odd Andersen

SINTEF ICT, Department of Applied Mathematics, P.O. Box 124 Blindern, N-0314, Oslo, Norway

ARTICLE INFO

Article history:
Received 2 September 2014
Received in revised form
1 November 2014
Accepted 3 November 2014
Available online 7 November 2014

Keywords:
CO_2 storage
Structural traps
Spill-point analysis

ABSTRACT

Geological carbon storage represents a substantial challenge for the subsurface geosciences. Knowledge of the subsurface can be captured in a quantitative form using computational methods developed within petroleum production. However, to provide good estimates of the likely outcomes over thousands of years, traditional 3D simulation methods should be combined with other techniques developed specifically to study large-scale, long-term migration problems, e.g., in basin modeling. A number of such methods have been developed as a separate module in the open-source Matlab Reservoir Simulation Toolbox (MRST).

In this paper, we present a set of tools provided by this module, consisting of geometrical and percolation type methods for computing structural traps and spill paths below a sealing caprock. Using concepts from water management, these tools can be applied on large-scale aquifer models to quickly estimate potential for structural trapping, determine spill paths from potential injection points, suggest optimal injection locations, etc. We demonstrate this by a series of examples applied on publicly available datasets. The corresponding source code is provided along with the examples.

1. Introduction

Geo-storage of CO_2 has been proposed as a possible strategy for mitigating global climate change. Under this approach, CO_2 is injected underground into deep saline aquifers, deep unmineable coal seams, depleted petroleum reservoirs, and so on (Benson et al., 2005). Determining the maximum amount of CO_2 that can be injected and securely contained is a key question.

When injected into a water-bearing formation, density differences will drive CO_2 to form a separate mobile phase (the CO_2 plume), which is driven upwards by buoyancy. To prevent CO_2 from moving directly upward, it is injected into a permeable formation bounded upwards by a sealing caprock. Below this seal, CO_2 will spread out and slowly migrate in the upslope direction. Disregarding other trapping mechanisms, this migration continues until the plume encounters a trap in the top surface where CO_2 will accumulate. Once a trap is filled, excess CO_2 will spill over and keep migrating upwards to the next trap, and so on until the top of the formation is reached.

In the short term, structural and stratigraphic trapping are key mechanisms for geological storage of CO_2. Other important mechanisms include residual, dissolution, and mineral trapping. As CO_2 migrates upward within the rock formation, the tail of the plume will gradually withdraw, and the pore space will fill again with resident brine when CO_2 injection has stopped. However, some CO_2 will remain trapped as immobile droplets in the void space between rock grains by capillary pressure from the water. This is known as residual trapping. Over time, injected CO_2 will also dissolve in the brine, which is referred to as solubility trapping. The dissolved CO_2 will form carbonic acid that might react with the reservoir rock to precipitate carbonate minerals, leading to mineral trapping.

Herein, our primary concern is to develop methods that can quickly compute bounds on the overall capacity for structural trapping and suggest good positions for placing injection points. These methods have later been combined with simulation tools based on an assumption of vertical equilibrium (Nilsen et al., 2014a, 2014b) to provide a comprehensive toolbox for optimizing injection strategies and simulating large-scale containment in a thousand-year perspective (Andersen et al., 2014; Lie et al., 2014; Nilsen et al., 2014c).

Structural traps correspond to local maxima of the top surface. First-order estimates of the corresponding trapping can be determined using relatively simple geometrical/topological algorithms that compute traps and catchment areas for the top surface of a given grid model. Simple percolation methods can be used to estimate spill paths, assuming that CO_2 is injected at an

* Corresponding author.
E-mail addresses: Halvor.M.Nilsen@sintef.no (H. Møll Nilsen), Knut-Andreas.Lie@sintef.no (K.-A. Lie), Olav.Moyner@sintef.no (O. Møyner), Odd.Andersen@sintef.no (O. Andersen).

http://dx.doi.org/10.1016/j.cageo.2014.11.002
0098-3004/

infinitesimal rate. These methods are implemented as free and open-source software in the Matlab Reservoir Simulation Toolbox (MRST) (Lie et al., 2012; The MATLAB Reservoir Simulation Toolbox, 2014a). The `MRST-co2lab` module (SINTEF ICT, 2014) includes a graphical user interface for interactively exploring structural trapping, input routines for industry-standard input formats, and scripts to download, unpack and process publicly available datasets (Halland et al., 2011; Eigestad et al., 2008; Norwegian Computing Center, 2013). Altogether, MRST provides a good platform for supporting reproducible research, which we demonstrate herein by providing complete scripts for all examples.

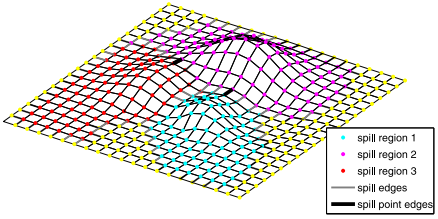


Fig. 2. Three interior and one exterior spill region with associated spill edges. The spill-point edge is the shallowest spill edge connected to a spill region.

2. The spill-point approach to trap analysis

We use the term *spill path* to refer to the path followed by CO_2 below the caprock on its buoyant migration, assuming infinitesimal flow. When a trap has been completely filled by CO_2, any additional quantity entering the trap will lead to an equivalent amount exiting. For infinitesimal quantities, the flow will follow a spill path out of the trap that either terminates in a different trap or exits the domain. As such, individual traps can be seen to be connected by spill paths, much the way lakes are connected by rivers.

In line with the convention of industry-standard reservoir descriptions, a surface is represented as a quadrangular mesh using a coordinate system whose z-axis is pointing downwards. We refer to this mesh as a *top-surface grid* when derived from a 3D reservoir model. Finding all traps and spill paths for a given model can be done by examining the geometry of this mesh. Although the process is simple in principle, it is sensitive to small changes in input data. We have therefore implemented two versions of the algorithm that work on dual interpretations of the quadrangular grid. The algorithms produce similar results in most cases, but sometimes the differences are significant. Comparing different outcomes can give an initial awareness of the uncertainty of the problem for a given top surface.

We refer to the two versions of the algorithm as the *corner-based* and *cell-based* approach. Both algorithms are based on the concept of *nodes* and *neighborhoods*. In the first version, nodes correspond to grid corners, whereas in the second version they correspond to cell centroids.

2.1. The basic trapping algorithm

We explain the key concepts and the basic algorithm for the corner-based algorithm, which treats the top-surface grid as a network of nodes and edges, with flow taking place between nodes along the edges. (The cell-based algorithm is similar, but uses cell centroids as nodes). To present the algorithm, some concepts used must be explained (see Fig. 1):

- the *neighborhood* of an interior node is defined as the node itself plus all nodes that are immediately connected;
- the shallowest node in a neighborhood is called a *local maximum*;
- the connection between a node and each of its surrounding neighbors defines a vector and the *upslope neighbor* of a node is the one with the steepest upward slope (if any).

The association between nodes and their upslope neighbors defines a directed tree, where infinitesimal, buoyancy-driven flow occurs along paths referred to as *spill paths*, connecting each node to its upslope neighbor. Each spill path ends up in a local maximum or at a boundary node. The *spill region* of an interior local maximum is defined as all nodes on paths leading into that maximum, see Fig. 2. All nodes on paths ending at a *boundary* node are assigned to the spill region of the exterior.

A *spill edge* is an edge in the mesh connecting nodes belonging to two distinct spill regions. When CO_2 flows upwards along a spill path, it will either exit the domain or accumulate near a local maximum until the surface of accumulated CO_2 reaches the shallowest spill edge of the associated spill region. This shallowest edge is called the *spill point edge*, the deepest of its two nodes called the *spill point*, and the corresponding depth is called the *spill depth*. (In degenerate cases, the spill point might not be unique). The pocket under the surface where CO_2 builds up before reaching the spill point is referred to as the *trap* associated with the local maximum. The mesh nodes within this trap are called *trap nodes* and define the *trap region* associated with this local maximum. The remaining part of the spill region is referred to as the *catchment area*.

Once the accumulated CO_2 reaches the spill point, it will start spilling out into a different spill region and follow the encountered spill path upwards. If this path ends up in a new local maximum, the trap from which CO_2 spilled out is said to be *upslope connected* to the trap associated with the new local maximum and the spill path is referred to as a *connection* or a *river*. The ensemble of traps and rivers form a directed graph. Traps associated with distinctly

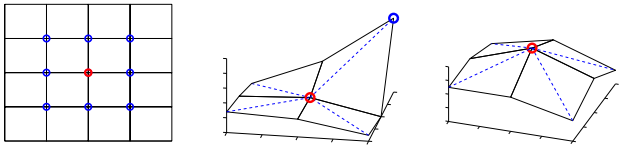


Fig. 1. *Left:* The neighborhood of an internal node consists of the node itself and its surrounding nodes. *Middle:* A node and its upslope neighbor marked in blue. *Right:* a local maximum does not have any upslope node. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

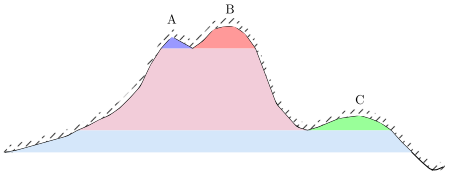


Fig. 3. Illustration of a trap hierarchy. *A*, *B* and *C* are local maxima with associated traps colored in blue, red and green. Trap C is upslope-connected to B, whereas A and B are upslope-connected to *each other* and are therefore subtraps of a larger trap AB shown in purple. Now, AB and C form local pockets of yet a larger trap structure ABC (cyan), whose spill point is at the far left of the domain. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

different local maxima may be upslope connected with each other, leading to cycles in this graph. This happens when the local traps constitute sub-pockets within a larger global trapping structure; as illustrated in Fig. 3. Any cycle in the graph can thus be replaced by a single trap created by combining the involved local maxima, see Fig. 4.

When two or more local traps combine in this way to define a larger trap, we say that they constitute *lower-level traps*, or *subtraps*, of a *higher-level trap*. The volume of the higher-level trap can be considerably larger than the combined volume of the lower-level traps. The process of detecting cycles and merging traps is repeated iteratively until the graph consisting of traps and rivers becomes acyclic. A trap that cannot be further merged is referred to as a *global trap* and is either upslope connected to another global trap or spills out of the domain.

Algorithm: Using the concepts explained above, the high-level description of the algorithm itself becomes:

1. Construct the graph $\mathcal{G}$ of individual nodes and their connections with upslope neighbors, one for each interior node that is not a local maximum. Connections to downslope neighbors are removed.
2. Assign each node to a unique spill region.
3. Identify spill edges.
4. Assign a spill-point edge to each spill region by choosing the highest of the spill edges crossing the boundary of the region.
5. Determine connections between local maxima by tracing the rivers leading from each region's spill-point edge(s) along the spill paths defined by $\mathcal{G}$. The gives a new graph $\mathcal{H}$ consisting of local maxima/traps connected by rivers.
6. Determine the global traps by iteratively removing cycles in $\mathcal{H}$ and merging the traps involved.
7. Compute the bulk volume geometrically inside each trap.

2.2. Interface and implementation in `MRST-co2lab`

To use the functionality in `MRST-co2lab`, the first thing one has to do is to create a semi-2D description of the top surface which includes a set of data objects that provide mapping between each cell in the 2D surface grid and a representation of the volumetric column that lie beneath in 3D. The top-surface grid can either be generated from a 3D volumetric grid using the function `topSurfaceGrid` or from a compatible set of depth and thickness maps. The common interface to the two versions of the algorithm described above is the function `trapAnalysis`. As an illustrative example of the type of information that can be obtained from this function, we consider a box geometry of $10{,}000 \times 5000 \times 50$ m^3. We introduce a sinusoidal perturbation of the top and bottom surfaces, move the box to a depth of 1000 m, and assign a uniform porosity of 0.25. Fig. 5 shows the 3D grid and the extracted top surface plotted on the same axes. Assuming that the surface represents an impermeable seal, the local domes will represent structural traps, which we identify using the trapping algorithms. All global traps can be identified from the array that associates an integer to each cell in the top-surface grid. For cells located *within* a trap, the integer represents the index of that trap; for other cells, the integer is a zero.

Next, a percolation type analysis can establish in which direction CO_2 will migrate from each cell if injected at an infinitesimal rate. The mapping between cells and spill regions is represented by a vector, with zero components for cells spilling out of the domain. The directed graph describing connections between traps is represented using an adjacency matrix, whereas cells on spill paths between traps are stored in a cell array. In the middle plot of Fig. 5, we visualize traps and connections.

In the non-degenerate case, each trap has either zero or one upslope connection. The graph of traps and connections will then consist of a set of separate trees. Each tree has a root, defined as a trap with no upslope neighbors, and a set of branches that each consists of traps connected to the root via rivers in the upslope direction. For each tree, we also define a set of leaf nodes, consisting of those traps which no other traps spill into. If any trap has more than a single upslope connection, the strict division into separate trees cannot be applied, but the concepts of roots, branches and leaf nodes remain meaningful, although a branch may now lead into more than one root node.

In Fig. 5, the directed graph consists of three trees: The largest tree emerges from trap number one, forms two main branches, and contains nine global traps. The second tree emerges from trap number two and consists of three global traps forming a single branch. The third tree consists of a single root, trap number three.

The total volume of a trap is limited by the top surface within the trap and the lateral plane at the spill depth. The pore volume of all traps can be computed using the function `computeTrapVolume`. For this particular model, the thirteen traps provide a combined trapping capacity amounting to 8% of the total pore volume. Bar plots of trap volumes and number of cells are shown to the right in Fig. 5.

`MRST-co2lab` also contains an interactive viewer that simplifies the trapping analysis. In forward mode, the user can select a point inside any trap or catchment area to identify the migration path(s) and upslope connected traps. Individual traps can also be inspected in more detail. In backward mode, the user can determine all traps downslope from a given point. Fig. 6 shows an example of using the viewer in forward mode. On the figure, cell number 1898 has been chosen as the injection point. This cell is within the catchment area of trap 11, which contains 13% of the total trap volume of the model. Further migration from here will reach traps 9, 6, 4 and 1, thereby utilizing 54% of the total trap volume. To produce the plot in Fig. 6, we pass parameters to `interactiveTrapping` specifying that we want to use the cell-based method, the model should be shaded, catchment areas displayed, and traps along the spill path marked using shades of gray.

The complete setup and all statements necessary to produce Figs. 5 and 6 can be found in the script `trappingExample1.m`.

3. Application to public data sets

In this section, we apply the tools described to different public data sets to estimate the potential for structural trapping. Spill paths, catchment areas, and structural traps are identified without any flow simulation, and this computation has a low cost and can

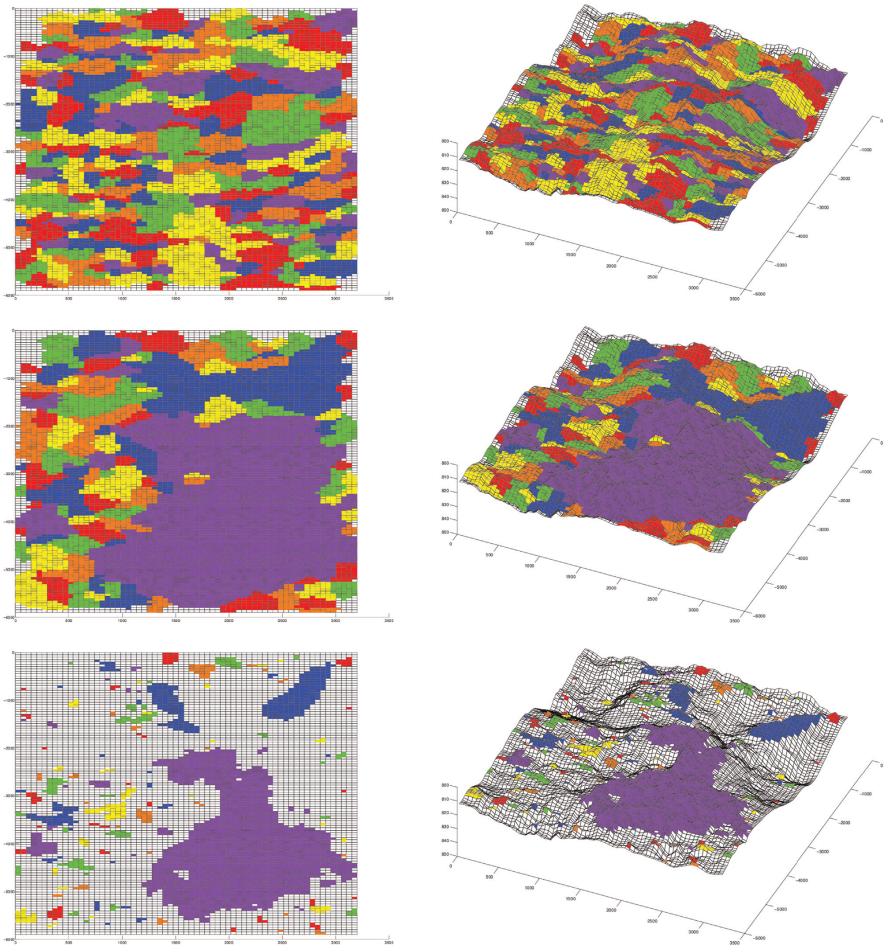


Fig. 4. By merging local maxima into global traps, 335 initial spill regions (top row) are reduced to 128 global spill regions (middle row) and traps (lower row).

be performed interactively even for large models. We therefore recommend that these simple geometrical analysis tools are used as starting points for more comprehensive simulation studies. By exploring the grid interactively one can, for instance, seek to determine the best possible injection site with regard to accessible structural trapping capacity, e.g., as discussed in detail in Andersen et al. (2014), Lie et al. (2014), and Nilsen et al. (2014c).

3.1. *Trapping capacity for North Sea aquifers*

The data sets published by the Norwegian Petroleum Directorate (NPD) as part of a recent CO_2 Storage Atlas (Halland et al., 2011) represent real aquifers from the Norwegian North Sea. These data cover large areas and are primarily meant for mapping. The extracted top-surface grids will be very coarse, but can still be used to provide indicative estimates of structural trapping capacity and likely outcomes for specific injection scenarios.

In the atlas, twenty-one geological formations have been individually assessed and grouped into saline aquifers considered as candidates for CO_2 injection. To establish a volumetric grid, a depth map of the top surface and a map of the formation thickness are required. The published data include formation thicknesses and depth maps for many of the formations in Fig. 7, but not all formations have both, and even when both are present, their coordinates do not always overlap. Nevertheless, using interpolation of non-matching, scattered data for the thickness maps, we were

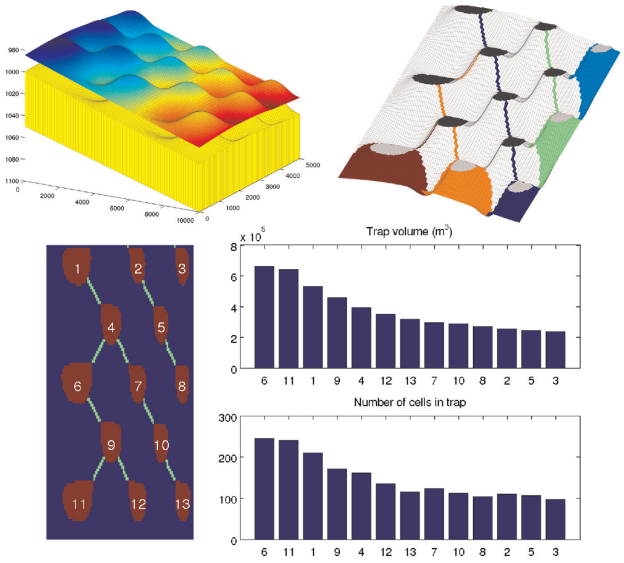


Fig. 5. Conceptual model illustrating the trapping structure. *Upper left:* The extracted top-surface grid has been shifted slightly upwards and colored by depth values to clearly distinguish it from the yellow 3D grid. *Upper right:* Leaf traps flattened at the spill level (light gray), the surrounding catchment areas (shown in different colors), and the spill-paths to the top of the domain; the spill levels of up-slope traps are shown in dark gray. *Lower left:* individual traps (red) and the rivers connecting them (green). *Lower right:* Pore volume and the number of cells contained in each trap. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

able to construct models of fourteen different sand volumes shown in Fig. 8. Table 1 reports estimated bulk volumes inside structural traps. The results are not directly comparable with Halland et al. (2011) which studies subsets or combinations of formations and incorporates petrophysical data and simulation-based adjustment factors that are not publicly available. We have chosen to present our results directly on individual formations as this gives unequivocal results that should be easy to reproduce for other researchers. In Andersen et al. (2014), Lie et al. (2014), and Nilsen et al. (2014c), the estimates reported herein are refined to account for geophysical rock properties and density variations in CO_2 with depth for the Sandnes, Skade, and Utsira formations. We also provide back-of-an-envelope estimates of the upper bounds on residual and solubility trapping.

As can be seen from Table 1, there are large differences in the fractions of the aquifer volumes contained within structural traps. There are also significant differences in the estimates provided by the two variants of the trapping algorithms, as many of the

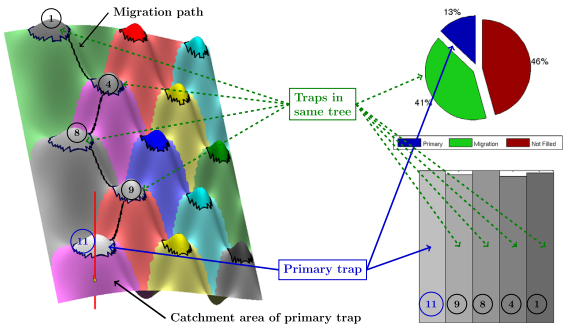


Fig. 6. Interactive view in forward mode. The pie chart shows the volume of the trap whose catchment area contains the injection point ('primary'), the combined volume of traps along the associated spill path ('migration'), and the volume of all other traps ('not filled'). The bar plot displays the volume of each trap along the spill path. The scale is logarithmic, since realistic scenarios tend to have widely varying trap sizes.

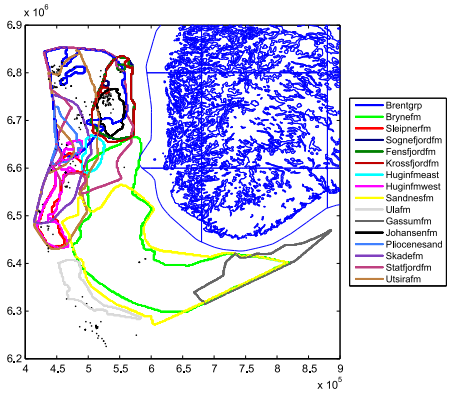


Fig. 7. Geological formations in the North Sea CO_2 Storage Atlas (Halland et al., 2011). The black dots indicate wells from the NPD's public database (http://factpages.npd.no/factpages/). The map of Norway comes from the Norwegian Mapping Authority (http://www.kartverket.no/) and is used for scale and rough positioning.

aquifers have traps with lateral sizes close to the data resolution. The complete source code of the example is given in script `describeAtlas.m`.

Of all models in the CO_2 Storage Atlas, the Hugin West Formation has the largest relative difference in trap volumes as estimated by the corner-based and cell-based algorithms, and thus serves as a useful case to highlight the differences between them, illustrating the instabilities inherent in the geometrical algorithm for identifying small traps. Fig. 9 shows that the top surface is relatively smooth and steep, with only eight identifiable structural traps. Compared with other formations, two of these are medium sized, while the remaining six are relatively small.

Because the two algorithms interpret the top-surface grid differently, they will generally assign a different number of cells and also compute different volumes for each trap. The corner-based method will tend to compute more precise spill points than the cell-based one, since cell centroids have originally been computed as grid corner averages. For trap number one, the cell-based method computes a shallower spill depth, and significantly fewer cells are identified as part of the trap. For trap number four and five, both approaches identify the same trap cells, but trap volumes differ due to different identified spill depths.

To avoid ambiguities, the cell-based method excludes cells whose centroid depths *equal* the depth of the trap's spill point, as such cells might belong to the boundary of multiple traps and their inclusion would in any case not contribute to the trap volume. For trap number six, the corner-based method computes the spill point at 2746.6 m, slightly deeper than the centroid of the cell identified as the spill point in the cell-based algorithm. This cell is therefore assigned as part of the trap only by the corner-based approach. The 20 cm depth difference may appear small, at least compared to the vertical resolution seen in 3D models for large-scale, long-term simulations, but leads to significantly different bulk volumes of the trap (3.5 vs. 3.2 million cubic meters) when trap height is multiplied by cell areas.

The corner-based approach does not necessarily include all cells surrounding a trap node as part of the trap. The projection of traps from corners to cells will only include cells whose centroids are shallower than the spill depth. Traps 2 and 3 in the corner-based approach consist of only one cell each. The spill depths determined by the cell-based algorithm for these two traps are somewhat deeper, thereby including one additional cell in each trap and yielding significantly different trap volumes.

In their current implementations, the corner-based method considers a 9-point neighborhood stencil, whereas the cell-based uses a 5-point stencil (similar to the industry-standard two-point discretization) and hence gives spill paths that tend to follow axial directions to a higher degree. This effect is seen on the boundary of the catchment areas in Fig. 9.

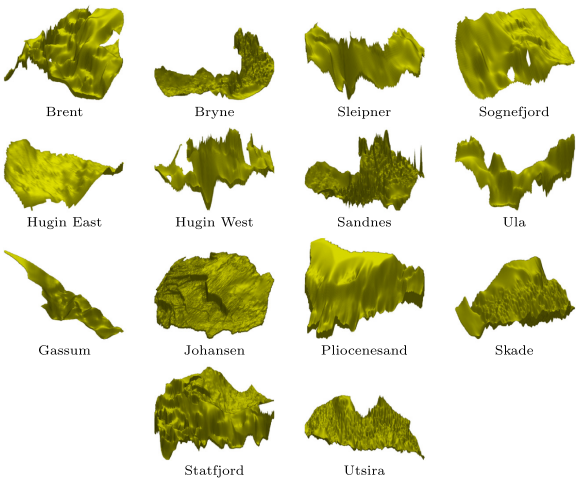


Fig. 8. Reconstructed grid models for fourteen different sand volumes derived from the public data sets accompanying the CO_2 Storage Atlas of the Norwegian North Sea (Halland et al., 2011).

Table 1
Total bulk volumes and bulk volumes inside structural traps for different formations from the CO_2 Storage Atlas (Halland et al., 2011) measured in cubic meters. Porosity and reference density must be supplied for each formation to derive estimates for CO_2 storage capacity.

Name	Cells	Depth		Rock	Corner-based traps		Cell-based traps	
		min	max	volume	volume	%	volume	%
Brentgrp	21,096	1659	5569	3.41e+12	9.63e+10	2.83	9.47e+10	2.78
Brynefm	46,585	271	4060	4.41e+12	3.33e+11	7.54	3.30e+11	7.48
Sleipnerfm	5116	2090	5323	2.63e+11	1.13e+09	0.43	1.07e+09	0.41
Sognefjordfm	9382	713	4145	9.06e+11	2.70e+10	2.98	2.91e+10	3.21
Huginfmeast	2264	2173	2893	9.26e+10	3.53e+08	0.38	2.89e+08	0.31
Huginfmwest	5513	1946	4617	3.75e+11	1.20e+09	0.32	8.34e+08	0.22
Sandnesfm	45,126	320	3411	1.55e+12	2.13e+11	13.75	2.13e+11	13.81
Ulafm	4544	2299	4536	3.99e+11	1.53e+09	0.38	1.61e+09	0.40
Gassumfm	35,043	305	3679	6.23e+11	4.78e+10	7.67	4.77e+10	7.65
Johansenfm	78,630	1822	3233	3.39e+11	2.60e+10	7.68	2.61e+10	7.71
Pliocenesand	13,520	260	641	2.90e+11	6.86e+07	0.02	7.15e+07	0.02
Skadefm	52,531	468	1257	2.33e+12	3.86e+09	0.17	3.86e+09	0.17
Statfjordfm	122,076	1636	6202	4.14e+12	9.37e+10	2.26	9.31e+10	2.25
Utsirafm	97,529	318	1391	3.84e+12	1.68e+10	0.44	1.68e+10	0.44

3.2. *Effects of data resolution: the Johansen Formation*

The Johansen Formation was previously proposed as a potential injection site in connection with the planned capture of CO_2 from the gas power plant at Mongstad. A full field model and four different sector models were developed in collaboration with NPD (Eigestad et al., 2009) and later made publicly available (Eigestad et al., 2008). A simplified subset of one of the sector models was later used in a code comparison study (Class et al., 2009; Wei and Saaf, 2009). Herein, we will use the same injection point as studied in Eigestad et al. (2009) and discuss qualitative differences in structural traps and spill-point paths computed on the full-field

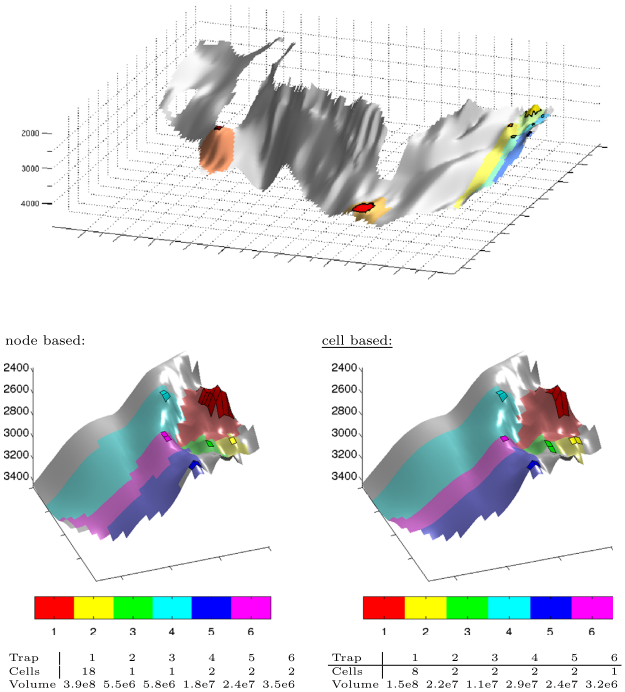


Fig. 9. Structural traps for the Hugin West Formation. *Upper plot:* The whole formation with structural traps and catchment areas given in different colors. *Lower plots:* The six traps identified by the two different algorithms in the southern parts of the formation located to the right in the upper plot; volume estimates are in cubic meters. Complete script: `trapsHuginWest.m`.

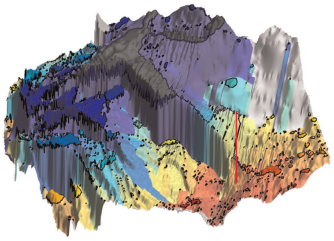


Fig. 10. Potential for structural trapping from a single injection point in the Johansen Formation predicted using a model from the CO_2 Storage Atlas (Halland et al., 2011). Dark colors signify structural traps, whereas light colors are the corresponding catchment areas. Light gray colors represent areas that spill to the perimeter of the model, whereas structural traps encountered along the spill path are shown in dark gray.

injection point is located in the accumulation area of a small trap that further connects to a series of small traps. An intricate migration path is followed that first moves up towards the blue–green area, then turns to follow the ridge at the sealing fault and finally ends in the large traps marked in dark gray color. The corresponding large trap for the 'full-field' model is shown in dark blue color, but is here not part of the migration path. Because of the lower resolution, most of the small-scale traps are not resolved, and the migration path ends up describing an almost straight line towards the crest of the model. Finally, notice that neither of the models provide information about the large fault.

3.3. Uncertainties in capacity estimates: the IGEMS data

Next, we will use a large ensemble of equiprobable top-surface realizations to illustrate the large uncertainties in capacity estimates that should be expected when working with models based

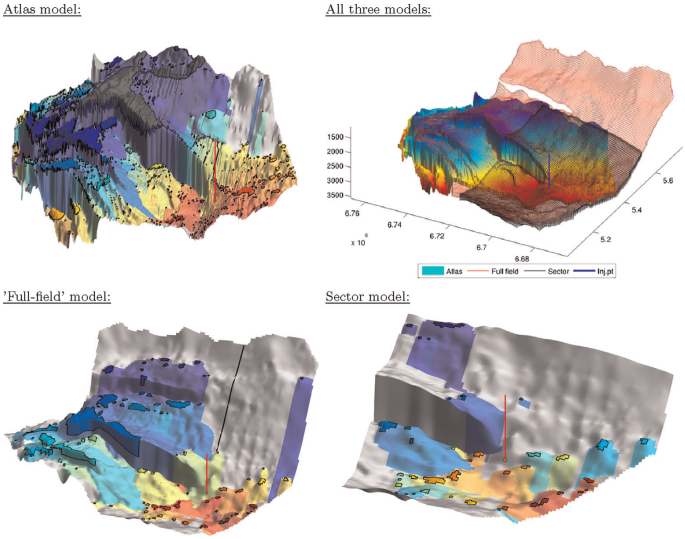


Fig. 11. Structural trapping for three different models of the Johansen Formation: the model from the CO_2 Storage Atlas (upper left) and a 'full field' model (lower left) and a sector model (lower right) developed by the Norwegian Petroleum Directorate. In the upper-right plot, the three models are superposed, with the Storage Atlas model colored by surface elevation, the sector model as a black grid, and the 'full-field' model as a red grid. Structural trapping is computed using the cell-based algorithm. Complete script for this example: `trapsJohansen.m`. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

model, one of the sector models ('NPD5'), and on a model with higher resolution derived from data supplied with the CO_2 Storage Atlas (Halland et al., 2011). The *quantitative* effects of using different spatial resolutions for the latter model are discussed in detail in Nilsen et al. (2014c). Fig. 10 shows the spill path, catchment areas, and structural traps for the atlas model, whereas Fig. 11 outlines the geographic extent of the three models and contrasts the differences in spill paths, catchment areas, and structural traps for all models.

The NPD models have much lower resolution than the atlas model and hence give smoother top-surfaces containing significantly fewer traps. The injection point is not located within a catchment area in the sector model. In the atlas model, the

on sparse data. The ensembles of synthetic aquifer models that make up the IGEMS data set (Norwegian Computing Center, 2013) were originally generated to investigate how variations in the top-surface morphology with a relief amplitude below seismic resolution would influence CO_2 storage capacity (Nilsen et al., 2012; Syversveen et al., 2012). Each model realization describes a large 30×60 km sandbox in the shape of an inverted gutter. Fifteen different types of top-surface morphologies were designed by combining three different stratigraphic scenarios—flat deposition, buried beach ridges in a flooded marginal-marine setting (FMM), and buried offshore sand ridges (OSS)—with five different structural scenarios: no faults, uniform (UP) or random (NP) fault displacement and length, and either a single 90° strike (1) or 30° and

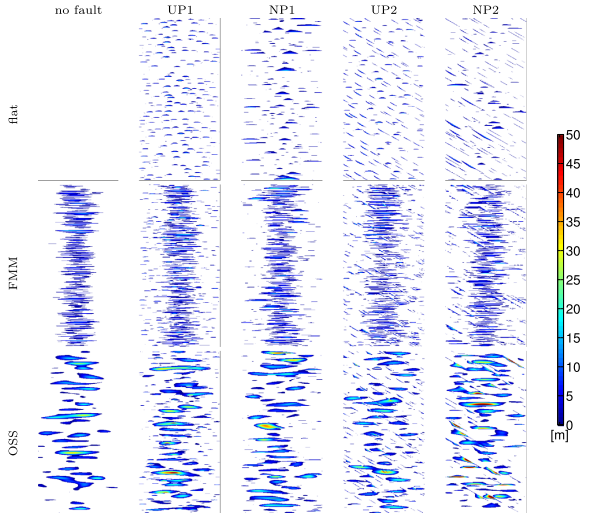


Fig. 12. Height inside structural traps in meters computed by the cell-based method for one realization of each of the fifteen different structural and stratigraphic scenarios from the IGEMS project. The *x*-axis is the 30 km east–west axis of the aquifer, and the *y*-axis the 60 km north–south axis. Complete script: `showIGEMS.m`.

90° strike directions (2). Offshore sand ridges with uniform faults and a single strike direction will hence be referred to as OSS UP1, and so on. Fig. 12 shows the corresponding fold and fault traps for each of the fifteen scenarios.

For each of these fifteen scenarios, one hundred different surfaces with a 100 × 100 m resolution were generated stochastically. Estimates of structural volumes reported in Nilsen et al. (2012) and Syversveen et al. (2012) were obtained using an early implementation of the cell-based method discussed herein. Fig. 13 compares average structural volumes computed by the corner-based and cell-based versions of the trapping algorithm. For the flat deposition there are only fault traps, and the two methods compute almost the same total volumes. Not surprisingly, the nonuniform structural cases, whose faults have 20–150 m displacement and 300–6000 m length, exhibit larger variation than the uniform cases, where faults have 100 m displacement and 4000 m length. The offshore sand ridge (OSS) cases are characterized by rather large lobes (amplitude <20 m, width 2–4 km, length 10–60 km, and spacing 2–4 km). Here, large fold traps dominate the smaller fault traps, and once again the relative deviations between the two methods are small. The flooded marginal-marine (FMM) scenarios, on the other hand, have much denser and smaller lobes (amplitude 1–10 m, width 10–300 m, length less than 15 km, spacing 40–300 m) which result in intricate patterns of fold and fault traps that are similar in size. Because the corner-based method always chooses spill points at least as deep as those identified by the cell-based method, the computed trap volumes will also be larger. In fact, for the flat, NP1, and OP1 structural scenarios, this difference in interpretation of the top surfaces gives larger differences in the averaged volumes than the standard deviation of among the different stochastic realizations. Finally, we notice that faults normal to the up-dip direction will increase the storage capacity, in particular for the FMM cases, whereas adding a second fault system with a strike angle of 30° will open some of the fold traps and hence lead to slightly lower structural trapping capacity.

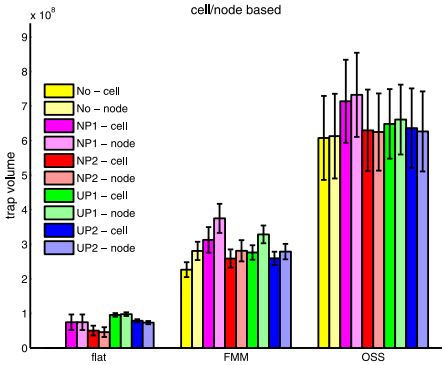


Fig. 13. Average volume available for structural trapping for one hundred realizations of each of the fourteen different top-surface scenarios defined in the IGEMS project. The vertical lines show the standard deviation. The five different structural scenarios are shown in different color: 'No' refers to cases without faults, 'UP' refers to uniform fault displacement and length, 'NP' refers to random fault displacement and length, '1' refers to a single 90° strike, and '2' refers to two cases with 30° and 90° strike directions. Complete script: `trapsIGEMS.m`.

By visualizing how branches in the trapping tree gradually fills up, the percolation type analysis used above can give an idea of the dynamics of a specific injection scenario, as illustrated in Fig. 14.

3.4. Finding optimal injection points: the Utsira Formation

In the final example, we demonstrate how information about structural traps and their connections can be used to guide

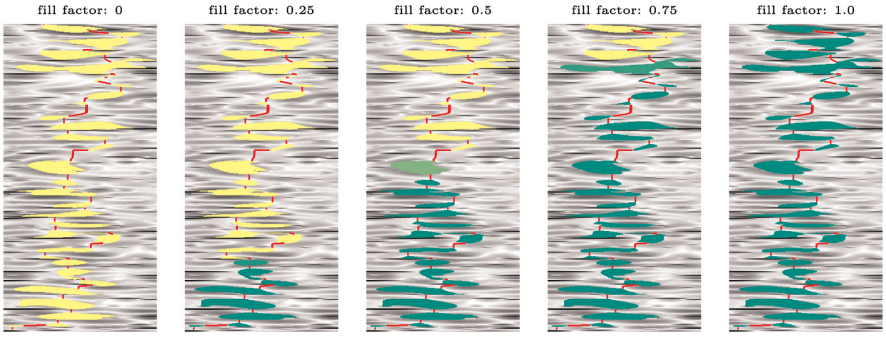


Fig. 14. Gradual filling of the main branch of the trapping tree for the scenario with offshore sand ridges and nonuniform faults with a 90° strike direction ('OSSNP1'). Complete script: `fillTreeIGEMS.m`.

injector placement. We consider the problem of determining where n_w wells should be placed in order to optimize the potential for structural trapping.

Assuming that the injection of CO_2 will take place at an infinitesimal rate, candidates for optimal injection points can be identified by considering the leaf nodes of the trees in the trapping structure. By identifying the n_w leaf traps that spill into the largest combined part of the trap network, and positioning the wells to spill into these, an optimal configuration is determined. The following "greedy" algorithm can be used:

1. Identify all leaf traps.
2. For each leaf trap, identify the set of upslope traps and compute the combined volume (including that of the leaf trap itself).
3. Find the leaf m with the largest combined volume, and add it to the list of optimal leaf traps.
4. Set the volume of traps upslope of m to zero.
5. Repeat from Step 2 until either n_w optimal leaf traps have been determined or there are no more trees with a non-zero volume.

This strategy ensures well positions that maximize the potential for structural trapping under the assumption of infinitesimal flow. However, in reality the CO_2 will be injected at a finite rate and form a volumetric plume with nonzero thickness and finite extent. One might therefore also consider locating the injection points on the ridges *between* trap regions corresponding to different trees (or distinctly different branches) with similarly sized upslope trapping volume. Injecting at these points would allow

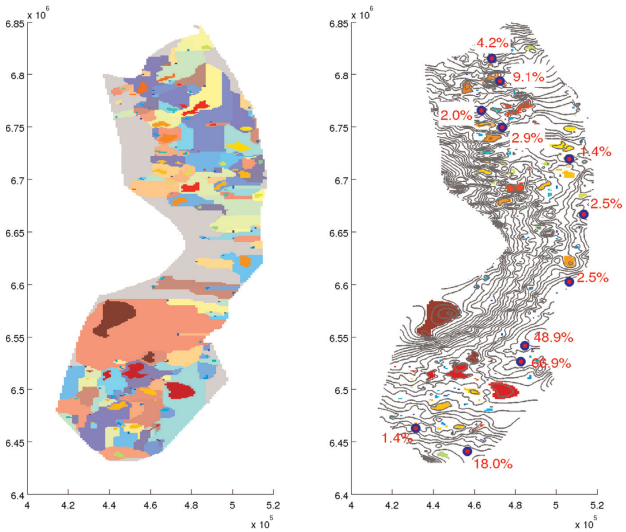


Fig. 15. Identification of good injection points in the Utsira Formation. *Left:* All traps and their associated catchment areas. *Right:* Traps colored by the total volume of the tree they belong to, eleven injection points, and the fraction of the total trap volume that can be reach by migration from each injection point.

CO_2 to migrate simultaneously into different trees (or branches). Sometimes, filling two medium-size trees could be a better injection strategy than filling a single large tree, both in terms of maximizing the structurally trapped volume, and balancing pressure buildup.

In Fig. 15 we have applied the strategy above on the Utsira Formation model from Halland et al. (2011) to compute the ten largest trapping trees, along with eleven injection points representing the best locations on ridges between two trees.

For realistic scenarios, additional factors need to be considered. The injection point should not be located too close to a spill region leading out of the domain, to avoid significant leakage when injecting at a finite rate. Injectivity and local heterogeneity should also be considered. Determining optimal injection *rates* remains an important task. In a practical workflow, the strategy outlined above can be used as a preprocessing stage to efficiently identify good starting points which can subsequently be improved upon using more comprehensive methods.

4. Concluding remarks

Through our examples we have tried to illustrate the large uncertainties and the resulting differences in simulated outcomes that can be expected and should be properly accounted for in real models. The actual numbers presented for real aquifers are based on the very limited data that are publicly available and should therefore not be taken literally as reliable estimates of actual storage capacities. However, given sufficient data, we believe that using simple tools for computing structural traps, catchment areas, and spill-point paths will be instructive as a means to rapidly investigate effects of data resolution, guide the placement of injection wells, and explore large portions of the parameter space.

We have presented two classes of algorithms for computing structural trapping and estimates of migration paths. They are geometrically dual in the sense that one considers connections between cell centers and the other relies on connections between cell corners. Both algorithms are inspired by ideas for primary migration of oil and gas, or equivalently, by concepts in water management like drainage areas and waterway networks. They have to the best of our knowledge not been used in the setting of geological CO_2 storage before. In particular, we have outlined the algorithms used, demonstrated their sensitivity to geometry variations, and demonstrated how to identify injection points with the largest potential for structural trapping. In general, we believe that much of the information needed to optimize a CO_2 injection scenario can be obtained using simplified tools that honor the main dynamics in the system, gravity flow, as also noticed by Singh et al. (2010), who used percolation type of calculations. In Andersen et al. (2014), Lie et al. (2014) and Nilsen et al. (2014c), we demonstrate how the tools described herein can be combined with vertical-equilibrium models and rigorous mathematical optimization to develop scenarios for the injection of hundreds of megatonnes of CO_2 into saline aquifers in the North Sea.

All methods are implemented using the CO_2 module of the Matlab Reservoir Simulation Toolbox, and are freely available as open-source code (SINTEF ICT, 2014). All examples are based on open data and come with full MRST scripts.

Acknowledgments

The work was funded in part by Statoil ASA and the Research Council of Norway through Grants No. 199878 (Numerical CO_2 laboratory) and 21564 (MatMoRA-II).

References

Andersen, O., Nilsen, H.M., Lie, K.-A., 2014. Reexamining CO_2 storage capacity and utilization of the Utsira Formation. In: ECMOR XIV – 14th European Conference on the Mathematics of Oil Recovery. Catania, Sicily, Italy;2014. pp. 8-11 September 2014, EAGE. doi:http://dx.doi.org/10.3997/2214-4609.20141809.

Benson, S.M., et al., 2005. Underground geological storage. In: IPCC Special Report on Carbon Dioxide Capture and Storage, Intergovernmental Panel on Climate Change. Cambridge University Press, Cambridge, U.K.; 2005, Ch. 5.

Class, H., Ebigbo, A., Helmig, R., Dahle, H.K., Nordbotten, J.M., Celia, M.A., Audigane, P., Darcis, M., Ennis-King, J., Fan, Y., Flemisch, B., Gasda, S.E., Jin, M., Krug, S., Labregere, D., Beni, A.N., Pawar, R.J., Sbai, A., Thomas, S.G., Trenty, L., Wei, L., 2009. A benchmark study on problems related to CO2 storage in geologic formations. Comput. Geosci. 13 (4), 409–434. http://dx.doi.org/10.1007/s10596-009-9146-x.

Eigestad, G., Dahle, H., Hellevang, B., Johansen, W., Lie, K.-A., Riis, F., Øian, E., 2008. Geological and fluid data for modelling CO2 injection in the Johansen formation (2008). URL ⟨http://www.sintef.no/Projectweb/MatMorA/Downloads/Johansen⟩.

Eigestad, G., Dahle, H., Hellevang, B., Riis, F., Johansen, W., Øian, E., 2009. Geological modeling and simulation of CO2 injection in the Johansen formation. Comput. Geosci. 13 (4), 435–450. http://dx.doi.org/10.1007/s10596-009-9153-y.

Halland, E.K., Johansen, W.T., Riis, F., (Eds.), CO_2 Storage Atlas: Norwegian North Sea, Norwegian Petroleum Directorate, P. O. Box 600, NO–4003 Stavanger, Norway, 2011. URL ⟨http://www.npd.no/no/Publikasjoner/Rapporter/CO2-lagringsatlas/⟩.

Lie, K.-A., Krogstad, S., Ligaarden, I.S., Natvig, J.R., Nilsen, H.M., Skaflestad, B., 2012. Open source MATLAB implementation of consistent discretisations on complex grids. Comput. Geosci. 16, 297–322. http://dx.doi.org/10.1007/s10596-011-9244-4.

Lie, K.-A., Nilsen, H.M., Andersen, O., Møyner, O., 2014. A simulation workflow for large-scale CO_2 storage in the Norwegian North Sea. In: ECMOR XIV – 14th European Conference on the Mathematics of Oil Recovery. Catania, Sicily, Italy; September 2014, EAGE, 2014, pp. 8–11. doi: http://dx.doi.org/10.3997/2214-4609.20141877.

Nilsen, H.M., Syversveen, A.R., Lie, K.-A., Tveranger, J., Nordbotten, J.M., 2012. Impact of top-surface morphology on CO2 storage capacity. Int. J. Greenh. Gas Control 11 (0), 221–235. http://dx.doi.org/10.1016/j.ijggc.2012.08.012.

Nilsen, H.M., Lie, K.-A., Andersen, O., Robust simulation of sharp-interface models for fast estimation of CO_2 trapping capacity, 2014a, submitted for publication. http://folk.uio.no/kalie/papers/co2lab-2.pdf.

Nilsen, H.M., Lie, K.-A., Andersen, O., Fully implicit simulation of vertical-equilibrium models with hysteresis and capillary fringe, 2014b, submitted for publication. http://folk.uio.no/kalie/papers/co2lab-3.pdf.

Nilsen, H.M., Lie, K.-A., Andersen, O., Analysis of CO_2 trapping capacities and long-term migration for geological formations in the Norwegian North Sea using `+mrst-co2lab+`, 2014c (submitted for publication). http://folk.uio.no/kalie/papers/co2lab-4.pdf.

Norwegian Computing Center, Impact of realistic geologic models on simulation of CO_2 storage, 2013. URL ⟨http://www.nr.no/IGEMS/⟩.

Singh, V., Cavanagh, A., Hansen, H., Nazarian, B., Iding, M., Ringrose, P., 2010. Reservoir modeling of CO_2 plume behavior calibrated against monitoring data from Sleipner, Norway. In: SPE Annual Technical Conference and Exhibition, 19–22 September 2010, Florence, Italy, 2010, SPE 134891-MS. doi: http://dx.doi.org/10.2118/134891-MS.

SINTEF ICT, The MATLAB Reservoir Simulation Toolbox: Numerical CO_2 laboratory (October 2014). URL ⟨http://www.sintef.no/co2lab⟩.

Syversveen, A.R., Nilsen, H.M., Lie, K.-A., Tveranger, J., Abrahamsen, P., 2012. A study on how top-surface morphology influences the storage capacity of CO_2 in saline aquifers. In: Abrahamsen, P., Hauge, R., Kolbjørnsen, O. (Eds.), Geostatistics Oslo 2012, vol. 17, Quantitative Geology and Geostatistics. Springer Netherlands, 2012, pp. 481–492. doi: http://dx.doi.org/10.1007/978-94-007-4153-9_39.

The MATLAB Reservoir Simulation Toolbox, version 2014a, ⟨http://www.sintef.no/MRST/⟩(May 2014).

Wei, L., Saaf, F., 2009. Estimate CO2 storage capacity of the Johansen formation: numerical investigations beyond the benchmarking exercise. Comput. Geosci. 13 (4), 451–467. http://dx.doi.org/10.1007/s10596-008-9122-x.

Paper III

III

Robust Simulation of Sharp-Interface Models for Fast Estimation of CO_2 Trapping Capacity in Large-Scale Aquifer Systems

Halvor M. Nilsen, Knut-Andreas Lie, Odd A. Andersen
Computational Geosciences, **20**, (2016)

Comput Geosci (2016) 20:93–113
DOI 10.1007/s10596-015-9549-9

ORIGINAL PAPER

Robust simulation of sharp-interface models for fast estimation of CO_2 trapping capacity in large-scale aquifer systems

Halvor Møll Nilsen[1] · Knut-Andreas Lie[1] · Odd Andersen[1,2]

Received: 1 October 2014 / Accepted: 20 November 2015 / Published online: 17 December 2015

Abstract Modeling geological carbon storage represents a new and substantial challenge for the subsurface geosciences. To increase understanding and make good engineering decisions, containment processes and large-scale storage operations must be simulated in a thousand-year perspective. Large differences in spatial and temporal scales make it prohibitively expensive to compute the fate of injected CO_2 using traditional 3D simulators. Instead, accurate forecast can be computed using simplified models that are adapted to the specific setting of the bouyancy-driven migration of the light fluid phase. This paper presents a family of vertically integrated models for studying the combined large-scale and long-term effects of structural, residual, and solubility trapping of CO_2. The models are based on an assumption of a sharp interface separating CO_2 and brine and can provide a detailed inventory of the injected CO_2 volumes over periods of thousands of years within reasonable computational time. To be compatible with simulation tools used in industry, the models are formulated in a black-oil framework. The models are implemented in `MRST-co2lab`, which is an open community software developed especially to study and optimize large-scale, long-term geological storage of CO_2. The resulting simulators are fully implicit and handle input from standard geomodeling tools.

Keywords CO2 storage · Vertical equilibrium · Compressibility · Dissolution

1 Introduction

Dipping saline aquifers comprise the largest volumes available for large-scale storage of CO_2. To accurately estimate the storage capacity of an open aquifer, one must determine the maximum amount of CO_2 that can be injected and how far and how fast the injected plume will migrate throughout the aquifer. The injection and migration processes are governed by delicate balances of various physical mechanisms (see Fig. 1) which may change with time and spatial location. Resolving these processes is a challenging multiscale problem that involves a large disparity in spatial and temporal scales. CO_2 is very mobile and can travel large distances, but the flow is usually confined to thin layers underneath a sealing caprock or other low-permeable vertical barriers. A typical saline aquifer considered for CO_2 storage can be viewed as a thin, slightly inclined sheet that spans thousands of square kilometers. This, in combination with differences in density between the supercritical CO_2 plume and the resident brine, means that the vertical fluid segregation will be almost instantaneous compared with the up-dip migration. The tendency of forming a relative flat fluid interface is an effect of the pressure distribution, which in turn depends strongly on the flow in the vertical direction, particularly near the interface. The vertical fluid distribution must also be accurately represented to avoid introducing large errors

✉ Odd Andersen
odd.andersen@sintef.no

Halvor Møll Nilsen
Halvor.M.Nilsen@sintef.no

Knut-Andreas Lie
Knut-Andreas.Lie@sintef.no

[1] SINTEF ICT, Applied Mathematics, P.O. Box 124 Blindern, 0314 Oslo, Norway

[2] Department of Mathematics, University of Bergen, P.O. Box 7803, 5020 Bergen, Norway

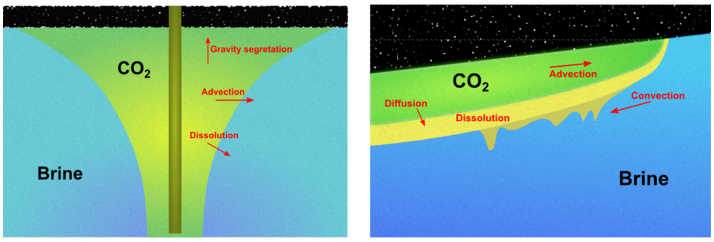


Fig. 1 Illustration of injection and trapping of CO_2 under a sealing caprock. The plots show forces active during the injection (*left*) and migration phase (*right*)

in the forecast of the updip migration. (For nonlinear relative permeabilities, a large error will result from replacing the average of the mobility by the mobility of the average phase distribution). The thin plume and sharp transition between CO_2 and brine means that high vertical resolution is required to compute the vertical phase distribution. High grid resolution is also needed in the lateral direction to resolve the unstable dynamics of convective dissolution, which retards and limits the plume migration and to account for small undulations in the caprock topography, that may both retard the migration of the plume and divert its path. In other words, simulating likely outcomes over a period of thousands of years is in most cases computationally intractable with conventional 3D reservoir simulators.

This is clearly demonstrated in the recent Stuttgart benchmark [1], in which a variety of commercial simulators and research codes were used by leading academic and industry experts to solve three model problems. Despite a large computational effort and significant work spent in eliminating differences in input data, the reported results are striking in their disparity, particularly for the study of how a supercritical CO_2 plume forms and migrates upward in a small region of the Johansen formation [2, 3] from the Norwegian North Sea. At the end of simulation, the reported results show almost no consensus in the simulated shapes of the CO_2 plume.

In our opinion, there is an urgent need to advance state-of-the-art in numerical simulation to improve the assessment of storage capacity, facilitate approval of specific storage projects, and help realize the storage potential both globally and locally on the Norwegian Continental Shelf. In terms of modeling, efforts are needed along several parallel paths. First, one must continue to develop reliable numerical methods that model fundamental flow physics accurately and correctly. Second, these methods must be combined in robust numerical formulations that account for varying degrees of coupling between the different physical effects that drive or influence CO_2 migration. These numerical formulations need to be sufficiently flexible so that one in a simple and case-dependent manner can combine the flow equations with thermal, geomechanical, and geochemical effects. Finally, the formulations must be implemented as trustworthy software that enables transparent comparisons of models, methods, and simulated outcomes.

We believe that the only way these challenges can be tackled is through collaboration, development, and extensive use of community software. To contribute to this and to accelerate transfer of knowledge developed in academic research projects to end-users in industry and the public sector, we have started developing a community software called `MRST-co2lab` [4–8] implemented on top of the open-source Matlab Reservoir Simulation Toolbox (MRST) [9–12]. Both MRST and its CO_2 module can be freely downloaded and used under the GNU General Public License v3.0. The software offers reliable modeling of realistic storage scenarios, enables interactive experimentation with various model assumptions like boundary conditions, fluid models and parameters, injection points and rates, amount of subscale trapping, and so on, and simplifies the development, implementation, and comparison of new models and computational methods. The software offers a hierarchy of models and tools of increasing computational complexity [7, 8], as well as a set of tutorials and examples that demonstrate and highlight how these tools can be applied to study fundamental flow physics as well as descriptions of realistic storage scenarios based on public data sets of the Johansen formation [2, 3], the Sleipner injetion [13], and saline aquifers from the Norwegian Continental Shelf [14].

Herein, we present a family of vertically-integrated models for studying the combined large-scale and long-term effects of structural, residual, and solubility trapping to provide detailed inventories of injected CO_2 volumes over periods of thousands of years. All models are based on the assumption of vertical equilibrium (VE) with a sharp interface that separates the injected CO_2 from the resident brine. Apart from the open-source implementation, the novelty of our work lies in a flexible and robust formulation that unifies work from the early period of reservoir simulation [15–18], when practical numerical aspects were primarily in focus, with recent extensions of the VE framework [19] that focus more on physical effects related to

large-scale CO_2 injection. The validity of the simplifying assumptions underlying VE models has been studied both with respect to spatial [20] and temporal [21] scales, and the utility of VE models is thoroughly discussed in, e.g., [22, 23]. Early studies focused on VE models with a sharp-interface assumption [24–26], and models that only account for the basic effects of buoyant migration were successfully used to simulate long-term migration in the Utsira [27] and Johansen [28] aquifers. Later, the class of VE models has been extended to incorporate most of the flow effects that are pertinent to large-scale migration, including compressibility [29], convective dissolution [30, 31], capillary fringe [32], small-scale caprock topography variations [33–35], various hysteretic effects [36–38], multiple geological layers [39, 40], and heat transfer [41]. In particular, several studies show that vertical equilibrium simulations compare well with 3D simulators on case studies of the Johansen aquifer [42] and the 9^{th} layer of the Sleipner injection [27, 43]. The assumption of vertical equilibrium not only reduces the number of spatial dimensions, and hence the number of grid cells, but will also reduce the coupling between pressure and fluid transport and improve the characteristic time constants of the problem [42]. As a results, VE simulations will typically be orders of magnitude faster and consume significantly less memory than conventional 3D simulators. In [23, 43], the authors report a simulation of CO_2 migration under the caprock at Sleipner, for which a VE simulator running for a few minutes on a single core produced forecasts with similar accuracy as a 3D simulation with TOUGH2 running for several hours on one hundred cores.

To develop our numerical framework, we first discuss modeling of various physical mechanisms including compressibility and retardation effects from subscale trapping. We then show how a general class of VE models can be recast as standard black-oil models using the traditional concept of pseudo-functions [44–47] and discuss the inclusion of dynamic dissolution effects. In particular, we point out in detail the approximations and numerical considerations needed to obtain flexible and efficient numerical formulations that resolve the main physical effect well within the accuracy normally available from input data. The overall formulation is implemented as an extension of existing black-oil solvers in MRST [12], which in turn have been implemented using automatic differentiation and hence enable simple computation of gradients and parameter sensitivities, e.g., through an adjoint formulation. This enables the users to easily perform sensitivity studies or formulate efficient strategies for rigorous mathematical optimization of large-scale injection strategies [6, 7, 48, 49]. In [50], we discuss how the framework can be extended to account for smooth transitions (capillary fringe) between pure CO_2 and brine. We also show how inclusion of more advanced flow physics naturally leads to pressure-dependence and hysteretic effects in the vertically-integrated relative permeabilities and capillary pressure and suggest implementation choices we think are important to make flexible and efficient VE simulators.

Whereas the focus of the present paper is on reduced models, the functionality provided by `MRST-co2lab` also includes other computational tools for analysis of CO_2 storage. In [5], we discuss tools for fast and interactive estimation of structural trapping and potential migration paths under the assumption that CO_2 is injected at an infinitesimal rate. These tools do not account for temporal aspects, but will nevertheless reveal important information such as accessible structural traps, spill points and migration paths, good injection locations, points where the CO_2 may leak out through open boundaries, etc. In [6–8], we outline how the various tools can be combined to create a flexible tool chain for estimating storage capacities and studying injection scenarios. In particular, we study and optimize strategies for injecting hundreds of megatonnes of CO_2 into various saline aquifers in the Norwegian North Sea using data sets from the recent CO_2 Storage Atlas [14]. Finally, we mention that `MRST-co2lab` also contains conventional 3D simulation capabilities and that work is in progress to include additional physics such as thermal, geochemical, and geomechanical effects.

Complete MATLAB scripts containing all the statements necessary to reproduce the figures presented in this paper can be downloaded as part of the 2015b release of `MRST-co2lab`.

2 Trapping mechanisms

With a sharp-interface VE model, it is simple to make accurate inventories of carbon trapping for specific scenarios, detailing how injected CO_2 volumes are separated into parts that can be considered safely contained and parts that may potentially leak back to the surface. The general trend is that CO_2 becomes more securely trapped with time as a result of trapping processes taking place at different rates that vary from days to years to thousands of years.

When CO_2 is injected into a deep subsurface rock formation, it forms a separate mobile, typically dense phase (the CO_2 plume) that invades the medium and displaces other liquids present in the pore space (typically: brine). The CO_2 phase is almost always less dense than the resident fluids and will therefore rise upwards and hence be replaced by other fluids. However, as the volume fraction of the CO_2 phase falls below a certain level, CO_2 becomes trapped in the pore space between rock grains by capillary pressure from the other fluids and stops flowing. This is referred to as *residual trapping* and the corresponding volumes of CO_2 are denoted *'residual'* in the inventory. At any point,

the inventory of the plume is therefore subdivided into two categories: *'residual (plume)'* refers to the fraction of the CO_2 column that will eventually stay behind in a residually trapped state when the plume leaves its current position, whereas *'movable (plume)'* is the remaining part that is free to migrate away from the current position.

In most relevant scenarios, CO_2 is injected under a sealing rock in which the capillary pressure inside pore throats is greater than the buoyancy pressure of the CO_2. The top seal will prevent the direct upward movement of the plume and if the seal is sloping, the CO_2 will form a thin layer underneath that slowly migrates in the upslope direction until it encounters a structural trap, i.e., a fold in the top surface inside which the CO_2 will accumulate. Once inside a trap, the CO_2 will remain *structurally trapped* unless the height of the plume creates a capillary pressure that enables the CO_2 to enter the seal. The structurally trapped volumes are therefore divided into two similar categories: *'residual (traps)'* will remain immobile and never leak, while *'movable (traps)'* could in principle escape through a crack in the top seal. Once a trap is filled, the CO_2 will spill over and continue migrating upward. CO_2 can also be trapped in *stratigraphic traps* because of changes in rock type within the storage layer, but this mechanisms is not represented in VE models.

The remaining trapping mechanisms are *solubility and mineral trapping*. Over time, CO_2 will dissolve in the resident fluids. Brine containing dissolved CO_2 is slightly denser than the surrounding fluids and will sink to the bottom of the rock formation, thereby trapping CO_2 more securely. This enables a mixing process that increasingly disperses CO_2 into brine over time. The dissolved CO_2 forms a weak carbonic acid that may react with the reservoir rock to form and precipitate carbonate minerals that bind CO_2 permanently to the rock. This process may be rapid or very slow and is not accounted for herein.

To summarize, the 'dissolved', 'residual (traps)', and 'residual' volumes are safely stored unless changes occur in the aquifer that alter the residual saturation of CO_2 or cause the CO_2 to effervesce from the formation water (think of the fizz when you open a bottle of carbonated water). The 'movable (traps)' volumes are immobilized and will be safely stored unless the structural traps contain leakage points. The remaining volumes will continue to migrate in the upslope direction, the 'residual (plume)' volumes will eventually remain as residually trapped and only the 'movable (plume)' volumes may leak if not trapped by another trapping mechanism at a later time. If the aquifer model has open boundaries, some of the injected CO_2 may also leave the computational domain during the simulation and these volumes will, in lack of a better word, be referred to as *'leaked'*. We emphasize that this does not mean that the corresponding CO_2 has leaked back to the atmosphere; in most cases, it will continue to migrate inside another rock volume that is outside the simulation model. In the inventory in Fig. 2, the various categories of CO_2 volumes have been stacked according to increasing risk of leakage, from dissolved CO_2 (dark green color) to volumes that are still movable (yellow/orange) or have already left the simulated domain (red).

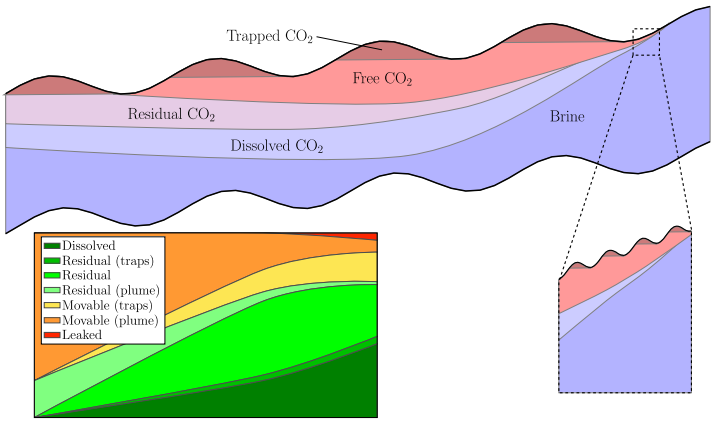


Fig. 2 Schematic of a vertical section of an aquifer showing the different forms in which the injected CO_2 can be present during migration. The *right inlet* shows small-scale undulations in the top surface which typically will not be resolved in a large-scale model. The *left inlet* shows a detailed inventory of various categories of trapped CO_2, stacked in terms of increasing leakage risk, as function of time from the end of injection

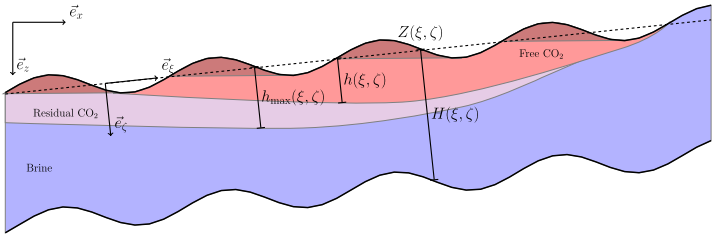


Fig. 3 Schematic of the fluid distribution and the coordinate systems used to derive the basic vertical-equilibrium model. Here, $h(\xi,\eta)$ is the interface between CO_2 and brine, and $h_{\max}$ represents the historical maximum value of h for a given location. The *dashed line* indicates assumed mean direction of flow within the aquifer (which is here illustrated as *straight*, but is allowed to be *slightly curved* in the general case)

3 Derivation of the basic VE model

Let brine and CO_2 be wetting and non-wetting fluids, respectively, and assume incompressible rock and fluids, no capillary forces, and impermeable top and bottom of the aquifer. Then, mass conservation and Darcy's law read:

$$\frac{\partial}{\partial t}(\phi s_\alpha) + \nabla\cdot\vec{v}_\alpha = q_\alpha, \tag{1}$$

$$\vec{v}_\alpha = -\mathbf{k}\lambda_\alpha\big(\nabla p - \rho_\alpha\vec{g}\big). \tag{2}$$

Here, ϕ denotes porosity, $\mathbf{k}$ permeability, p pressure, and $\vec{g}$ the gravity vector, whereas s_α, λ_α, and q_α denote saturation, mobility, and volumetric source for phase $\alpha = \{w, n\}$.

We introduce a curvilinear coordinate system $(\vec{e}_\xi, \vec{e}_\eta, \vec{e}_\zeta)$ whose orientation is defined to align closely with the global system $(\vec{e}_x, \vec{e}_y, \vec{e}_z)$, but be slightly tilted so that $\vec{e}_\zeta$ locally is perpendicular to the main flow direction (disregarding small and medium-scale oscillations). The tilt only depends on ξ and η, is assumed to vary smoothly and at all points remain modest (a few degrees). We further write $\vec{g} = \vec{g}_\parallel + g_\zeta\vec{e}_\zeta$ and $\nabla = \nabla_\parallel + \vec{e}_\zeta\partial_\zeta$, where $\parallel$ refers to the $(\vec{e}_\xi, \vec{e}_\eta)$ components of a vector/operator. In the new coordinate system, the aquifer is described by the top surface $Z(\xi,\eta)$ and its thickness $H(\xi,\eta)$ in the ζ-direction, see Fig. 3. Capital letters are used for quantities in the upscaled model.

Integrating (1) from top to bottom of the aquifer, neglecting distortions from the curvilinear nature of the coordinate system,[1] we obtain:

$$\frac{\partial}{\partial t}\Big[\int_Z^{Z+H} S_\alpha\phi\,d\zeta\Big] + \nabla_\parallel\cdot\Big[\int_Z^{Z+H}\vec{v}_\alpha\,d\zeta\Big] = \int_Z^{Z+H} q_\alpha\,d\zeta. \tag{3}$$

By design of the coordinate system, the flow along $\vec{e}_\zeta$ will take place very rapidly compared with the migration in the $(\vec{e}_\xi, \vec{e}_\eta)$ direction. Hence, we assume hydrostatic equilibrium in the $\vec{e}_\zeta$ direction. Since capillary pressure is neglected, CO_2 and brine will at equilibrium be separated by a sharp interface located a distance $h(\xi,\eta)$ from the caprock along $\vec{e}_\zeta$. Setting the pressure datum P_Z at the top surface, the pressure at a given depth ζ is determined by

$$p(\zeta) = \begin{cases} P_Z + \rho_{\mathrm{n}} g_\zeta(\zeta - Z), & Z \le \zeta \le Z + h, \\ P_Z + \rho_{\mathrm{n}} g_\zeta h + \rho_w g_\zeta(\zeta - Z - h), & Z + h \le \zeta \le Z + H. \end{cases} \tag{4}$$

Figure 3 shows how each vertical column is divided into three regions:

- The CO_2 plume between Z and $Z + h$ with residual brine saturation $s_{w,r}$, CO_2 saturation $1 - s_{w,r}$, and CO_2 mobility $\lambda_{\mathrm{n},e} = \lambda_{\mathrm{n}}(1 - s_{w,r})$.
- The residual region between h and $h_{\max}$ with CO_2 saturation $s_{\mathrm{n},r}$, brine saturation $1 - s_{\mathrm{n},r}$, and brine mobility $\lambda_{w,e} = \lambda_w(1 - s_{\mathrm{n},r})$.
- The region below $h_{\max}$ filled by brine.

We hence define vertically-averaged quantities:

$$\mathbf{K} = \frac{1}{H}\int_Z^{Z+H}\mathbf{k}_\parallel\,d\zeta, \tag{5}$$

$$\Lambda_{\mathrm{n}}(h) = \frac{1}{H}\Big[\int_Z^{Z+h}\lambda_{\mathrm{n},e}\mathbf{k}_\parallel\,d\zeta\Big]\mathbf{K}^{-1} \tag{6}$$

$$\Lambda_w(h, h_{\max}) = \frac{1}{H}\Big[\int_{Z+h}^{Z+h_{\max}}\lambda_{w,e}\mathbf{k}_\parallel\,d\zeta + \int_{Z+h_{\max}}^{Z+H}\lambda_w(1)\mathbf{k}_\parallel\,d\zeta\Big]\mathbf{K}^{-1} \tag{7}$$

Combining these expressions with Darcy's law (2) and setting $\Delta\rho = \rho_w - \rho_{\mathrm{n}}$, we obtain vertically-integrated fluxes:

$$\vec{V}_{\mathrm{n}} = -H\Lambda_{\mathrm{n}}\mathbf{K}\big[\nabla_\parallel(P_Z - \rho_{\mathrm{n}} g_\zeta Z) - \rho_{\mathrm{n}}\vec{g}_\parallel\big], \tag{8}$$

$$\vec{V}_w = -H\Lambda_w\mathbf{K}\big[\nabla_\parallel(P_Z - \rho_w g_\zeta Z) - g_\zeta\Delta\rho\nabla_\parallel h - \rho_w\vec{g}_\parallel\big]. \tag{9}$$

To develop the usual fractional-flow formulation, we introduce total velocity $\vec{V} = \vec{V}_{\mathrm{n}} + \vec{V}_w$, assume that $\mathbf{K}$ and

[1] A detailed explanation of the approximations related to the curvilinear system is provided in Appendix A.

Λ_α commute (**K** isotropic or $\mathbf{k}_\parallel$ constant in ζ), and sum (3) over phases. After some manipulations, we obtain a pressure equation:

$$\begin{aligned}\nabla_\parallel \cdot \vec{V} &= Q,\\ \vec{V} &= -H\Lambda\mathbf{K}\big[\nabla_\parallel P_z\\ &\quad -(\rho_{\mathrm{n}}\mathbf{F}_{\mathrm{n}}+\rho_w\mathbf{F}_w)(\vec{g}_\parallel+g_\zeta\nabla_\parallel Z)-\mathbf{F}_w g_\zeta\Delta\rho\nabla_\parallel h\big],\end{aligned} \tag{10}$$

where $\Lambda(h,h_{\max}) = \Lambda_w(h,h_{\max}) + \Lambda_{\mathrm{n}}(h)$ and $\mathbf{F}_\alpha(h,h_{\max}) = \Lambda_\alpha\Lambda^{-1}$, and a transport equation:

$$\frac{\partial}{\partial t}\Phi+\nabla_\parallel\Big[\mathbf{F}_{\mathrm{n}}\vec{V}+\Delta\rho\mathbf{K}\Lambda_w\mathbf{F}_{\mathrm{n}}\big[\vec{g}_\parallel+g_\zeta\nabla_\parallel(Z+h)\big]\Big]=Q_{\mathrm{n}}, \tag{11}$$

where $\Phi(h,\xi,\eta) = \int_Z^{Z+h}\phi(1-s_{w,r})\,d\zeta$ and $Q_{\mathrm{n}} = \int_Z^{Z+h} q_{\mathrm{n}}\,d\zeta$. This is the so-called h-formulation of the VE model. If ϕ and **K** are constant in the ζ-direction, we get the following upscaled porosity and relative mobilities:

$$\begin{aligned}\Phi(h) &= \phi(1-s_{w,r})h+\phi s_{n,r}(h_{\max}-h),\\ \Lambda_{\mathrm{n}}(h) &= h\lambda_{\mathrm{n},e},\\ \Lambda_w(h,h_{\max}) &= (h_{\max}-h)\lambda_{w,e}+(H-h_{\max})\lambda_w(1).\end{aligned}$$

To simplify implementation and avoid having to treat the curvilinear grid explicitly, we express the transport equation in terms of global coordinates by writing $\vec{g}_\parallel + g_\zeta\nabla_\parallel(Z+h) = |g|\nabla_\parallel z_{\mathrm{n}}$, where z_{n} is the true vertical position of the CO_2–brine interface, and approximating $\nabla_\parallel$ by ∇_{xy} to obtain:

$$\frac{\partial}{\partial t}\Phi(h,x,y)+\nabla_{xy}\Big[\mathbf{F}_{\mathrm{n}}\vec{V}+\Delta\rho\mathbf{K}\Lambda_w\mathbf{F}_{\mathrm{n}}\big[|g|\nabla_{xy}(Z+h)\big]\Big]=Q_{\mathrm{n}}. \tag{12}$$

As an additional approximation, we consider h to be measured along $\vec{e}_z$. Errors associated with the transformation of the grid and the transport equation are discussed in more detail in Appendix A.

In the absence of viscous forces, the stationary state of the system is given by the balance between gravity and the 'capillary pressure' $\nabla_{xy}h$. Writing the transport equation in physical coordinates honors explicitly the condition from the underlying 3D model that a horizontal interface is a stationary state.

Example 1 Consider a simple 1D model with a sealing caprock given by

$$z_t = D - L_1\sin\Big(\frac{x}{L_1}\Big)\tan\theta + A\sin\Big(\frac{2\pi x}{L_2}\Big), \tag{13}$$

where D is maximum depth, $\theta = 0.03$ is the initial tilt angel, $L_1 = 20$ km is the characteristic length of the antiform structure, and $L_2 = 0.3$ km is the wave-length L_2 of the small-scale structures; see Fig. 4. For the amplitude, we use either $A = 0$ or $A = 2$ m. The density of CO_2 is assumed to follow a model taken from [51] with a temperature variation given by

$$T = z\mathrm{K_T} + \mathrm{T_s}, \qquad K_T = 30\mathrm{K/km}, \quad T_s = 286\mathrm{K}. \tag{14}$$

Given a hydrostatic pressure computed from a constant brine density of 1100 kg/m^3, there will be different regimes of density variation depending on the depth of the aquifer as shown in Fig. 4. Here, we set $D = 2300$ m, at which the variation in density is relatively small. (The temperature varies from 335 to 357 K). In all calculations, we use isotropic permeability of 1 Darcy and constant porosity of 0.2. The brine and rock volumes are considered to be linear functions of pressure, with respective coefficients $4.3\cdot10^{-5}$ bar^{-1} and $1.0\cdot10^{-5}$ bar^{-1}, based on a reference pressure of 100 bar. The storage scenario consists of 50 years of injection at an annual rate of $760\cdot10^6$ kg/year. The model is discretized with 1000 uniform grid cells, and we use uniform time steps of 2 years during the injection period and 20 years during the subsequent 2000 year migration period.

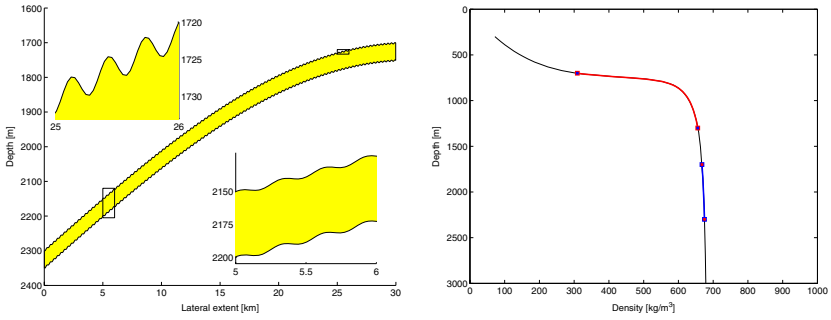


Fig. 4 *Left:* the geometry of Example 1. *Right:* the density of CO_2 given hydrostatic pressure; the *blue line* represents a deep model with $D = 2300$ m and the *red line* corresponds to a model with $D = 1300$. (From `showAquiferModel.m` and `showDensityVariation.m`)

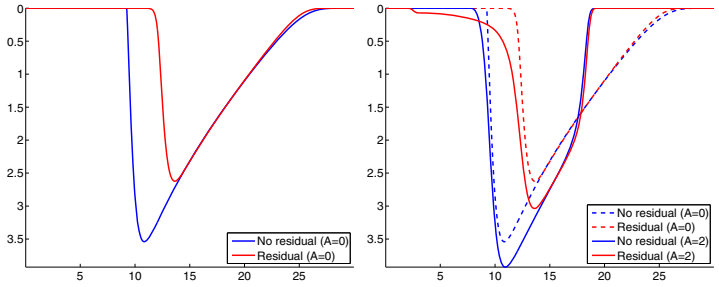


Fig. 5 Thickness of the CO_2 plume simulated by sharp-interface models for the 1D sloping aquifer in Example 1 after approximately 700 years. The *left plot* shows the effect of extending the basic model (*blue line*) with residual trapping (*red line*) for a surface without small-scale structures. In the *right plot*, these two outcomes (shown as *dashed lines*) are compared with calculations that also include small-scale undulations; in the plot, the solutions are averaged over a region that is larger than the small-scale undulations. (From `residualExample.m`)

In its simplest form, the VE model describes a plume that continues to migrate upward until it either reaches the top of the formation, or until all of the CO_2 has become trapped in geometric structures in the caprock. If nonzero residual saturation is introduced, the migrating plume will leave behind a trail of residually trapped CO_2. As a result, the tail of the plume will move faster than the tip, which means that CO_2 injected into an infinite aquifer without structural traps will only spread a finite distance, see [52]. The left plot of Fig. 5 compares plume thickness computed with and without residual trapping after 700 years.

The presence of small-scale variation (rugosity) in the top seal will generally lead to trapping of small amounts of CO_2, thereby retarding the migration of the plume [33]. This, in turn, changes the solution to also include a shock at the front of the plume, as seen in the left plot of Fig. 5 and in more detail for the case with no residual trapping in Fig. 6. In Section 4.1, we will develop effective relative-permeability models that capture the retardation effect caused by caprock rugosity.

The four simulations were performed using a general-purpose, fully-implicit black-oil solver from MRST [12]; more details will be given in the next section. Computational costs are reported in Table 1. With residual trapping and rugosity, the prescribed 20-year time step is straddling the stability limit and hence the nonlinear solver was forced to halve many of the time steps, which explains the significant increase in CPU time. On the other hand, the relatively high computational cost is a MATLAB artifact that poorly reflects the efficiency of the underlying algorithm: With only 1000 cells in the model, the computational overhead induced by the combination of MATLAB and automatic differentiation is significant. In these particular runs, only 3 % of the total CPU time was spent solving linear systems. This is a known issue with MRST, which is significantly diminished if one goes to models with more unknowns.

Finally, we compare the simulation outcome of the VE model with that of a full 3D model for the case with smooth caprock ($A = 0$). The 3D model has a vertical resolution of 30 cells, with progressively thinner cells towards

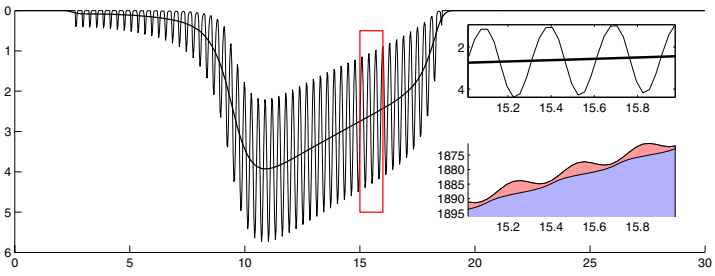


Fig. 6 Sharp-interface simulation of the 1D aquifer with small-scale undulations in the top surface. The thin oscillatory *black line* is the thickness of the CO_2 plume after approximately 700 years and the *thick line* is the average. The *upper inlet* shows a magnification of the *red square*, while the *lower inlet* shows the corresponding vertical fluid distribution in the real coordinate system. (From `residualExample.m`)

Table 1 Computational cost for the simulations in for Example 1 observed using MRST 2015a with MATLAB R2014a on an Intel Xeon 3.47 GHz CPU

	No residual	Residual
Smooth caprock	42 s	63 s
Caprock with rugosity	64 s	172 s

the top of the aquifer to better resolve the shape of the plume. The lateral resolution is 200 cells for both models. We use linear relative permeabilities and zero capillary pressure. As we can see from the left plot in Fig. 7, there is an almost perfect match between the models without residual saturation, except that plume in the 3D simulation has a characteristic staircase shape that will be present in all 3D simulations unless the width of the capillary fringe exceeds the height of the grid blocks. When residual saturation is included, the match between models still remains close. The required computational times to simulate 2000 years of migration using our laptop (Core i7-4500U processor, 8 GB RAM) for the VE simulations without/with residual saturations were 64 and 66 s, respectively. For the corresponding 3D simulations, the runtimes were 673 and 1005 s. We emphasize, however, that neither the VE nor the 3D simulation codes have been optimized for speed.

The simple aquifer geometry introduced above will be used repeatedly throughout the paper as a means to illustrate and discuss how including additional physical mechanisms affects the plume migration. In the next example, we consider a scenario with a touch of more geological realism.

Example 2 The Pliocenesand model from the CO_2 Storage Atlas [53] consist of 13,484 active cells and describes a ridge ending in a large plain. The actual sand body lies too shallow to be a candidate for CO_2 storage, but the model can be used as a test case if we increase its burial depth to, e.g., a thousand meters. The top surface has almost no fine-scale structure and thus allows for a very low percentage (0.02 %) of structural trapping compared to the overall volume of the whole sand body. To store CO_2, one should therefore primarily consider residual and solubility trapping. We set the pressure to 100 bar at the top point and assume a homogeneous porosity of 25 % and a homogeneous, isotropic permeability of 1200 mD. Ten mega-tonnes of CO_2 are injected annually for 50 years from a single injection point halfway down the slope. Brine has constant density 975.86 kg/m^3, viscosity 0.3086 cP, residual saturation equal 0.1, and an end-point mobility of 0.2142. The CO_2 phase has constant density 686.54 kg/m^3, viscosity 0.056641 cP, residual saturation 0.2, and an end-point mobility of 0.85. In the simulation, we use time steps of 2 years during injection and 10 years afterward.

The injected CO_2 plume moves upward towards the plain, leaving behind a trail of residually trapped CO_2. Figure 8 shows the CO_2 plume at the end of injection and after 100 years along with an instantaneous volumetric inventory. From the pie chart, we see that 6 % of the injected CO_2 has been residually trapped after 100 years, while the remaining volume is still inside the plume. However, a significant fraction of the plume volume (20 % of the injected volume) will eventually remain as residually trapped when the plume leaves its current position.

After 500–600 years, the tip of the plume has crossed the upper plain and reached the open boundary. The amount that leaks will roughly be proportional to the square of the plume thickness. Since the tip of the plume is very thin and the caprock is almost horizontal, the leakage is insignificant

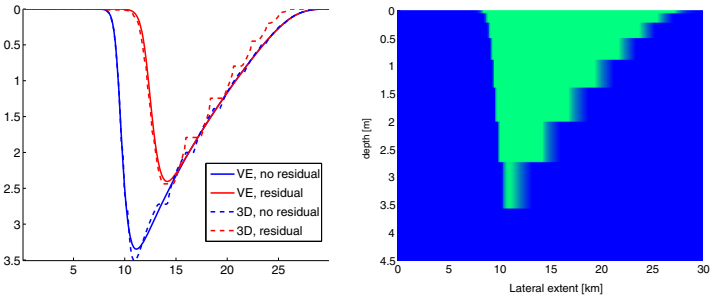


Fig. 7 Comparison between the VE models and corresponding full 3D models for the 1D sloping aquifer in Example 1 after approximatively 700 years. The *left plot* compares the plume thicknesses resulting from VE and 3D simulations with (*red*) and without (*blue*) residual trapping included. The *right plot* illustrates the upper 4.5 m of the aquifer 3D model (flattened, for presentational clarity) and the CO_2 saturation after approximatively 700 years for the case without residual trapping. The complete phase segregation is evident, and we also see how the vertical grid discretization is related to the *'staircase' shaped curves* shown on the *left plot*

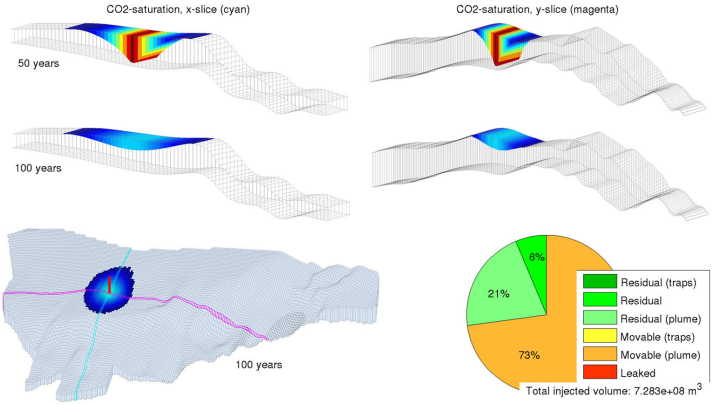


Fig. 8 Injection into the Pliocenesand aquifer from a single injection point. The plots show fluid distribution at the end of the injection and after 50 years of migration, as well as an inventory of the trapping observed after 100 years. (From `firstPlioExample.m`)

at first and only visually noticeable in the CO_2 inventory in Fig. 9 after another 200–300 years. At the end of simulation, only 3 % has left our computational domain, and the mobile CO_2 has reached a region with a very low sloping angle and is thus almost immobilized.

Figure 10 reports CPU time per step in a sequentially implicit simulator that utilizes a standard two-point pressure solver and a fully-implicit transport solver from the `incomp` module of MRST [10] and a fully-implicit black-oil solver based on automatic differentiation [12]. Both simulators have time-step control and will, if necessary, reduce the time steps to ensure convergence. During the injection period, the flow is dominated by the 'advective' $\mathbf{F}_n\vec{V}$ term in Eq. 12 resulting from heightened pressure in the near-well region, and the sequentially implicit solver is significantly more efficient, partially because of the significant overhead in the black-oil solver as discussed in Example 1. During the first 600 years after injection ceases, the migration is dominated by the hyperbolic $\nabla_{xy}Z$ term in Eq. 12, and the two simulators are equally efficient. After this point, the parabolic $\nabla_{xy}h$ term, which here plays the role of capillary forces in a conventional flow model, gradually becomes more important. The influence of the parabolic transport terms is more difficult to resolve in a sequential formulation and hence we see a gradual increase in the CPU time of each transport step. After 1260–1280 years, most of the mobile plume has reached the flat plane, and the plume is approaching a steady state. The dynamics of this state is governed mainly by the coupling of the gravitational $\nabla_{xy}Z$ and the parabolic $\nabla_{xy}h$ term and the effects of these terms in the pressure equation. These forces seek to enforce a flat interface between the CO_2 plume and the underlying brine. Operator-splitting methods, and in particular methods based on a total velocity formulation, are not suited for cases with such strong coupling between pressure and transport. In our case, the transport solver has to aggressively cut the time steps, which results in a dramatic increase in the CPU time. The fully-implicit formulation, on the other hand, is robust

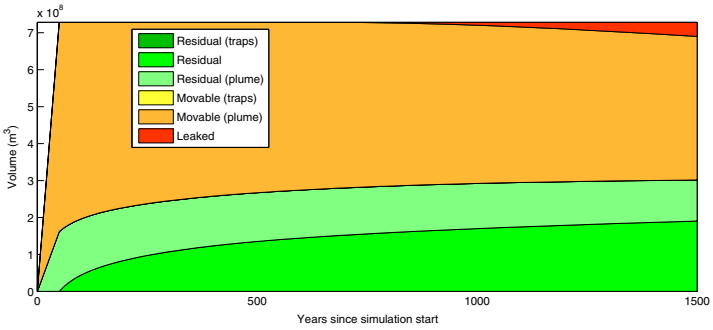


Fig. 9 Detailed inventory of the CO_2 trapping process in the Pliocenesand formation over a period of 1500 years. (From `firstPlioExample.m`)

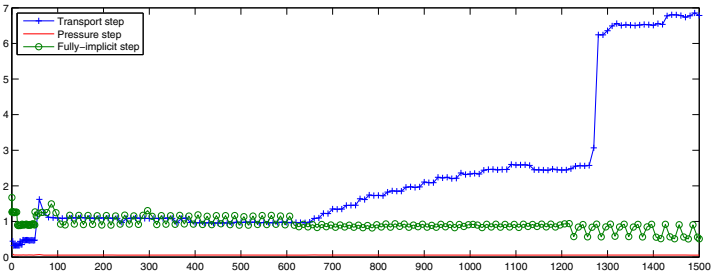


Fig. 10 CPU times in seconds for each step in a sequentially implicit simulation and in a fully-implicit, black-oil type simulation of the Pliocenesand injection measured for MRST 2015a with MATLAB R2014a on an Intel Xeon 3.47 GHz CPU

with respect to changing force balances and can, in particular, accurately resolve the near steady-state towards the end of the simulation.

As an alternative to using h as the primary unknown, one can use the fractional content $S = h/H$ of CO_2 in the vertical column to derive the so-called S-formulation having equations that look like standard pressure and transport equations known from the petroleum literature with S playing the role of saturation. In the rest of the section, we will detail the derivation for vertically homogeneous rock properties ϕ and $\mathbf{k}$. By integration over the column, it is easy to show that:

$$\begin{aligned} S(h, h_{\max}) &= \tfrac{h(1-s_{w,r})+(h_{\max}-h)s_{\mathrm{n},r}}{H}, \\ S_{\max}(h_{\max}) &= \tfrac{h_{\max}(1-s_{w,r})}{H}. \end{aligned} \tag{15}$$

where we recall that $s_{w,r}$ and $s_{\mathrm{n},r}$ are constants. Inversely, h can be written in terms of S and $S_{\max}$:

$$\begin{aligned} h(S, S_{\max}) &= H\,\tfrac{S(1-s_{w,r})-S_{\max}s_{\mathrm{n},r}}{(1-s_{w,r})(1-s_{w,r}-s_{\mathrm{n},r})}, \\ h_{\max}(S_{\max}) &= \tfrac{HS_{\max}}{1-s_{w,r}}. \end{aligned} \tag{16}$$

To obtain a set of standard transport equations formulated in S, we transform our coordinate system so that $Z \equiv 0$ and then write:

$$\begin{aligned} \Lambda_{\mathrm{n}}(S,\ S_{\max}) &= \tfrac{1}{H}\Big[\textstyle\int_0^h \lambda_{\mathrm{n},e}\mathbf{k}_{\|}\,d\zeta\Big]\mathbf{K}^{-1} = \tfrac{h(S,S_{\max})}{H}\lambda_{\mathrm{n},e} \\ \Lambda_{w}(S,\ S_{\max}) &= \tfrac{1}{H}\Big[\textstyle\int_h^{h_{\max}} \lambda_{w,e}\mathbf{k}_{\|}\,d\zeta + \int_{h_{\max}}^{H} \lambda_{w}(1)\mathbf{k}_{\|}\,d\zeta\Big]\mathbf{K}^{-1} \\ &= \tfrac{h_{\max}(S_{\max})-h(S,S_{\max})}{H}\lambda_{w,e} + \tfrac{H-h_{\max}(S_{\max})}{H}\lambda_{w}(1) \end{aligned}$$

where the last equality sign in each equation assumes vertically constant rock properties. This formulation is easy to implement in a standard reservoir simulator; all one has to do is to replace the evaluation of mobility and capillary pressure functions by special functions that compute vertically integrated quantities. Likewise, the S-formulation is a more natural starting point when extending the vertical equilibrium model to black-oil type models or including physical effects like compressibility and hysteresis. A more detailed discussion of hysteresis is given in [50].

4 Extending with more flow physics

In this section, we present a general formulation that includes residual trapping, compressibility, dissolution, and resulting hysteresis effects. Apart from capillary fringe, which is discussed in [50], our models incorporate the physical effects discussed by Gasda et al. [30], but the formulation and notation will follow that of standard black-oil models used in the petroleum industry (see Appendix B) to make similarities and differences with models implemented in standard simulators as clear as possible.

The effects discussed in this section can be divided in two categories. The first category consists of capillary and hysteretic effects, which change effective mobilities and capillary pressures, and whose upscaled effect can be derived from considering one vertical column at a time. In Example 1, we also saw how small amounts of CO_2 trapped inside small-scale undulations in the top surface may retard the plume migration, which will in turn lead to hysteretic effects and changes in the effective mobility. Retardation phenomena of this form can therefore also be included in the first category. The second category includes compressibility, which affects the form of the mass balance equations and dissolution that also changes the number of unknowns and equations.

4.1 Sub-scale caprock variations

In potential CO_2 storage sites, the top surface that separates the permeable aquifer from the overlying caprock will have natural variability that spans several length scales. Whereas the characteristic dip angle caused by regional uplift may be constant for hundreds of kilometers, the topography on a kilometer scale will be more varied and contain different types of antiform structures (domes, anticlines, traps, etc) that can be identified from seismic surveys. In [33], it was demonstrated that various kinds of roughness straddling the scale of seismic resolution will impact the plume and potentially cause significant retardation in its updip migration. On an even smaller scale, the top surface is

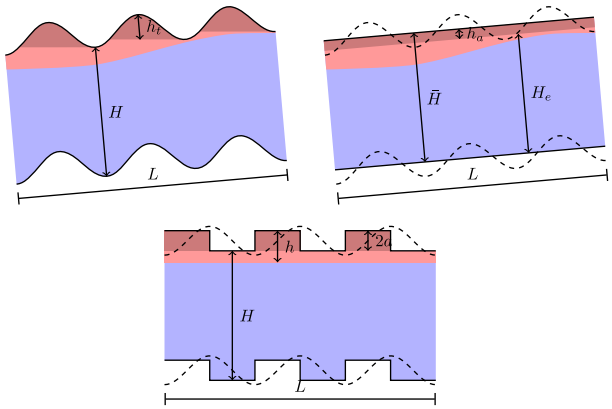


Fig. 11 Upscaling of small-scale caprock undulations. *Upper left:* the representative volume to upscaled, where h_t is the height of the CO_2 column inside the sub-scale traps. *Upper right:* the sub-scale undulations have been replaced by a flat accretion layer inside which CO_2 is immobile, giving an aquifer with a reduced effective height H_e. *Lower:* sub-scale undulations are assumed to be step functions of amplitude a and the dip angle is disregarded to simplify the analytic calculation of upscaled relative permeability

characterized by submeter fluctuations that can only be observed in outcrop studies. Effective models that capture sub-scale roughness and rugosity effects [34, 35] are generally needed to study realistic migration scenarios, because even though VE models are highly efficient, they cannot resolve all relevant structural features in large-scale formations. Here, we briefly review such effective models based on analytical and numerical upscaling techniques for the case without residual trapping in 1D. Figure 11 shows a schematic of our conceptual setup.

In the simplest approach to upscaling, we assume that caprock undulations create a volume in which CO_2 becomes trapped and cannot flow. This volume is represented as a layer of zero horizontal permeability (accretion layer), whose thickness is constant over the averaging length $h_a = \frac{1}{L}\int_0^L h_t(x)\,dx$, see Fig. 11 for notation. Introducing the accretion layer reduces the effective height that is accessible to fluid flow to $H_e = \bar{H} - h_a$, where $\bar{H}$ is the mean aquifer thickness over the length scale L. This model cannot distinguish the effect of different caprock topographies that give the same trapped volumes.

To capture the effect of sub-scale undulations more accurately, we average the depth-integrated rock and rock-fluid properties in the horizontal direction. For permeability, we use a simple homogeneous-equation approach for a periodic medium [54], resulting in an effective permeability $\bar{K} = [(\bar{H}/L)\int_0^L (KH)^{-1}\,dx]^{-1}$. For the relative mobilities, we use a steady-state method that assumes periodicity in the lateral direction (inflow across the left boundary equals outflow across the right boundary) and computes average relative permeabilities as function of volume-averaged saturations values $\bar{S}$. We start by observing that the total velocity is zero after injection has ceased. To obtain a steady-state solution that corresponds to a given average saturation $\bar{S}^i$ inside our averaging volume, we can therefore solve

$$\phi\partial_t s^i + \nabla\Big[\mathbf{k}\frac{\lambda_w\lambda_n}{\lambda_w+\lambda_n}(\Delta\rho\vec{g} + \nabla p_c)\Big] = 0, \quad s^i(\vec{x},0) = \bar{S}^i, \tag{17}$$

until steady state subject to periodic conditions in the x-direction and no flow across the top and bottom surfaces. For each steady-state solution $s^i(\vec{x})$, we reconstruct phase pressures from

$$\nabla\vec{v}_\alpha^i = 0, \qquad \vec{v}_\alpha^i = -\mathbf{k}\lambda_\alpha(s_\alpha^i)\nabla\big[p_\alpha^i - \rho_\alpha\vec{g}\big] \tag{18}$$

subject to a pressure drop Δp in the x-direction and no flow across the top and bottom surfaces. This gives averaged phase fluxes F_α, which can be combined with the upscaled Darcy's law for each phase, $F_\alpha = -\bar{H}\bar{K}\bar{\Lambda}_\alpha\Delta p/L$, to derive values for $\bar{\Lambda}_\alpha$. By repeating this procedure for an increasing sequence of saturation values, we can compute a set of averaged mobilities $\bar{\Lambda}_\alpha$ as function of $\bar{S}$ or an equivalent average plume height $\bar{h}$.

Analytical expressions can be defined if we make further simplifying assumptions [34]. First, we can approximate the fine-scale undulations by a geometric form defined relative to a top surface that is flat on the averaging scale; see the lower plot in Fig. 11. With a sharp interface assumption, it then follows that

$$\begin{aligned}\bar{\Lambda}_n &= \Big[\tfrac{\bar{H}\bar{K}}{L}\textstyle\int_0^L \frac{1}{hK\lambda_{n,e}}\,dx\Big]^{-1},\\ \bar{\Lambda}_w &= \Big[\tfrac{\bar{H}\bar{K}}{L}\textstyle\int_0^L \frac{1}{(\bar{H}-h)K\lambda_w(1)}\,dx\Big]^{-1}.\end{aligned} \tag{19}$$

Second, for small pressure gradients and an averaging scale much smaller than the total domain, we assume that the CO_2–brine interface is flat on the fine scale so that $h(x)$

is a simple function of the function describing the surface topography and Eq. 19 can be computed analytically. From the form of Eq. 19, we also see that $\bar{\Lambda}_n = 0$ if $h(x) = 0$ at some point, and hence, CO_2 will only be able to flow if the interface with brine lies deeper than the lowest point in the local topography.

Example 3 To compare the simple accretion layer model with effective models obtained by assuming sinusoidal or square undulations, we revisit the 1D aquifer from Example 1. In the true model, we set $A = 2$ and use the different upscaled relative permeabilities to compute approximate solutions for the model geometry without undulations ($A = 0$). Because the large-scale structure of the aquifer is the left half of an antiform, the thickness of the CO_2 layer that will be trapped inside small-scale undulations can simply be computed as $h_t(x) = \max_{r\geq x} z_t(r) - z_t(x)$. Then, the resulting function $h_t(x)$ is averaged over a length scale $L \gg L_2$ to obtain

$$a(x) = \Big[\int_{-L/2}^{L/2} h_t(x+r)w(r)\,dr\Big] / \Big[\int_{-L/2}^{L/2} w(r)\,dr\Big],$$

where w is a standard Gaussian mollifier function. Figure 12 illustrates the different steps of this estimation process.

Figure 13 compares the true plume depth with approximate solutions obtained by the accretion-layer model and the two analytical approaches. One might have expected that the model based on sinusoidal substructure would be closest to the fine-scale simulation, but by a coincidence, the simple accretion-layer model is most accurate. The plot to the right in Fig. 13 compares the relative permeabilities of the three effective models with the numerically upscaled relative permeability sampled at three different dip angles $\theta = 0$, 0.0162, and 0.03 that correspond to the angle at the top, midway and at the deepest point of the actual formation, respectively. The plot shows that when the height of the tip of the plume is in the range 2–4, the relative permeability of the accretion-layer model is much closer to those of the numerically upscaled models, and this model will therefore give the best match for the front shock speed. The accretion curve is significantly different from the numerically upscaled curves for $\theta < 0.03$, and hence, this model cannot be expected to provide a good approximation towards the top of the antiform. We also note that whereas the trailing rarefaction wave seen in the fine-scale solution can be explained by the convex shape of the numerically upscaled relative permeability, it cannot be predicted by the concave curves of the 'sinus' and 'square' effective models. Better coarse-scale simulation can be obtained by introducing effective models that depend on both the local trapping volume and the local dip angle. In 2D, this would immediately involve tensorial relative-permeability effects.

4.2 Compressibility and dissolution

In this section, we extend the basic VE model to compressible flow including dissolution of CO_2 into brine and write the resulting model as a set of two-dimensional, two-phase, black-oil equations. We start by writing the upscaled Darcy's law for each phase as,

$$\begin{aligned} \vec{v}_w &= -\lambda_w(p_w, s_w, s_{\max})[\nabla p_w - g\rho_w(p_w, c)\nabla Z], \\ \vec{v}_n &= -\lambda_{rn}(p_n, s_n, s_{\max})[\nabla p_n - g\rho_n(p_n)\nabla Z]. \end{aligned} \tag{20}$$

Here, lower-case symbols have been used to emphasis the similarity with the standard black-oil model and will henceforth represent upscaled quantities unless specified otherwise. The water phase is assumed to consist of brine containing a concentration c of dissolved CO_2, while hysteretic effects are modeled by tracking the maximal saturation $s_{\max}$, as defined in Eq. 15. The viscosities, relative permeabilities, and densities all depend on pressure, but are assumed to be constant within the vertical column of the underlying 3D model. Unlike in Section 3, we now operate with distinct phase pressures p_α, which are evaluated at the

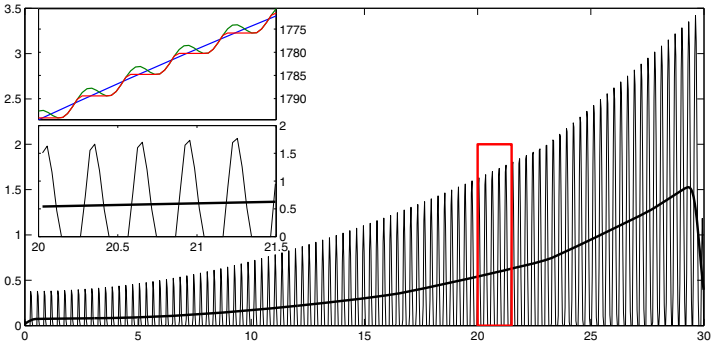


Fig. 12 Estimation of the parameter characterizing sub-scale undulations. The *thin line* is the local trap height $h_t(x)$ and the *thick line* is the averaged height $a(x)$. The *lower inlet* shows a zoom of $h_t(x)$ and $a(x)$ inside the *red square*. The *upper inlet* zooms in on the top surface (*green*), the spill-point level of the local traps (*red*), and the top surface for the model without small-scale undulations (*blue*). (From `upscaleRelPerms.m` and `showUpscaledRelperms.m`)

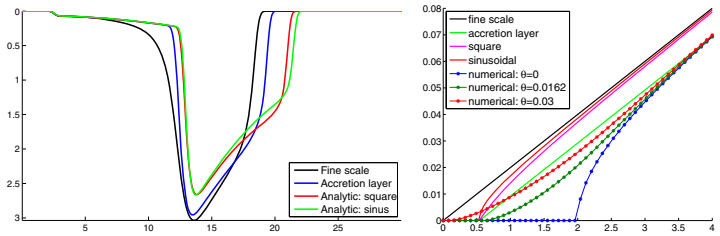


Fig. 13 Upscaling of sub-scale caprock topography. *Left:* The fine-scale solution compared with approximate solutions computed using an accretion-layer model and two analytical models that assume sinusoidal and square sub-scale undulations, respectively. *Right:* The corresponding upscaled relative permeabilities shown as functions of h and sampled at $x = 20$ km. In addition, *lines with markers* show effective relative permeabilities obtained by direct numerical upscaling of the 1D aquifer model for three different dip angles θ. (From `upscalingExample.m` and `showUpscaleRelPerms.m`)

top surface. In other words, p_α refers to pressures defined in cells of the 2D surface grid that follows the caprock topography and should not be confused with pressures inside cells of a 3D volumetric model. If $p_{\alpha,i}$ represents the fine-scale phase pressures at the CO_2-brine interface, it follows by assuming hydrostatic conditions that

$$\begin{aligned} p_w &= p_{w,i} - g\rho_w h, \\ p_\mathrm{n} &= p_{\mathrm{n},i} - g\rho_\mathrm{n} h, \\ p_c &= p_\mathrm{n} - p_w = p_e + g\Delta\rho h \end{aligned} \tag{21}$$

Here, $p_e = p_{\mathrm{n},i} - p_{w,i}$ is the fine-scale entry pressure, whereas p_c can be understood as an upscaled 'capillary pressure'. Because h can be obtained from the upscaled saturation by Eq. 16, the relation between the phase pressures can be written as a function of saturation and pressure:

$$p_w = p_\mathrm{n} - p_c(p, s_\mathrm{n}, s_\mathrm{max}) \tag{22}$$

In a standard black-oil model, p_c is usually a function of s_n and s_max only. To evaluate densities, we introduce the (reciprocal) formation-volume factors b_α for each phase, which relate the densities at reservoir conditions to the densities $\rho_{\alpha,s}$ at surface conditions

$$\begin{aligned} \rho_w(p,c) &= \left(\rho_{w,s} + c\rho_{\mathrm{n},s}\right) b_w(p,c), \\ \rho_\mathrm{n}(p) &= \rho_{\mathrm{n},s} b_\mathrm{n}(p). \end{aligned} \tag{23}$$

Next, the conservation of CO_2, water, and dissolved CO_2 is written as

$$\begin{aligned} \tfrac{\partial}{\partial t}\left[\phi b_\mathrm{n} s_\mathrm{n} + \phi c b_w(1-s_\mathrm{n})\right] + \nabla\cdot\left[b_\mathrm{n}\vec{v}_\mathrm{n} + c b_w \vec{v}_w\right] &= q_\mathrm{n}, \\ \tfrac{\partial}{\partial t}\left[\phi b_w(1-s_\mathrm{n})\right] + \nabla\cdot\left(b_w\vec{v}_w\right) &= q_w, \\ \tfrac{\partial}{\partial t}\left[\phi c b_w(1-s_\mathrm{n})\right] + \nabla\cdot\left(c b_w\vec{v}_w\right) &= q_{\mathrm{n,dis}}. \end{aligned} \tag{24}$$

The effect of dissolution on long-term CO_2 storage has been widely discussed, and in particular, the possibility of enhanced dissolution because of convective mixing [55–57]. For an overview of this topic see [58]. The studies are predominantly theoretical and carried out for homogeneous systems or idealized heterogeneous cases, see [59]. For real cases, the only data on dissolution rates is an estimated upper limit of 1.8 % per year for the Sleipner injection [60]. This estimate is consistent with a later study based on inversion of gravimetric data [61]. For convective mixing in porous media, one can only find indirect evidence of dissolution for analogous problems like salt dissolution, [62, 63]. Even for convection of heat, the few conclusive measurements are of convection induced purely by heating from below, see [64, 65] for reviews.

Despite the uncertainty that surrounds dissolution, we have included this effect in our models to be able to illustrate how dissolution may potentially affect the long-term fate of injected CO_2. For a standard black-oil type of equation, the dissolution of gas is assumed to be instantaneous as long as the fluid phase is undersaturated. That is, the concentration c corresponds to the solution gas/oil ratio r_s used for petroleum systems. To mimic this type of model, we drop the last conservation law in Eq. 24 and set $c = c_\mathrm{max}$ and $s_\mathrm{max}(t) = \max_{\tau\le t} s_\mathrm{n}(\tau)$ wherever $s_\mathrm{n} > 0$. We refer to this as the *instantaneous dissolution model*. Compared with the general black-oil formulation in Appendix B, this would be a model in which brine is given the role of the oleic phase and the supercritical CO_2 is given the role of the gaseous phase.

More accurate modeling of dissolution will introduce two main changes compared with traditional black-oil models: (i) dissolution is rate limited and (ii) dissolution affects the hysteresis variable,[2] i.e., s_max or h_max which tracks the maximum height at which CO_2 exists. The CO_2 will dissolve into brine from the layer of residual CO_2 saturation

[2]Here, hysteresis refers to the vertically-integrated equations and should not be confused with path-dependence between different imbibition and drainage curves in the fine-scale relative permeability and capillary functions, which is discussed in more detail in [36–38, 50]. Because of the sharp-interface assumption, only the end states of the bounding curves enter the equations.

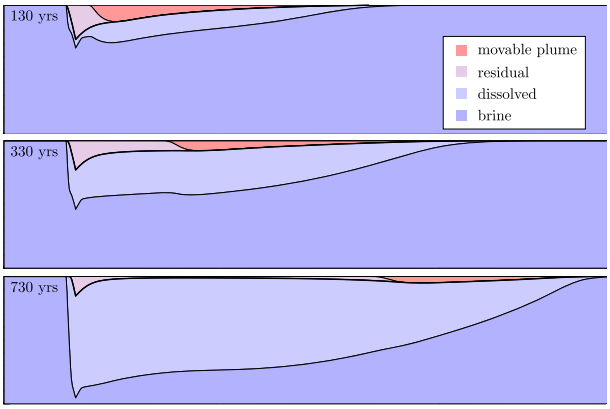


Fig. 14 Evolution of the CO_2 distribution for $D = 1300$ m, $A = 0$, residual saturation, and rate-dependent dissolution. (From `dissolutionExample.m`)

that separates the brine phase from the free CO_2 plume, or directly from the plume when no such layer is present (i.e., when $s_{\max} = s_n$). In `MRST-co2lab`, the corresponding dissolution rate is modeled as

$$q_{n,\mathrm{dis}} = \begin{cases} C\phi, & \text{if } s_{\max} > 0 \text{ and } c < c_{\max}, \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

The dynamics of the maximal saturation $s_{\max}$ used to model hysteresis is represented using two different equations. The first equation describes how dissolution of CO_2 that is residually trapped below the free plume causes $s_{\max}$ to decay

$$\frac{\partial b_n s_{\max}}{\partial t} = -q_{n,\mathrm{dis}}, \text{ if } s_{\max} > s_n, \tag{26}$$

whereas the second equation states that $s_{\max}$ increases with s_n in regions where CO_2 is invading or stagnant,

$$\frac{\partial s_{\max}}{\partial t} = \frac{\partial s_n}{\partial t}, \text{ if } s_{\max} = s_n. \tag{27}$$

Note that $s_{\max}$ can never be less than s_n by definition. We call this the *rate-dependent dissolution model*.

Example 4 To include compressibility effects, we move our 1D aquifer to a depth between 1300 and 850 m. Figure 14 shows the evolution of the CO_2 plume for the case with rate-dependent dissolution and a top surface without small-scale undulations. Figure 15 illustrates how small-scale caprock undulations can effect the plume dynamics. In the top plot, we see how the interplay between undulations and tilt angle gives areas with very thin plume. To resolve these variations in an upscale sense, the effective relative permeability curves discussed in Example 3 obviously should be spatially dependent. In the bottom plot, we see that the residually trapped CO_2 is converted to dissolved CO_2

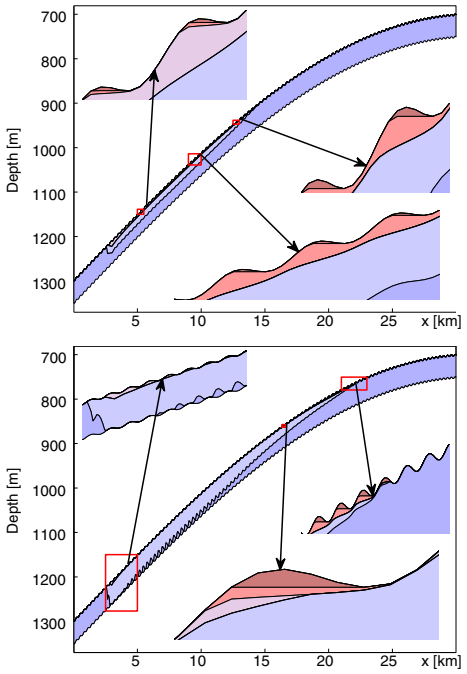


Fig. 15 The CO_2 distribution in global coordinates after 330 years (*top*) and after 930 years (*bottom*) for the case with compressibility, residual saturation, and rate-dependent dissolution. (From `dissolutionExample.m`)

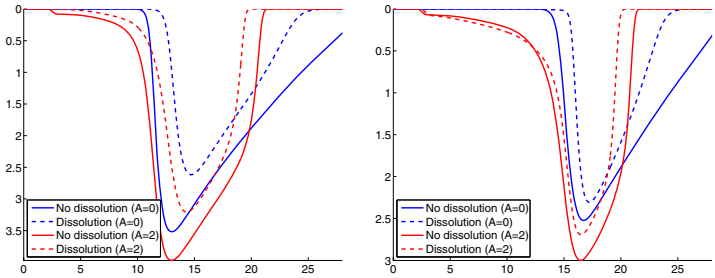


Fig. 16 The flowing CO_2 phase 900 years after the injection has stopped. *Dashed/solid lines* denote models with/without rate-dependent dissolution, whereas *red/blue colors* signify cases with/without small-scale caprock undulations. The residual saturation is $s_{w,r} = 0$ to the left and $s_{w,r} = 0.2$ to the right. (From `dissolutionExample.m`)

except near the well, whereas near the tip of the plume, most of the CO_2 is trapped inside the small-scale undulations. Finally, Fig. 16 compares the flowing CO_2 computed by different VE models and shows how small-scale undulations retard the plume migration, whereas residual trapping and dissolution both reduce the thickness of the plume; dissolution also reduces the thickness of the layer of residual CO_2.

In the next example, we will compare the instantaneous and rate-dependent dissolution models on the Pliocenesand model.

Example 5 The Pliocenesand formation introduced in Example 2 is assumed to have a burial depth of at least 1200 m and a rock compressibility of 10^{-5} bar^{-1}. An annual amount of 5 Mt/year is injected over 50 years from a single injection well located at model coordinates (485, 6647) km. Both brine and CO_2 are compressible: brine has constant compressibility equal $4.3 \cdot 10^{-5}$ bar^{-1}, whereas the density of the CO_2 follows the Span & Wagner PVT model [51] as computed by the open-source `coolprop` package [66] with surface density equal 760 kg/m^3. The dissolution process is assumed to either be non-existent, instantaneous, or follow (25) with $C = 0.05$ m/year and $c_{\max} = 0.03$. This gives is a value which is in the range estimated in the literature [55, 56, 67]. Figure 17 illustrates how one can conceptually think of the fluid distributions in the three resulting sharp-interface models.

Figure 18 shows two snapshots of the evolution of the CO_2 plume along with detailed carbon inventories resulting from the three different simulation models. Several interesting effects can be observed: Without dissolution, the plume will leave behind a relatively large amount of residual CO_2 as it propagates in the upslope direction. After approximately 750 years, the tip of the plume has reached the outskirts of the flat plain (seen to the left in the two snapshots) and starts to exit the simulated domain through the open boundary. After 3000 years, 24 % of the injected CO_2 has left the formation, 20 % is still free to move, while the remaining 56 % can be considered safely stored. With instantaneous dissolution, on the other hand, a large fraction of the injected CO_2 will be dissolved and this will retard the plume migration so that the tip of the plume does not reach the open boundaries within the simulation period. After 3000 years, 99.6 % of the injected CO_2 can be considered safely stored. With rate-dependent resolution, the retardation effect is much less than with instantaneous dissolution and the tip of the plume therefore reaches the outer boundary after approximately 1500 years and starts to leak. By comparing the two upper plots, we also see that the gradual dissolution causes the extent of the residual CO_2 to retract during the 500 year period between the snapshots. After 3000 years, 78 % of the injected CO_2 is dissolved, 6 % is

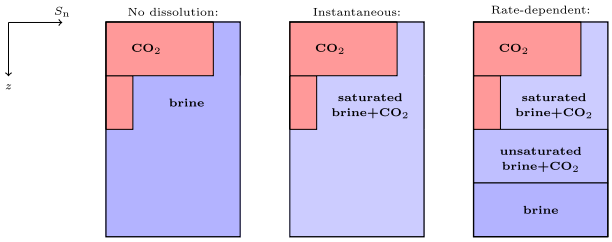


Fig. 17 Conceptual fluid distribution in the three different sharp-interface models used to study the effects of dissolution on the Pliocenesand formation. (In reality, the CO_2-saturated brine will be denser than the resident brine and hence sink to the bottom)

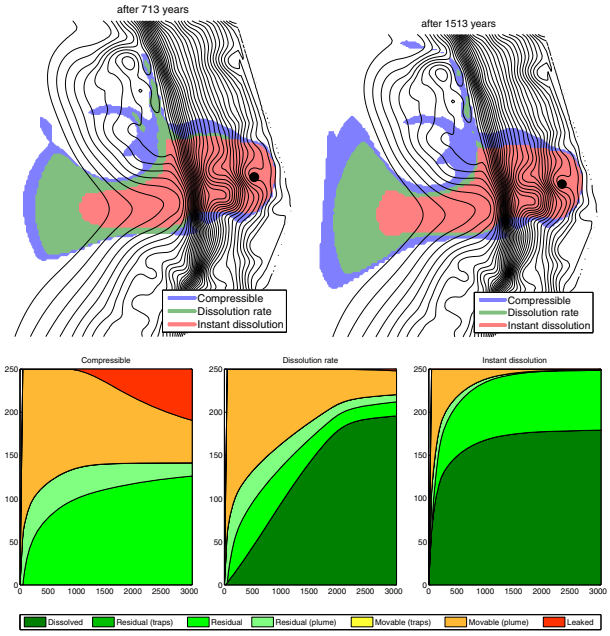


Fig. 18 Modeling the effects of dissolution on the CO_2 migration for a single injection point in the Pliocenesand formation. The *upper plots* show the extent of the residual CO_2 at two different instances in time computed by models without dissolution, with instantaneous dissolution and with rate-dependent dissolution. The *lower plots* show the corresponding carbon inventories (in units mega-tonnes and years since injection started). (From `secondPlioExample.m` and `showSecondPlioExample.m`)

residually trapped, 0.8 % has leaked, and 11 % is still free to move. The total CPU times for the three simulations are 189, 425 and 1118 s, respectively, on an Intel Xeon 3.47 GHz CPU. For the rate-dependent simulation, a large number of time-step cuts contribute to the high computational cost.

Finally, we remark that we have so far not considered lateral transport of CO_2 dissolved in brine induced by density differences. To model this effect, one has to introduce different velocities for water with CO_2 and water without CO_2. Our models did not include this effect since it is approximately one thousand times slower than the movement of the CO_2 plume: a factor one hundred is caused by less density difference with water, and a factor ten can be attributed to the larger viscosity compared to CO_2. However, transport of dissolved CO_2 may be important from a numerical point-of-view because without this effect there is no physical diffusion to smooth out and stabilize the concentration c. Making the other approximation based on putting the denser CO_2-rich water just below the CO_2 plume, which is the mental picture made when introducing convection-enhanced dissolution, will lead to a negative diffusion term that may amplify oscillations.

5 Choice of numerical methods

The VE models discussed above have the same structure as traditional equations for multiphase flow in porous media. The dominant discretization within reservoir simulation is the fully-implicit method with phase-based upstream-mobility weighting and two-point flux approximation. Alternatively, one can use the implicit pressure, explicit saturation (IMPES) method, in which the time step is limited by the CFL restriction on the explicit saturation update, or a sequential splitting that combines a pressure solver with an appropriate explicit or (semi-)implicit saturation solver. Several commercial simulators have vertical-equilibrium options that work with both fully-implicit and sequential solvers (see, e.g., [27, 68]). This option has been added both to improve computational performance and numerical accuracy for thin and/or well-segregated reservoirs. However, none of these simulators can incorporate all the model features discussed herein. Conversely, most research codes used in the literature to study CO_2 migration are based on a simple IMPES or sequential-splitting method and are not publicly available. In particular, the VESA code [26], which to the best of our knowledge is the most general

code reported in the literature, uses a nonstandard non-conservative version of IMPES, but nevertheless seems to work rather well.

We have previously used sequential splitting methods for the simplified incompressible models discussed in Section 3, and this choice has proved to be both efficient and robust in the sense that these methods produce stable solutions [27, 33]. In [42], we also pointed out that the coupling between pressure and transport tends to be weak for migration-type scenarios formulated in a VE framework compared to a standard 3D simulation. Also, for more complex models, our experience is that the combination of stable splitting methods (e.g., as discussed in [69]) with explicit and highly parallel transport solvers [70, 71] will efficiently resolve the primary flow effects for a range of relevant resolutions. This approach is particularly efficient for rectilinear grids with coarse resolutions like seen in the CO_2 Storage Atlas of the Norwegian Continental Shelf [14].

On the other hand, sequential and explicit methods have limitations on fine grids and for almost stationary states. In particular, the gravity effect that creates a flat CO_2 interface (e.g., inside a structural trap) turns up as a second-order term in the transport equation, similar to how capillary forces appear in traditional reservoir simulation. As a result, the transport equation will have a strong parabolic character and will be more tightly coupled to the pressure equations. It is well known within the reservoir simulation community that spatially-dependent and strong capillary pressure is best simulated using fully-implicit methods, since a tight coupling of pressure and transport will reduce the stability and put severe restrictions on the time step of IMPES and other sequential-splitting methods. Implicit methods can also be preferable during the injection stage since they have the ability to take long time steps. An example of such behavior was illustrated in Fig. 10.

For all these reasons, `MRST-co2lab` offers two different types of solvers: (i) sequential-splitting methods with explicit or implicit transport solvers to provide efficient solution of the basic model discussed in Section 3, and (ii) fully-implicit solvers for VE formulations that can easily handle steady-states and incorporate more physical flow effects like compressibility, hysteresis, and dissolution as discussed in Section 4. To the best of our knowledge, fully-implicit VE solvers that include all the main trapping mechanisms have not been reported previously in the literature. We believe that such solvers are particularly important to efficiently and robustly resolve the important stationary states of flat CO_2 interfaces. The fully-implicit solvers in `MRST-co2lab` are implemented using automatic differentiation [12], which makes it straightforward to compute gradients from adjoint equations. Having gradients is important to determine parameter sensitivities or if one wants to use rigorous mathematical optimization methods to propose plausible injection points and strategies.

Acknowledgments The work was funded in part by Statoil ASA and the Research Council of Norway through grants no. 199878 (Numerical CO_2 laboratory), 215641 (MatMoRA-II), 243729 (Simulation and optimization of large-scale, aquifer-wide CO2 injection in the North Sea).

Appendix A: change of coordinate systems

In Section 3, we derived a transport equation on the form

$$\frac{\partial}{\partial t}\Phi(h)+\nabla_{\parallel}\Big[\mathbf{F}_{\mathrm{n}}(h)\vec{V}+\mathbf{G}_{\mathrm{n}}(h)\big[\vec{g}_{\parallel}+g_{\zeta}\nabla_{\parallel}(Z+h)\big]\Big]=Q_{\mathrm{n}},$$

which involves operators $\nabla_{\parallel}$ that are defined relative to a rotated coordinate system that locally aligns with the top surface of the aquifer. The disadvantage of operating in this curvilinear coordinate system is that it is difficult to calculate volumes and that the absolute height of the CO_2 surface will depend on geometric interpretations. The latter is a disadvantage since a stationary state without driving forces, which in physical space is a horizontal interface between CO_2 and brine, will generally not correspond to a constant value of $h(\xi,\eta)$. Hence, it is more complicated to derive numerical schemes that honor this important physical property exactly. In our implementation, we have therefore chosen to work with transport equations formulated directly in the original physical coordinate system. This will potentially introduce two errors.

The first error is associated with the approximation $\nabla_{\parallel}\approx\nabla_{xy}$, which results in an error of the order θ^2. However, for the important stationary case, we have equilibrium in all directions so the VE assumption will in fact be exact in the new coordinate system. Also, for plumes of significant thickness, the interface between CO_2 and water will be near horizontal and assuming equilibrium in the vertical direction will be an approximation that is at least as good as assuming equilibrium in the direction perpendicular to the top surface.

The second error comes from a change of geometry in our interpretation of the physical quantities. In the original formulation, the height h is defined relative to the vector $\vec{e}_{\zeta}$, which is constant if we use a coordinate system aligned with the mean dip angle and variable if we use a curvilinear coordinate system that follows the top surface. When working in the physical coordinates, the height value will

be interpreted in the z-direction. Hence, we will calculate the volume of CO_2 by $h_z A_{xy}$ and interpret all lengths as measured in the xy plane, see Fig. 19. For a planar surface, we have that $A_{xy} = A_{\xi\eta}\cos\theta$ and $h_\zeta = h_z \cos\theta$, from which it follows that the volume is preserved. For a curved surface, the volume attached to each surface patch will generally by a function for which the side faces are not fully aligned with $\vec{e}_\zeta$. Replacing this by a cuboid aligned with the physical coordinates will introduce an error. The bottom plot in Fig. 19 illustrates this; here, the column is a trapezoid with sides that deviate an angle $\theta_{12} = \frac{1}{2}(\theta_1 - \theta_2)$ and $\theta_{23} = \frac{1}{2}(\theta_2 - \theta_3)$, respectively, from being a rectangular. By replacing this trapezoid by a rectangular with the same base and height, we make the following error per unit distance

$$\begin{aligned}\frac{\Delta V}{\Delta x} &= \frac{1}{2}\frac{h^2}{\Delta x}\big[\tan\theta_{12} + \tan\theta_{23}\big] \\ &= \frac{1}{2}\frac{h^2}{\Delta x}\big[\tan\big(\tfrac{1}{2}(\theta_1 - \theta_2)\big) + \tan\big(\tfrac{1}{2}(\theta_2 - \theta_3)\big)\big] \\ &\approx \frac{1}{2}\frac{h^2}{\Delta x}\big[\tfrac{1}{2}(\theta_1 - \theta_2) + \tfrac{1}{2}(\theta_2 - \theta_3)\big] \\ &= \frac{1}{2}h^2\,\frac{\theta_1 - \theta_3}{2\Delta x} \approx -\frac{1}{2}h^2\frac{\partial\theta}{\partial x}.\end{aligned}$$

In other words, we get an error depending on the second derivative of the top surface. The approximation we have adopted is valid for small tilt angles θ and small curvature. It greatly simplifies the numerical code and processing of the grid since there is no need to introduce the local coordinate system. For realistic models, the spatial dimensions are generally working in our favor: grid cells will typically have lengths of the order 100–1000 m, the total height of an aquifer will at most be a few hundred meters, and the mean dip angle up to a few degrees.

In the derivation of the equations, we have integrated the permeability and porosity in the ζ-direction. In the cases where heterogeneity in the vertical direction is considered, we integrate in the z-direction or more precisely in the direction of the grid column in the case of corner-point grids and assume that the main flow will take place in the xy direction. Alternatively, we could integrate vertically at the start, assuming hydrostatic distribution in the vertical direction. This, however, introduces an error proportional to $\theta|v|$, where $|v|$ is the Darcy flow velocity. The end result of the derivation will be the same as we use, but the derivation will indicate a larger error of order $\mathcal{O}(\theta)$. For gravity driven flow, $|v|$ is of order $\mathcal{O}(\theta)$, which means that the pressure reconstruction is of order $\mathcal{O}(\theta^2)$. In addition, the pressure reconstruction is only used inside derivatives, and hence the error in the flux depends on the derivative of θ for constant $|v|$. This derivation is somewhat simpler when deriving the practical method, but we believe that our derivation gives a better description of the combined approximations of the VE assumptions and the numerical implementations in a realistic setting.

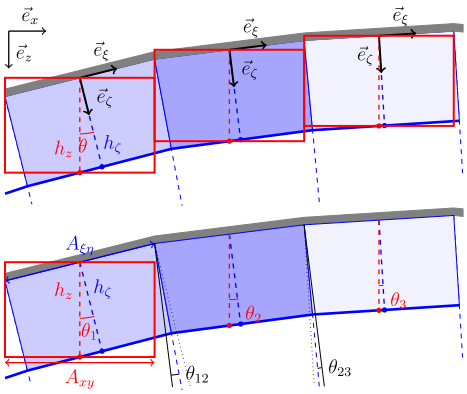


Fig. 19 Approximation of the column geometry for a plume (*thick blue line*) under a sloping caprock (*thick gray line*). The depth of the plume is defined pointwise for each ξ in the direction of the vector $\vec{e}_\zeta$. For a discrete grid model, this gives volumes as shown by the *blue boxes* that can be associated with the centroids of the surface grid. To simplify the calculation, we reinterpret this volume as being along vertical pillars along $\vec{e}_z$, as shown by the *red boxes*. (The aspect ratios in the figures are greatly exaggerated for illustration purposes; typical θ values are a few degrees)

Appendix B: the black-oil equations

The predominant way of simulating multiphase flow in petroleum reservoirs is through the so-called 'black-oil model' in which various chemical species are lumped together to form two components at surface conditions, a denser 'oil' component and a lighter 'gas' component. At reservoir conditions, these two components can be partially or completely dissolved in each other depending on pressure and temperature, forming two hydrocarbon phases, a liquid oleic phase, and/or a gaseous phase, that may coexist with an aqueous phase.

By convention, the black-oil equations are formulated as conservation of the gas, oil, and water component at standard surface conditions combined with a simple PVT model that uses pressure-dependent functions to related fluid volumes at surface and reservoir conditions. To state the equations, we define the inverse formation-volume factors $b_\ell = V_\ell^s / V_\ell$, where V_ℓ^s and V_ℓ denote the volumes occupied by a bulk of component ℓ at surface and reservoir conditions, respectively. The solubility of gas in oil is usually modeled through the pressure-dependent solution gas-oil ratio, $r_{so} = V_g^s / V_o^s$ defined as the volume of gas, measured at standard conditions, that at reservoir

conditions is dissolved in a unit of stock-tank oil. The black-oil framework also opens for oil vaporized in gas, and the presence of the hydrocarbon components in the aqueous phase and water in the gaseous and oleic phases, but these possibilities are not needed for our purpose herein. With this, we can now write the system of continuity equations for our three-component, three-phase system as:

$$\begin{aligned}
\partial_t\big(\phi b_o s_o\big) + \nabla\cdot\big(b_o\vec{v}_o\big) - b_o q_o &= q_o,\\
\partial_t\big(\phi b_w s_w\big) + \nabla\cdot\big(b_w\vec{v}_w\big) - b_w q_w &= q_w,\\
\partial_t\big[\phi\big(b_g s_g + b_o r_{so} s_o\big)\big] + \nabla\cdot\big(b_g\vec{v}_g + b_o r_{so}\vec{v}_o\big)&\\
-\big(b_g q_g + b_o r_{so} q_o\big) &= q_g.
\end{aligned}$$

In addition, we need various closure relationships, which particularly includes relations that express the capillary pressures as functions of phase saturations

$$p_o - p_w = P_{cow}(s_w, s_o), \qquad p_g - p_o = P_{cgo}(s_o, s_g).$$

References

1. Class, H., Ebigbo, A., Helmig, R., Dahle, H.K., Nordbotten, J.M., Celia, M.A., Audigane, P., Darcis, M., Ennis-King, J., Fan, Y., Flemisch, B., Gasda, S.E., Jin, M., Krug, S., Labregere, D., Beni, A.N., Pawar, R.J., Sbai, A., Thomas, S.G., Trenty, L., Wei, L.: A benchmark study on problems related to CO2 storage in geologic formations. Comput. Geosci. **13**(4), 409–434 (2009). doi:10.1007/s10596-009-9146-x
2. Eigestad, G., Dahle, H., Hellevang, B., Riis, F., Johansen, W., Øian, E.: Geological modeling and simulation of CO2 injection in the Johansen formation. Comput. Geosci. **13**(4), 435–450 (2009). doi:10.1007/s10596-009-9153-y
3. Eigestad, G., Dahle, H., Hellevang, B., Johansen, W., Lie, K.-A., Riis, F., Øian, E.: Geological and fluid data for modelling CO2 injection in the Johansen formation. http://www.sintef.no/Projectweb/MatMorA/Downloads/Johansen (2008)
4. SINTEF ICT, The MATLAB Reservoir Simulation Toolbox: Numerical CO_2 laboratory. http://www.sintef.no/co2lab 2014
5. Nilsen, H.M., Lie, K.-A., Møyner, O., Andersen, O.: Spill-point analysis and structural trapping capacity in saline aquifers using MRST-co2lab. Comput. Geosci. **75**, 33–43 (2015). doi:10.1016/j.cageo.2014.11.002
6. Nilsen, H.M., Lie, K.-A., Andersen, O.: Analysis of trapping capacities in the Norwegian North Sea using MRST-co2lab. Comput. Geosci. **79**, 15–26 (2015). doi:10.1016/j.cageo.2015.03.001
7. Lie, K.-A., Nilsen, H.M., Andersen, O., Møyner, O.: A simulation workflow for large-scale CO_2 storage in the Norwegian North Sea. Comput. Geosci. **1**, 1–16 (2015). doi:10.1007/s10596-015-9487-6
8. Andersen, O., Lie, K.-A., Nilsen, H.M.: An open-source toolchain for simulation and optimization of aquifer-wide CO2 storage, Energy Procedia. In: the 8th Trondheim Conference on Capture, Transport and Storage, pp. 1–10 (2015)
9. The MATLAB Reservoir Simulation Toolbox, version 2015a, http://www.sintef.no/MRST/ (2015)
10. Lie, K.-A., Krogstad, S., Ligaarden, I.S., Natvig, J.R., Nilsen, H.M., Skaflestad, B.: Open source MATLAB implementation of consistent discretisations on complex grids. Comput. Geosci. **16**, 297–322 (2012). doi:10.1007/s10596-011-9244-4
11. Lie, K.-A.: An Introduction to Reservoir Simulation Using MATLAB: User guide for the Matlab Reservoir Simulation Toolbox (MRST), SINTEF ICT. http://www.sintef.no/Projectweb/MRST/publications, 1st Edition (2014)
12. Krogstad, S., Lie, K.-A., Møyner, O., Nilsen, H.M., Raynaud, X., Skaflestad, B.: MRST-AD – an open-source framework for rapid prototyping and evaluation of reservoir simulation problems. In: SPE Reservoir Simulation Symposium, pp. 23–25. Houston. doi:10.2118/173317-MS (2015)
13. International Energy Agency, Sleipner benchmark model (2012). http://www.ieaghg.org/index.php?/2009112025/modelling-network.html
14. Halland, E.K., Mujezinović, J., Riis, F. (eds.): CO_2 Storage Atlas: Norwegian Continental Shelf, Norwegian Petroleum Directorate, P.O. Box 600, NO-4003 Stavanger, Norway. http://www.npd.no/en/Publications/Reports/Compiled-CO2-atlas/ (2014)
15. Martin, J.C.: Some mathematical aspects of two phase flow with application to flooding and gravity segregation. Prod. Monthly **22**(6), 22–35 (1958)
16. Coats, K.H., Nielsen, R.L., Terune, M.H., Weber, A.G.: Simulation of three-dimensional, two-phase flow in oil and gas reservoirs. Soc. Pet. Eng. J. Dec, 377–388 (1967)
17. Martin, J.C.: Partial integration of equation of multiphase flow. Soc. Pet. Eng. J. Dec, 370–380 (1968)
18. Coats, K.H., Dempsey, J.R., Henderson, J.H.: The use of vertical equilibrium in two-dimensional simulation of three-dimensional reservoir preformance. Soc. Pet. Eng. J. Mar., 68–71 (1971)
19. Nordbotten, J.M., Celia, M.A.: Geological Storage of CO2: Modeling Approaches for Large-Scale Simulation. Wiley, Hoboken (2012)
20. Yortsos, Y.: A theoretical analysis of vertical flow equilibrium. Transp. Porous Media **18**(2), 107–129 (1995). doi:10.1007/BF01064674
21. Court, B., Bandilla, K.W., Celia, M.A., Janzen, A., Dobossy, M., Nordbotten, J.M.: Applicability of vertical-equilibrium and sharp-interface assumptions in CO2 sequestration modeling. Int. J. Greenhouse Gas Control **10**, 134–147 (2012). doi:10.1016/j.ijggc.2012.04.015
22. Bandilla, K.W., Celia, M.A., Birkholzer, J.T., Cihan, A., Leister, E.C.: Multiphase modeling of geologic carbon sequestration in saline aquifers. Groundwater **53**(3), 362–377 (2015). doi:10.1111/gwat.12315
23. Celia, M.A., Bachu, S., Nordbotten, J.M., Bandilla, K.W.: Status of CO2 storage in deep saline aquifers with emphasis on modeling approaches and practical simulations. Water Resour. Res. **51**(9). doi:10.1002/2015WR017609
24. Nordbotten, J.M., Celia, M.A.: Similarity solutions for fluid injection into confined aquifers. J. Fluid Mech. **561**, 307–327 (2006). doi:10.1017/S0022112006000802
25. Hesse, M.A., Orr, F.M., Tchelepi, H.A.: Gravity currents with residual trapping. J. Fluid. Mech. **611**, 35–60 (2008)
26. Gasda, S.E., Nordbotten, J.M., Celia, M.A.: Vertical equilibrium with sub-scale analytical methods for geological C02 sequestration. Comput. Geosci. **13**(4), 469–481 (2009). doi:10.1007/s10596-009-9138-x
27. Nilsen, H.M., Herrera, P.A., Ashraf, M., Ligaarden, I., Iding, M., Hermanrud, C., Lie, K.-A., Nordbotten, J.M., Dahle, H.K., Keilegavlen, E.: Field-case simulation of CO2-plume migration using vertical-equilibrium models. Energy Procedia **4**(0), 3801–3808 (2011). doi:10.1016/j.egypro.2011.02.315
28. Gasda, S.E., Nordbotten, J.M., Celia, M.A.: Application of simplified models to CO2 migration and immobilization in large-scale geological systems. Int. J. Greenh. Gas Control **9**, 72–84 (2012). doi:10.1016/j.ijggc.2012.03.001
29. Andersen, O., Gasda, S., Nilsen, H.M.: Vertically averaged equations with variable density for CO_2 flow in

porous media. Transp. Porous Media **107**, 95–127 (2015). doi:10.1007/s11242-014-0427-z

30. Gasda, S.E., Nordbotten, J.M., Celia, M.A.: Vertically-averaged approaches to CO2 injection with solubility trapping. Water Resour. Res. **47**, W05528 (2011). doi:10.1029/2010WR009075
31. Mykkeltvedt, T.S., Nordbotten, J.M.: Estimating effective rates of convective mixing from commercial-scale injection. Environ. Earth Sci. **67**(2), 527–535 (2012). doi:10.1007/s12665-012-1674-3
32. Nordbotten, J.M., Dahle, H.K.: Impact of the capillary fringe in vertically integrated models for CO_2 storage. Water Resour. Res. **47**(2), W02537 (2011). doi:10.1029/2009WR008958
33. Nilsen, H.M., Syversveen, A.R., Lie, K.-A., Tveranger, J., Nordbotten, J.M.: Impact of top-surface morphology on CO2 storage capacity. Int. J. Greenh. Gas Control **11**(0), 221–235 (2012). doi:10.1016/j.ijggc.2012.08.012
34. Gasda, S.E., Nilsen, H.M., Dahle, H.K., Gray, W.G.: Effective models for CO_2 migration in geological systems with varying topography. Water Resour. Res. **48**(10). doi:10.1029/2012WR012264
35. Gasda, S.E., Nilsen, H.M., Dahle, H.K.: Impact of structural heterogeneity on upscaled models for large-scale CO2 migration and trapping in saline aquifers. Adv. Water Resour. **62**(Part C (0)), 520–532 (2013). doi:10.1016/j.advwatres.2013.05.003
36. Gasda, S.E., du Plessis, E., Dahle, H.K.: Upscaled models for modeling CO2 injection and migration in geological systems. In: Bastian, P., Kraus, J., Scheichl, R., Wheeler, M. (eds.) Simulation of Flow in Porous Media, Vol. 12 of Radon Series on Computational and Applied Mathematics, pp. 1–38. De Gruyter, Berlin (2013). doi:10.1515/9783110282245.1
37. Doster, F., Nordbotten, J.M., Celia, M.A.: Hysteretic upscaled constitutive relationships for vertically integrated porous media flow. Comput. Visual. Sci. **15**, 147–161 (2012). doi:10.1007/s00791-013-0206-3
38. Doster, F., Nordbotten, J., Celia, M.: Impact of capillary hysteresis and trapping on vertically integrated models for CO2 storage. Adv. Water Resour. **62**(Part C), 465–474 (2013). doi:10.1016/j.advwatres.2013.09.005
39. Bandilla, K.W., Celia, M.A., Elliot, T.R., Person, M., Ellett, K.M., Rupp, J.A., Gable, C., Zhang, Y.: Modeling carbon sequestration in the illinois basin using a vertically-integrated approach. Comput. Vis. Sci. **15**(1), 39–51 (2012). doi:10.1007/s00791-013-0195-2
40. Bandilla, K.W., Kraemer, S.R., Birkholzer, J.T.: Using semi-analytic solutions to approximate the area of potential impact for carbon dioxide injection. Int. J. Greenh. Gas Control **8**, 196–204 (2012). doi:10.1016/j.ijggc.2012.02.009.
41. Gasda, S., Stephansen, A., Aavatsmark, I., Dahle, H.: Upscaled modeling of CO2 injection and migration with coupled thermal processes. Energy Procedia **40**, 384–391 (2013). doi:10.1016/j.egypro.2013.08.044
42. Ligaarden, I.S., Nilsen, H.M.: Numerical aspects of using vertical equilibrium models for simulating CO_2 sequestration. In: Proceedings of ECMOR XII–12th European Conference on the Mathematics of Oil Recovery, p. 2010. EAGE, Oxford
43. Bandilla, K.W., Celia, M.A., Leister, E.: Impact of model complexity on CO_2 plume modeling at Sleipner. Energy Procedia **63**, 3405–3415 (2014). doi:10.1016/j.egypro.2014.11.369. 12th International Conference on Greenhouse Gas Control Technologies, GHGT-12
44. Jacks, H.H., Smith, O.J.E., Mattax, C.C.: The modeling of a three-dimensional reservoir with a two-dimensional reservoir simulator – the use of dynamic pseudo functions. Soc. Pet. Eng. J. **13**(3), 175–185 (1973). doi:10.2118/4071-PA
45. Kyte, J.R., Berry, D.W.: New pseudo functions to control numerical dispersion. Soc. Pet. Eng. J **15**(4), 269–276 (1975). doi:10.2118/5105-PA
46. Stone, H.L.: Rigorous black oil pseudo functions. In: SPE Symposium on Reservoir Simulation , 17–20 February, Anaheim, California, Society of Petroleum Engineers. doi:10.2118/21207-MS (1991)
47. Barker, J., Thibeau, S.: A critical review of the use of pseudorelative permeabilities for upscaling. SPE Reservoir Eng. **12**(2), 138–143 (1997). doi:10.2118/35491-PA
48. Andersen, O., Nilsen, H.M., Lie, K.-A.: Reexamining CO_2 storage capacity and utilization of the Utsira Formation. In: ECMOR XIV – 14^{th} European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8-11 September 2014, EAGE. doi:10.3997/2214-4609.20141809 (2014)
49. Lie, K.-A., Nilsen, H.M., Andersen, O., Møyner, O.: A simulation workflow for large-scale CO_2 storage in the Norwegian North Sea. In: ECMOR XIV – 14^{th} European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8-11 September 2014, EAGE. doi:10.3997/2214-4609.20141877 (2014)
50. Nilsen, H.M., Lie, K.-A., Andersen, O.: Fully implicit simulation of vertical-equilibrium models with hysteresis and capillary fringe. Comput. Geosci. (2015). doi:10.1007/s10596-015-9547-y
51. Span, R., Wagner, W.: A new equation of state for carbon dioxide covering the fluid region from triple-point temperature to 1100 K at pressures up to 800 MPa. J. Phys. Chem. Ref. Data **25**(6), 1509–1597 (1996)
52. Juanes, R., MacMinn, C.W., Szulczewski, M.L.: The footprint of the CO2 plume during carbon dioxide storage in saline aquifers: Storage efficiency for capillary trapping at the basin scale, Transp. Porous Media **82**(1), 19–30 (2010). doi:10.1007/s11242-009-9420-3
53. Halland, E.K., Johansen, W.T., Riis, F. (eds.): CO_2 Storage Atlas: Norwegian North Sea, Norwegian Petroleum Directorate, P. O. Box 600, NO–4003 Stavanger, Norway. http://www.npd.no/no/Publikasjoner/Rapporter/CO2-lagringsatlas/ (2011)
54. Renard, P., De Marsily, G.: Calculating equivalent permeability: a review. Adv. Water Resour. **20**(5), 253–278 (1997)
55. Riaz, A., Hesse, M., Tchelepi, H.A., Orr, F.M.: Onset of convection in a gravitationally unstable diffusive boundary layer in porous media. J. Fluid. Mech. **548**, 87–111 (2006). doi:10.1017/S0022112005007494
56. Pau, G.S.H., Bell, J.B., Pruess, K., Almgren, A.S., Lijewski, M.J., Zhang, K.: High-resolution simulation and characterization of density-driven flow in CO2 storage in saline aquifers. Adv. Water Resour. **33**(4), 443–455 (2010). doi:10.1016/j.advwatres.2010.01.009
57. Hassanzadeh, H., Pooladi-Darvish, M., Keith, D.W.: Scaling behavior of convective mixing, with application to geological storage of CO2. AIChE J. **53**(5), 1121–1131 (2007). doi:10.1002/aic.11157
58. Elenius, M.T.: Convective mixing in geological carbon storage, Ph.D. thesis, University of Bergen. https://bora.uib.no/handle/1956/5540 (2011)
59. Cheng, P., Bestehorn, M., Firoozabadi, A.: Effect of permeability anisotropy on buoyancy-driven flow for CO2 sequestration in saline aquifers. Water Resour. Res. **48**(9). doi:10.1029/2012WR011939
60. Alnes, H., Eiken, O., Nooner, S., Sasagawa, G., Stenvold, T., Zumberge, M.: Results from Sleipner gravity monitoring: Updated density and temperature distribution of the CO2 plume. Energy Procedia **4**(0), 5504–5511 (2011). doi:10.1016/j.egypro.2011.02.536
61. Hauge, V.L., Kolbjrnsen, O.: Bayesian inversion of gravimetric data and assessment of CO_2 dissolution in the utsira formation. Interpretation **3**(2), SP1–SP10 (2015). doi:10.1190/INT-2014-0193.1

62. Van Dam, R.L., Simmons, C.T., Hyndman, D.W., Wood, W.W.: Natural free convection in porous media: First field documentation in groundwater. Geophys. Res. Lett. **36**(11). doi:10.1029/2008GL036906
63. Van Dam, R.L., Eustice, B.P., Hyndman, D.W., Wood, W.W., Simmons, C.T.: Electrical imaging and fluid modeling of convective fingering in a shallow water-table aquifer. Water Resour. Res. **50**(2), 954–968 (2014). doi:10.1002/2013WR013673
64. Anderson, M.P.: Heat as a ground water tracer. Ground Water **43**(6), 951–968 (2005). doi:10.1111/j.1745-6584.2005.00052.x
65. Saar, M.: Review: Geothermal heat as a tracer of large-scale groundwater flow and as a means to determine permeability fields. Hydrogeol. J. **19**(1), 31–52 (2011). doi:10.1007/s10040-010-0657-2
66. Bell, I.H., Wronski, J., Quoilin, S., Lemort, V.: Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolProp. Ind. Eng. Chem. Res. **53**(6), 2498–2508 (2014). doi:10.1021/ie4033999
67. Benson, S.M. et al.: Underground geological storage. In: IPCC Special Report on Carbon Dioxide Capture and Storage, Intergovernmental Panel on Climate Change. Ch. 5. Cambridge University Press, Cambridge (2005)
68. Cavanagh, A.: Benchmark calibration and prediction of the Sleipner CO2 plume from 2006 to 2012. Energy Procedia **37**, 3529–3545, gHGT-11 (2013). doi:10.1016/j.egypro.2013.06.246
69. Lie, K.-A., Nilsen, H.M., Rasmussen, A.F., Raynaud, X.: Fast simulation of polymer injection in heavy-oil reservoirs based on topological sorting and sequential splitting. SPE J. **19**(6), 991–1004 (2014). doi:10.2118/163599-PA
70. Ertsås, M.: Vertically integrated models of CO_2 migration: GPU accelerated simulations, Master's thesis, University of Oslo. http://urn.nb.no/URN:NBN:no-28112 (2011)
71. Prestegård, E.K.: A GPU accelerated simulator for CO2 storage, Master's thesis, Norwegian University of Science and Technology. https://daim.idi.ntnu.no/masteroppgave?id=10345 (2014)

Paper IV

Fully-Implicit Simulation of Vertical-Equilibrium Models with Hysteresis and Capillary Fringe

Halvor M. Nilsen, Knut-Andreas Lie, Odd A. Andersen
Computational Geosciences, **20**, (2016)

IV

Comput Geosci (2016) 20:49–67
DOI 10.1007/s10596-015-9547-y

ORIGINAL PAPER

Fully-implicit simulation of vertical-equilibrium models with hysteresis and capillary fringe

Halvor Møll Nilsen[1] · **Knut-Andreas Lie**[1] · **Odd Andersen**[1,2]

Received: 1 October 2014 / Accepted: 27 October 2015 / Published online: 26 November 2015

Abstract Geological carbon storage represents a new and substantial challenge for the subsurface geosciences. To increase understanding and make good engineering decisions, containment processes and large-scale storage operations must be simulated in a thousand year perspective. A hierarchy of models of increasing computational complexity for analysis and simulation of large-scale CO_2 storage has been implemented as a separate module of the open-source Matlab Reservoir Simulation Toolbox (MRST). This paper describes a general family of two-scale models available in this module. The models consist of two-dimensional flow equations formulated in terms of effective quantities obtained from hydrostatic reconstructions of vertical pressure and saturation distributions. The corresponding formulation is fully implicit and is the first to give a mass-conservative treatment and include general (non-linearized) CO_2 properties. In particular, the models account for compressibility, dissolution, and hysteresis effects in the fine-scale capillary and relative permeability functions and can be used to accurately and efficiently study the combined large-scale and long-term effects of structural, residual, and solubility trapping.

✉ Odd Andersen
odd.andersen@sintef.no

Halvor Møll Nilsen
halvor.m.nilsen@sintef.no

Knut-Andreas Lie
knut-andreas.lie@sintef.no

[1] Department of Applied Mathematics, SINTEF ICT, P.O. Box 124 Blindern, N–0314 Oslo, Norway

[2] Department of Mathematics, University of Bergen, Bergen, Norway

Keywords CO2 storage · Vertical equilibrium · Capillary fringe · Hysteresis

1 Introduction

Carbon capture and storage (CCS), which refers to the capture of CO_2 from large point sources in power generation and industry followed by permanent storage, is one strategy for reducing emissions contributing to climate change. Geological sequestration provides an option for permanent storage of CO_2 at an industrial scale. At present, it is widely accepted that geological carbon storage is possible and can be conducted with reasonable safety for moderately large injection volumes. Nevertheless, if geological storage is to make a significant impact on global emissions, storage operations must be scaled up to a level that involves the injection of gigatonnes of CO_2 per year. CO_2 can potentially be stored in depleted oil or gas fields, unmineable coal seams, and saline aquifers. Herein, we consider large-scale utilization of saline aquifers to store gigatonnes of CO_2.

Before a saline aquifer can be used for large-scale carbon storage, any operator, investor, or governmental agency taking an environmental, societal, or financial risk in a storage operation needs to perform extensive mapping, analysis, and planning to determine its operational performance, financial costs, and long-term security. The only viable way to make such assessments upfront is through computerized model studies that aim to forecast the likely outcomes of the storage operation. This, in turn, requires a good understanding of the injection and migration processes, which are governed by a number of physical mechanisms involving multiple spatial and temporal scales. Analysis and planning will take place in a number of stages that each will require different models and computational methods. During the initial

screening and early injection planning, one primarily needs tools for studying the effects of various trapping mechanisms and determining large-scale storage capacity and injection properties. The large degree of uncertainty inherent in the input parameters (aquifer geometry, petrophysical properties, boundary conditions and pressure regimes, fluid properties, etc.) requires flexible and computationally inexpensive tools that can quickly be turned around to study alternative model assumptions and gradually be refined to account for increasingly complex description of the aquifer geology and flow physics. The tools should also be formulated so that they can use mathematical optimization methods to develop optimal injection scenarios. As one approaches detailed injection planning and the operational phase, one will need to use advanced 3D multiscale multiphysics simulators to study the local and short-term interplay of pressure buildup and multiphase, geomechanical, thermal, and geochemical effects. Likewise, as injection progresses, one will need tools that can perform continuous integration of data and eventually function alongside with instrumentation for long-term monitoring of the injected CO_2 plumes.

Experience has shown that storage sites also vary a lot in operational constraints and physical characteristics [1]. One therefore needs a toolchain of interoperable mathematical models and computational methods so that computer modeling can be adapted on a case-by-case basis to account for the salient physical effects at a level of accuracy that is commensurate with the available data. As a step in this direction, we have developed `MRST-co2lab` [2], which is an add-on module to the open-source Matlab Reservoir Simulation Toolbox (MRST) [3–6]. Through its various add-on modules, MRST offers capabilities for industry-standard reservoir simulation as well as a number of multiscale and multiphysics capabilities that can be used to perform detailed 3D simulation of CO_2 injection on a reservoir scale. The purpose of `MRST-co2lab`, on the other hand, is to provide functionality for modeling and simulation of CO_2 storage in a long-term, large-scale perspective. To this end, the software offers two kind of computational tools: (i) methods that rapidly identify spill paths and estimate upper bounds on structural, residual, and solubility trapping without the use of temporal information [7] and (ii) methods that forecast the likely outcomes of injection operations in a long-term, large-scale perspective by simulating the combined effects of structural, residual, and solubility trapping in a vertically averaged sense assuming a vertical equilibrium. The purpose of the current paper and its companion [8] is to describe the latter simulation capabilities in detail. In a series of related papers [9–11], we combine the tools from `MRST-co2lab` to estimate storage capacities using data sets from the recent CO_2 Storage Atlas of the Norwegian Continental Shelf [12], which describe large-scale saline aquifer systems, as well as more localized models describing the Sleipner injection site and parts of the Johansen formation. In particular, by combining the fast simulation capabilities with mathematical optimization methods, we develop efficient workflows that can be used to suggest placement of storage hubs and optimized injection strategies for large-scale utilization of basin-scale aquifer systems. Finally, we mention that work is in progress to include additional physics like thermal, geochemical, and geomechanical effects in `MRST-co2lab`.

2 Simulating CO_2migration in large-scale aquifer systems

In most realistic cases, the CO_2 injected into a saline aquifer will be in a supercritical phase that is less dense than the surrounding brine and will therefore form a distinct plume that tends to migrate upwards unless held back by one or more trapping mechanisms. The migration process will only cease when all the injected CO_2 has been permanently immobilized by trapping mechanisms or otherwise leaked back out.

As has been clearly demonstrated in benchmark studies [13, 14], it is challenging, if possible at all, to use state-of-the-art methods for 3D simulation to provide spatially resolved simulation of the gravity-controlled migration and trapping of a buoyant CO_2 phase beneath caprock barriers under near-hydrostatic conditions. Moreover, this is a long-term process that can go on for millennia in large sloping aquifer systems. Even for conceptual models with very simple physics, 3D simulators tend to underestimate migration velocities by more than 30 % [14]. This is caused by a disparity in physical scales: Whereas the aquifers themselves may be thousands of square kilometers in areal extent the formation height will typically be a couple of hundreds of meters or less. Gravity-dominated flow during the long migration period leads to the formation of a very thin CO_2 plume, which means that the important trapping and migration processes take place on a meter scale or less. High resolution in the vertical direction is therefore required to resolve the gravity segregation inside of the plume, whose upward migration is sensitive to small undulations in the caprock topography that may both retard the migration of the plume and divert its path. Likewise, high grid resolution is required in the lateral direction to resolve the unstable dynamics of convective dissolution. Dissolution retards and limits the extent of the plume migration and can increase the storage capacity by a factor two or more compared to what is attainable by structural trapping alone. Looking at time scales, one observes that because of the large difference in density between the supercritical CO_2 plume and the resident brine and the fact that the aquifer can be viewed

as a thin, slightly inclined sheet, the segregation of fluids can be considered almost instantaneous compared with the lateral movement of the CO_2 plume. When studying plume migration in a thousand-year perspective, it is therefore reasonable to assume that the fluids at all times are segregated and in vertical equilibrium, see, e.g., [15].

Another argument against the use of traditional 3D simulators to model large-scale, long-term migration of CO_2 in large aquifer systems is the lack of reliable data describing the rock formations. It is well known that geological heterogeneity strongly impacts the movement of fluids in subsurface rock formations regardless of whether the fluid movement is driven by viscous forces on a reservoir scale or by buoyancy forces on a basin scale. However, although saline aquifers have a complex layered structure, detailed modeling of this structure is not supported by available data: Seismic surveys have limited coverage, and there are few exploration wells that can provide core samples. Geological models therefore tend to be relatively homogeneous and caprock topography is in many cases the primary geological parameter that determines long-term CO_2 migration. This is particularly true for the public data sets that describe the enormous volumes of potential storage capacity on the Norwegian Continental Shelf, see, e.g., [12], which is the main motivation for our work.

With a vertical equilibrium assumption, the flow of a thin CO_2 plume can be approximated in terms of its thickness to obtain a 2D depth-integrated simulation model. This type of semi-analytical approach has long traditions in different branches of physics. One example is the shallow-water or Saint Venant equations that describe the flow under a free surface. Another example is the Dupuit approximation in hydrology from 1863, which states that if the water table in an unconfined aquifer is relatively flat and the groundwater is hydrostatic, then the water discharge is proportional to the thickness of the saturated aquifer [16]. In the petroleum industry, similar models were developed more than 50 years ago to study segregated flow [17–20]. In recent years, vertical-equilibrium (VE) models have been extensively used to study gravity-driven CO_2 migration, see [15], and the validity of the assumptions lading up to VE models has been investigated with respect to spatial [21] and temporal [22] scales. Early methods assumed a sharp interface between CO_2 and the resident brine and were successfully applied to study long-term CO_2 migration in the large-scale Utsira [23] and Johansen [24] aquifers. Since then, the VE formulation has been extended to include many relevant physical effects, such as compressibility [25], convective dissolution [26], capillary fringe [27], small-scale caprock topography variations [28–30], as well as various hysteretic effects [31, 32]. Presently, VE formulations can be used to efficiently simulate CO_2 migration and resolve the delicate balances governing structural, residual, and solubility trapping over thousand of years. Using an analytical description for the vertical fluid distribution not only reduces the dimensions of the problem but also lessens the coupling and increases the time constants of the dynamic model. Moreover, important information of the heterogeneities in the underlying 3D medium is preserved, and the errors introduced by assuming vertical equilibrium are in many cases significantly smaller than the errors induced by the overly coarse resolution needed to make a 3D simulation computationally tractable [33]. Recently, it was also demonstrated how one can efficiently simulate vertical-equilibrium models even when the vertical integrals for efficient properties cannot be approximated by closed-form expressions and must be integrated on the fly [34]. Altogether, extensive numerical verification and validation shows that vertically-integrated models in general, and vertical equilibrium models in particular, are attractive means to increase resolution while saving computational cost. For the Sleipner injection, for instance, it has been shown that vertical-equilibrium models give equally good match, if not better, than commercial and research simulators based on traditional 3D discretizations [23, 35, 36].

Through `MRST-co2lab`, we have tried to develop an open-source framework for vertically-integrated modeling of CO_2 that includes the majority of the model features reported in the literature, including in particular fine-scale and upscaled capillary effects, convective dissolution, fine-scale and upscaled hysteretic behavior, vertical heterogeneity in lateral permeability, as well as retardation effects of unresolved caprock variations. Vertically-integrated models have the same structure as traditional equations for multiphase flow in porous media. In fact, although this is not common in the literature, these models can be formulated in the black-oil framework which is the industry standard for simulating petroleum reservoirs. Several commercial reservoir simulators have vertical-equilibrium options that work with both fully-implicit and sequential solution strategies, but to the best of our knowledge, neither of these simulators currently support recent features developed to study the combined large-scale and long-term effects of structural, residual, and solubility trapping of CO_2. On the other hand, most research codes used in the literature to study CO_2 sequestration are based on simple IMPES or sequential-splitting methods that generally lack the robustness of fully-implicit solvers. In particular, the VESA code [37], which to the best of our knowledge is the most general code reported in the literature, uses a nonstandard non-conservative version of IMPES, but nevertheless seems to work rather well. However, like most other research codes, it is not available to the public and thus cannot be used for reproducible research.

To ensure that the models can be seamlessly integrated with standard models used for 3D simulation and easily

be implemented in commercial reservoir simulators, all VE models in `MRST-co2lab`, except for very simple sharp-interface models, are formulated in a black-oil type framework. Likewise, for the numerical discretization, we use standard solution methods developed over the last decades within petroleum reservoir simulation. Apart from adapting to the industry-standard in petroleum engineering, our main motivation for using such fully-implicit methods is that they are particularly robust and efficient for transient simulations approaching a steady state, which will be the case when CO_2 is trapped, as discussed in more detail in [8]. An attractive feature of MRST is that it greatly simplifies the development of fully implicit simulators for complex, coupled systems through a combination of abstract discretization operators and algorithmic differentiation for automatic computation of Jacobians [5, 6]. This greatly simplifies the task of computing gradients and parameter sensitivities using an adjoint formulation and extend the VE models with new features like capillary and hysteretic effects. Apart from a rapid development cycle in MATLAB compared with compiled languages, the main purpose of using MRST is to make the new methods interoperable with existing functionality. A wide range of traditional 3D models and simulators is already implemented in the software, including standard black-oil models and certain EOR models, as well as thermal, geochemical, and simple mechanical effects. MRST also provides efficient preconditioners and multigrid solvers as well as adjoint methods for efficient optimization.

In [8], we discuss vertical-equilibrium models for simulating structural, residual, and solubility trapping based on an assumption of a sharp interface between the injected CO_2 and formation water. These models are highly computationally efficient but also have important limitations, such as a tendency of overestimating the velocity of the tip of the CO_2 plume [27]. In the current paper, we present a more comprehensive framework for VE models that also takes into account fine-scale capillary pressure effects that will create capillary fringes, i.e., smooth transitions between pure CO_2 and pure brine. We discuss in detail the modeling of hysteretic behavior and its impact on the computed effective quantities and present a number of numerical examples that compare various approximations that can be made to simplify the mathematical models and speed up the computations. To keep the presentation as simple as possible, we only consider 2D aquifer cross sections which translate to 1D VE models. Examples of more comprehensive simulation on real 3D models can be found in [9–11, 38, 39].

3 Basic VE models with a capillary fringe

We consider the evolution of a CO_2 plume in a tilted aquifer. As time goes by, structural, residual, and solubility trapping will separate the injected CO_2 into different forms as illustrated in Fig. 1.

In the figure, CO_2 and brine are assumed to be separated by a sharp interface, leading to a fluid distribution consisting of five zones: CO_2 confined in structural traps, free flowing CO_2, brine with residual CO_2, brine with dissolved CO_2, and pure brine. More details about sharp-interface models and the carbon inventory as provided by `MRST-co2lab` is given in [8]. For models with a capillary fringe, an approximate inventory can be computed by reinterpreting the vertical fluid distribution in terms of a sharp-interface model.

The assumption of a sharp interface is reasonable when fluid saturations go from minimum to maximum values

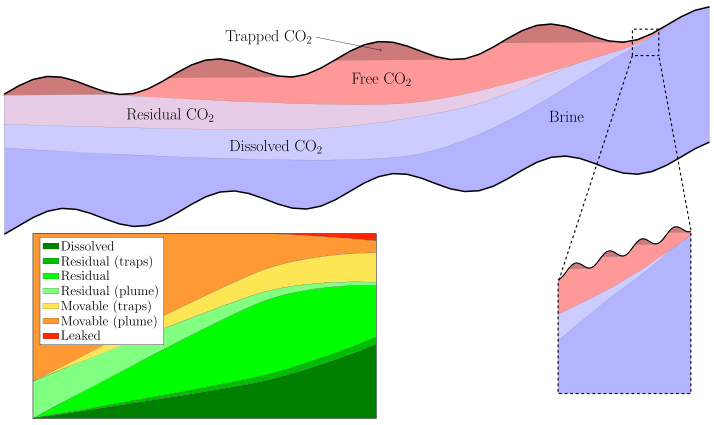


Fig. 1 Schematic of a vertical section of an aquifer showing the different forms in which the injected CO_2 can be present during migration. The right inlet shows small-scale undulations in the top surface which typically will not be resolved in a large-scale model. The left inlet shows a detailed inventory of various categories of trapped CO_2, stacked in terms of increasing leakage risk, as function of time from the end of injection

Comput Geosci (2016) 20:49–67 53

for a change in capillary pressure that is negligible compared with the change in pressures across the whole local thickness of the aquifer. However, if this is not the case, a capillary fringe will develop, within which saturations will vary gradually orthogonal to the main flow direction. The main large-scale effect of introducing a capillary fringe is reduced effective CO_2 mobility for small plume heights, which slows down the advancement of the plume front and introduces a rarefaction at the trailing edge. The effect is qualitatively similar to the retardation effect induced by small-scale caprock undulations, which is discussed in detail in [8].

The starting point for our framework is the standard equations for a two-phase flow system on the fine scale,

$$\frac{\partial}{\partial t}(\phi s_\alpha) + \nabla \cdot \vec{v}_\alpha = q_\alpha, \tag{1}$$

$$\vec{v}_\alpha = -\mathbf{k}\lambda_\alpha(\nabla p_\alpha - \rho_\alpha \vec{g}). \tag{2}$$

Here, brine and CO_2 are modeled as wetting and non-wetting fluids, respectively, ϕ denotes porosity, $\mathbf{k}$ permeability, and $\vec{g}$ the gravity vector, whereas p_α, s_α, λ_α, and q_α denote pressure, saturation, mobility, and volumetric source for phase $\alpha = \{w, n\}$. For simplicity, we limit our discussion in this section to cases without hysteresis, i.e., we assume no residual CO_2 in the brine zone, which is a valid assumption in the advancement (drainage) zone of the CO_2 plume; we will return to models with hysteresis in Section 5. Likewise, the model equations include rock compressibility but disregard fluid compressibility. The effect of variable density in vertical-equilibrium models was investigated by Andersen et al. [25], who concluded that for the vast majority of long-term, large-scale storage systems it is sufficient to account for CO_2 density variations only in the lateral direction (i.e., a semi-compressible model). Both fully compressible and semi-compressible models have been implemented in MRST, but for brevity we herein disregard fluid compressibility when discussing how to derive expressions for effective properties by vertical integration.

As in [8], we introduce a curvilinear coordinate system $(\vec{e}_\xi, \vec{e}_\eta, \vec{e}_\zeta)$ that follows the main directions of flow as illustrated in Fig. 2. In the following, subscript $\parallel$ refers to the $(\vec{e}_\xi, \vec{e}_\eta)$ components of a vector or operator. The new coordinate system is discussed in more detail in [8].

Integrating (1) from top to bottom of the aquifer (and neglecting distortions from the curvilinear nature of the coordinate system [8]), gives:

$$\frac{\partial}{\partial t}\Big[\int_Z^{Z+H} s_\alpha \phi \, d\zeta\Big] + \nabla_\parallel \cdot \Big[\int_Z^{Z+H} \vec{v}_\alpha \, d\zeta\Big] = \int_Z^{Z+H} q_\alpha \, d\zeta. \tag{3}$$

The flow component along $\vec{e}_\zeta$ is zero or negligible, and we therefore assume that the phase pressures p_n and p_w are in hydrostatic equilibrium along this direction (which we henceforth will refer to as 'vertical'). Herein, we will make the simplifying assumption that the density is constant within each vertical column, which is reasonable in most cases in which VE-models are suited for modeling CO_2 storage [25]. Moreover, to simplify the notation, we will integrate from 0 to H instead of from Z to $Z + H$ when considering a single column. Integrating the left two terms of Eq. 3 requires treating fine-scale phase saturation and pressure as functions of depth. If we let $p_{w,i}$ and $p_{n,i}$ denote the phase pressures at the interface between CO_2 and water, defined as the maximum depth h at which mobile CO_2 is present, we can then determine the capillary pressure p_c as a function of depth according to the following formula:

$$\begin{aligned} p_w(z;h) &= p_{w,i} + g\rho_w(z-h), \\ p_n(z;h) &= p_{n,i} + g\rho_n(z-h), \\ p_c(z;h) &= p_{c,i} + g\Delta\rho(z-h). \end{aligned} \tag{4}$$

Above, $\Delta\rho = \rho_n - \rho_w$ is the density difference and $p_{c,i} = p_{n,i} - p_{w,i}$ is the capillary pressure at $z = h$, which is the capillary pressure necessary to initiate displacement of brine by CO_2, also called the *entry pressure*. In the case

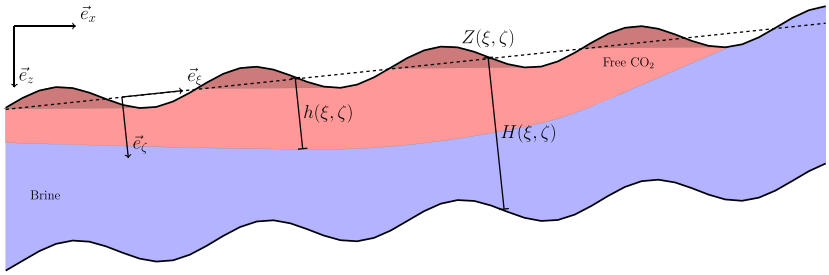


Fig. 2 Schematic of the coordinate systems used to derive vertical-equilibrium models. The *dashed line* indicates main assumed direction of flow within the aquifer, which is here illustrated as straight, but is allowed to be slightly curved in the general case. The aquifer is described by the top surface $Z(\xi, \eta)$ and its thickness $H(\xi, \eta)$ in the ζ-direction

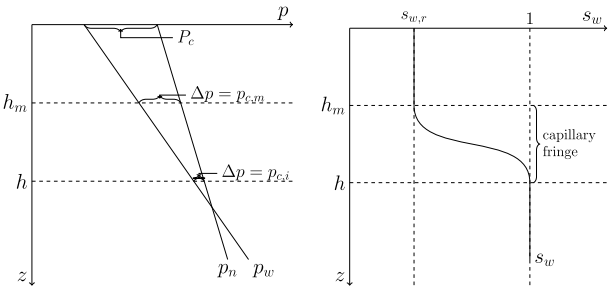


Fig. 3 Diagrams showing phase pressures (*left*) and brine saturation (*right*) as functions of depth z. In the left plot, the capillary pressures at depths h and h_m are indicated. The zone where saturation varies with depth is identified as the capillary fringe on the right plot

without hysteresis, h is equivalent to the maximum depth with nonzero CO_2 saturation s_n. The formula (4) assumes constant vertical density, but an extension of the VE formulation without this assumption is studied in [25], in which case the vertical pressure profile is reconstructed by solving an ordinary differential equation.

The left diagram in Fig. 3 illustrates the hydrostatic pressure profiles of CO_2 and brine and identifies the entry pressure $p_{c,i}$ as well as the capillary pressure value $p_{c,m}$, above which brine stops flowing entirely. The diagram also identifies the capillary pressure at the top surface, P_c, which is sometimes interpreted as the upscaled capillary pressure. The corresponding brine saturation is illustrated in the right diagram, where we see that it varies smoothly between h_m and h (within the capillary fringe) and assumes constant values elsewhere.

From the expression of fine-scale capillary pressure, and assuming a given, monotone function $\hat{p}_c(s_w)$ relating capillary pressure to brine saturation (henceforth, a hat will denote an input function given either as an analytic expression or in tabulated form), we can now reconstruct the fine-scale saturation in the region where $\hat{p}_c^{-1}$ exists:

$$s_w(z) = \begin{cases} s_{w,r} = \hat{p}_c^{-1}\big(p_{c,m}\big), & z \le h_m \\ \hat{p}_c^{-1}\big(p_{c,i} + g\Delta\rho(z-h)\big), & h_m \le z \le h, \\ 1 = \hat{p}_c^{-1}(p_{c,i}), & h \le z. \end{cases} \tag{5}$$

Given this function, we can calculate the upscaled quantities:

$$\begin{aligned} S_{\mathrm{n}} &= \frac{1}{\Phi H}\int_0^H [1 - s_w(z)]\phi(z)\,\mathrm{dz}, \\ \Lambda_{\mathrm{n}} &= \frac{1}{H}\left[\int_0^H \mathbf{k}_\parallel(z)\lambda_{\mathrm{n}}\big(1 - s_w(z)\big)\,\mathrm{dz}\right]\mathbf{K}^{-1}, \\ \Lambda_w &= \frac{1}{H}\left[\int_0^H \mathbf{k}_\parallel(z)\lambda_w\big(s_w(z)\big)\,\mathrm{dz}\right]\mathbf{K}^{-1}, \end{aligned} \tag{6}$$

where the petrophysical properties have been upscaled as follows:

$$\Phi = \frac{1}{H}\int_0^H \phi\, d\zeta, \quad \mathbf{K} = \frac{1}{H}\int_0^H \mathbf{k}_\parallel\, d\zeta.$$

Whereas (5) here has been derived from the incompressible model described by Eqs. 1–2, the definitions remain the same also for a model with compressible fluids. In the rest of the paper, we focus on the evaluation of effective properties in cases where density can be approximated as constant within each vertical column, i.e., as described by the semi-compressible model of [25].

We are now in a position to formulate the upscaled flow equations, which can be developed as explained in [8] using either h or S_n as the primary variable. In addition, the coarse-scale pressure P is taken as the water pressure at the top of the aquifer, $p_w(0) = p_{w,i} - gh\rho_w$, and this is the pressure we will use when evaluating densities henceforth. Implications of the choice of independent variables will be discussed in more details in the next section.

4 Evaluation of effective properties

A main choice distinguishing different VE models is how to reconstruct the saturation profile and evaluate the effective parameters of the upscaled flow equations based on this reconstruction. The full tree of dependencies among fine-scale and upscaled variables is summarized in Fig. 4. In this

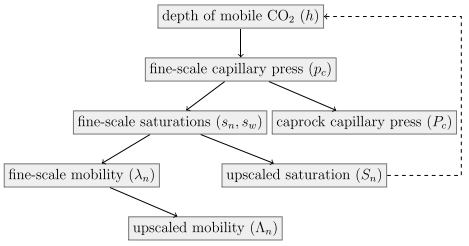


Fig. 4 Dependence among fine-scale and upscaled parameter in a general VE model with capillary fringe. If S_n is the independent variable, then h must first be computed by inverting $S_n(h)$ before one can reconstruct fine-scale fluid distribution and compute upscaled quantities

Table 1 The choice of independent variables and the shape of the vertically-averaged relative-permeability and capillary-pressure functions are affected by three factors in the underlying 3D model: vertical variations in permeability and porosity, capillary pressure, and compressibility

Model type			k_r		p_c	
rock parameters	capillary	compressible	variables	feature	variables	feature
constant	no	no	S	linear	S	linear
varying $\mathbf{k}$	no	no	S	nonlinear	S	linear
varying ϕ	no	no	S	nonlinear	S	nonlinear
constant	no	yes	S	nonlinear	$\boldsymbol{S}$, $\boldsymbol{P}$	nonlinear
constant	yes	no	S	nonlinear	S	nonlinear
constant	yes	yes	$\boldsymbol{S}$, $\boldsymbol{P}$	nonlinear	$\boldsymbol{S}$, $\boldsymbol{P}$	nonlinear
varying ϕ	no	yes	$\boldsymbol{S}$, $\boldsymbol{P}$	nonlinear	$\boldsymbol{S}$, $\boldsymbol{P}$	nonlinear

section, we look at how the choice of primary variables as well as the physical effects included in the model impact the computation of effective properties and compared the pros and cons of the S-formulation and the h-formulations from a numerical point-of-view.

To prepare for this discussion, we will first look at new functional dependencies that appear in the effective properties in general VE models. For constant rock parameters, incompressible fluids, and no capillary forces, the upscaled relative permeabilities and capillary pressure are linear functions of S. If we, for a moment, disregard residual and solubility trapping, there are three main factors that qualitatively change the upscaled relative permeabilities and capillary pressures (see Table 1): (i) vertical heterogeneity in permeability and porosity, (ii) capillary pressure at the fine scale, and (iii) compressibility. Compared with traditional black-oil models, the new qualitative feature introduced is that relative permeabilities and capillary pressure may become pressure dependent for certain combination of these three effects, as summarized in Table 1. Adding an extra functional dependence will increase the computational cost of the property evaluation, while introducing a nonlinearity will generally cause a stronger coupling between the flow and transport equations in the VE model. Small-scale undulations in the caprock will also impact effective relative permeabilities, as explained in [8].

A formulation of the VE model in terms of saturations, compatible with the standard black-oil framework, would only require minimal changes to conventional reservoir simulators and could therefore be considered the most practical choice. From a numerical point-of-view, this gives accumulation terms ($H\Phi S$) that are simple to evaluate: S is the primary unknown whereas H and Φ are functions of the spatial variables and can be computed by preprocessing the geological model. On the other hand, to evaluate the effective mobilities by integrating along the vertical columns of the underlying 3D grid, we need the vertical fluid distribution which can be computed from Eq. 5 once p_c is known. To compute the capillary pressure from Eq. 4, we must first compute h by inverting $S_n(h)$, which requires solving at least one nonlinear equation. This problem is avoided entirely if the model is formulated in terms of plume height because computation of capillary pressure, vertical fluid distribution, and effective mobilities can then be based on function evaluations only. On the other hand, the accumulation term now requires evaluation of an integral along the vertical column to determine $S_n(h)$.

A simple solution to speed up the computation of nonlinear functions is to use precomputed tables. To treat the general case rigorously, such tables will be four-dimensional to reflect the dependency on p, H, S, and $S_{\max}$ for homogeneous cases. Here, $S_{\max}$ represents the historical maximum value of S, and its role is further discussed in Section 5. For heterogeneous media, a separate three-dimensional table for each column would be needed. In other words, a fully rigorous treatment by precomputed tables would require more multidimensional tables than what is practical and efficient for large-scale simulations and require function evaluation at many points that will not be seen in the simulation. To obtain a practical method, we have therefore implemented different approximations in `MRST-co2lab` to speed up the evaluation of effective properties and generally make the S-formulation of VE models more computationally tractable. Table 2 summarizes some cases where simplified models can be obtained. We have found the sharp-interface model (first row) and the capillary fringe approximation (third

Table 2 Table of simplified models for relative permeability

Compressible	Rock param. vert.	Capillary	Aquifer height	Representation
yes	constant	no		analytic, closed form
no	constant	yes	$H \equiv$const	three simple tables of saturation
yes	constant	yes	$h < H$	three 1D tables: $\widetilde{v}(\Delta p)$, $\widetilde{k}_{rw}(\Delta p)$, $\widetilde{k}_{rn}(\Delta p)$
yes	varying	no		calculate p_c, k_{rw}, k_{rn} by sums for each column

row) to be particularly important. The use of one of these simplified models will be sufficient in many cases, especially given the general lack of detailed knowledge about large-scale aquifers.

4.1 Vertically homogeneous porosity and permeability

If we assume a vertically homogeneous case, it is possible to transform all the integrals in Eq. 6 so that the integrands are independent of the density difference. To this end, we define a new integration variable $p = g\Delta\rho(z-h)$, set $\Delta p = gh\Delta\rho = P_c - p_{c,i}$, and introduce the function $\hat{s}_{\mathrm{n}}(\cdot) = (1 - \hat{p}_c^{-1})(\cdot)$. Then, using (5) and assuming that $h < H$, we see that the integrals in Eq. 6 can be written as,

$$\begin{aligned} S_{\mathrm{n}}(\Delta p) &= \frac{1}{gH\Delta\rho}\int_0^{\Delta p} \hat{s}_{\mathrm{n}}(p_{c,i}+p)\,\mathrm{dp} \\ \Lambda_{\mathrm{n}}(\Delta p) &= \frac{1}{gH\Delta\rho}\int_0^{\Delta p} \lambda_{\mathrm{n}}\big(\hat{s}_{\mathrm{n}}(p_{c,i}+p)\big)\,\mathrm{dp} \\ \Lambda_w(\Delta p, h) &= \frac{H-h}{H}\lambda_w(1) + \frac{1}{gH\Delta\rho}\int_{p_{c,m}}^{p_{c,i}+\Delta p} \lambda_w\big(\hat{p}_c^{-1}(p)\big)\,\mathrm{dp}. \end{aligned} \tag{7}$$

The integrals in Eq. 7 are independent of the aquifer height and the density difference (and consequently the global pressure) and can be efficiently implemented using three one-dimensional arrays tabulating them as functions of the pressure difference across the depth of the plume. Notice that as long as $h < H$, Eq. 7 is only different from Eq. 6 if the density is not constant within the column. By using the inverse of the S_{n} table, the tables for the effective mobilities can alternatively be precomputed as functions of S. When used as an approximation, Eq. 7 will be referred to as *P-scaled table*. In the rare case that $h > H$ (i.e., when saturation at $z = H$ is positive), the limits of the integrals in Eq. 7 are incorrect. To obtain values that are always correct, we need to subtract integrals over $[0, g\Delta\rho\max(h-H,0)]$, which also can be expressed in terms of the original tables and replace h by $\min(h, H)$ in the first term of Λ_w. Unfortunately, this system cannot be calculated by only inverting the table, but will require solving a nonlinear system. An approximate, but more computationally efficient approach is to use a linear approximation in this zone. In the case that the fine-scale model has multiple rock types that each has different relative permeability and capillary pressure curves, a unique set of tables must be developed for each rock type.

The expression above does not take into account the fact that the capillary pressure depends on permeability. To introduce this effect, the fine-scale capillary pressure can be approximated in terms of a Leverett J-function, scaled by the permeability,

$$\hat{p}_c(s_w) = \sigma\cos(\theta)\sqrt{\frac{\phi}{|\mathbf{K}|}}J(s_w) = \hat{C}_p^{-1}J(s_w), \tag{8}$$

where σ is surface tension and θ the contact angle. The inverse mapping from capillary pressure values to saturation can now be approximated by $\hat{p}_c^{-1}(p) \approx J^{-1}(\hat{C}_p p)$. Introducing the variable change from p to $y = \hat{C}_p p$ in the integral in Eq. 7, we can calculate all quantities for all columns using only three tables, even if permeabilities differ between grid columns. These tables will be tabulated in values of $\hat{C}_p p$. The resulting approximation will be referred to as *P-K-scaled table*.

4.2 Linear capillary fringe

To investigate the effect of a capillary fringe, a simple linear model for the fringe is instructive. A more complicated physical based model can be found in [32]. If we define the height of the fringe as $h_c = (p_{c,m} - p_{c,i})/g\Delta\rho$, there will be four different types of vertical saturation distributions as illustrated in Fig, 5.

The fine-scale relative permeabilities are assumed to follow a Corey-type model, i.e., $\hat{k}_{r,\mathrm{n}}(s_e) = k_{\mathrm{n}}^e(1-s_e)^{n_{\mathrm{n}}}$ and $\hat{k}_{r,w}(s_e) = k_w^e s_e^{n_w}$, where $s_e = (s_w - s_{w,r})/(1-s_{w,r})$. Let us first consider the case in Fig. 5a. Assuming that ϕ and $\mathbf{k}$ are constant in the vertical direction, we derive an expression for the upscaled saturation as follows:

$$\begin{aligned} S_n(h) &= \frac{1}{H}\int_0^H s_n(z)\,\mathrm{dz} = \frac{1}{H}\int_0^h \Big(\frac{h}{h_c}(1-s_{w,r})\Big)\Big(1-\frac{z}{h}\Big)\,\mathrm{dz} \\ &= \frac{1-s_{w,r}}{H}\int_0^h \frac{h-z}{h_c}\,\mathrm{dz} = \frac{(1-s_{w,r})}{H}\,\frac{1}{2}\Big(\frac{h}{h_c}\Big)^2 \end{aligned}$$

The derivations for the other parameters and cases are similar. To summarize all four cases, we first define the function

$$f(h,n) = \begin{cases} \frac{h_c}{n+1}\left(\frac{h}{h_c}\right)^{n+1}, & \text{if } h < \min(h_c, H), \\ \frac{h_c}{n+1}\left[\left(\frac{h}{h_c}\right)^{n+1} - \left(\frac{h-H}{h_c}\right)^{n+1}\right], & \text{if } H < h < h_c, \\ \frac{h_c}{n+1} + (h-h_c), & \text{if } h_c < h < H, \\ \frac{h_c}{n+1}\left(1 - \left(\frac{H-h}{h_c}\right)^{n+1}\right) + h - h_c, & \text{if } \max(h_c, H) < h, \end{cases}$$

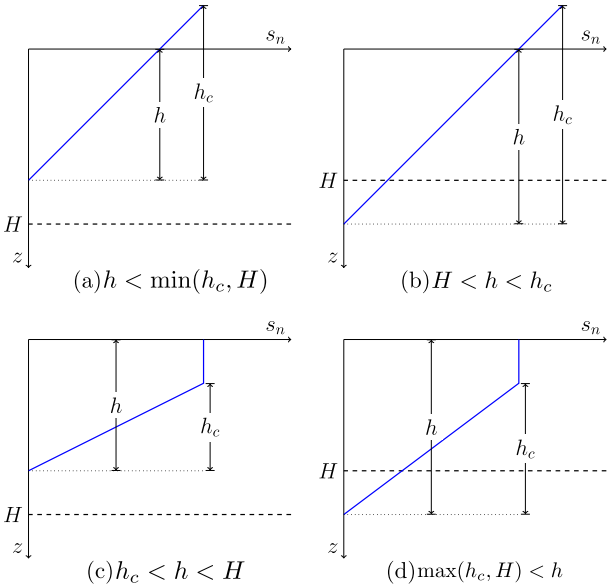


Fig. 5 The four types of vertical saturation distribution arising for a model with a linear capillary fringe. Here, H is the aquifer height, h is the height of the plume, and h_c is the height of a fully developed capillary fringe

and then write the upscaled quantities as

$$\begin{aligned} S_{\mathrm{n}}(h) &= \frac{(1 - s_{w,r})}{H} f(h, 1) \\ \Lambda_{\mathrm{n}}(h) &= \frac{\lambda_{\mathrm{n}}^e}{H} f(h, n_{\mathrm{n}}) \\ \Lambda_w(h) &= \frac{\lambda_w^e}{H} f(H - h + h_c, n_w). \end{aligned} \tag{9}$$

4.3 Comparison of models

If we summarize our discussions so far, we have introduced five different ways of reconstructing the vertical fluid distribution and computing effective properties for use in the averaged flow equations:

- the sharp-interface model presented in [8];
- linear capillary fringe (9);
- P-scaled table (7) with effective properties tabulated against Δp
- P-K-scaled table: same as P-scaled but with capillary pressure given by a Leverett-J function scaled by $\mathbf{K}$;
- S table: the exact relationships (5) and (6).

To compare these models, we will consider a simple example.

Example 1 As our fine-scale model, we assume a homogeneous rock and use Corey type relative permeabilities $k_{r\alpha} = s_\alpha^3$ with end-point scaling so that $k_{r\mathrm{n}}(1 - s_{w,r}) = 1$. The relationship between saturation and capillary pressure is assumed to be on the form $\hat{p}_c(s) \propto s^{-1/2}$ (i.e., a Brooks–Corey type curve with $\alpha = 1/2$), which gives us the following inverse functions

$$\begin{aligned} \hat{p}_c^{-1}(p) &= \max\Big[\Big(\frac{1}{1 + \hat{C}_p p}\Big)^2, s_{w,r}\Big], \\ \hat{J}_c^{-1}(\tilde{p}) &= \max\Big[\Big(\frac{1}{1 + \tilde{p}}\Big)^2, s_{w,r}\Big]. \end{aligned} \tag{10}$$

Moreover, we assume $\sigma \cos(\theta) = 30$ mPa, porosity 0.1, and permeability 100mD. The scaling parameter $\hat{C}_p$ is set to $(0.4 g \Delta\rho H)^{-1}$, and the height of the capillary fringe is set to $h_c = 0.3H$.

Figure 6 shows reconstructed saturation distribution, relative permeability for CO_2, and coarse-scale capillary pressure for the case with $h = 15$ m, $H = 25$ m, and

$s_{w,r} = 0$. For the sharp-interface model, the saturation is zero below $h = 15$ and one above, giving a linear relative permeability. For the other models, we notice the nonlinearity of the upscaled relative permeabilities and capillary pressure for small saturations. For the linear capillary case, the saturation change across the linear fringe is equal $h_c/2H$, and hence, the upscaled functions will only be nonlinear for $s < (1 - s_{w,r})(h_c/2H)$, which equals 0.15 in this example. For the other models, the transition to a linear regime is more gradually and happens for plume heights $h < C/(\Delta\rho g H) \approx 0.4$, or equivalently in the saturation range for which upscaled capillary pressure is comparable in value to the fine-scale capillary pressure. This transition regime can be seen on the lower plot in Fig. 6 for the region $s < 0.2$.

The discrepancies between the relative permeability and the capillary curves for the P-K-scaled model, on one hand, and the P-scaled and the exact curves, on the other, are small for thin plumes. Moreover, we see that the P-K scaled table has a bit smaller capillary fringe than the P-scaled table and the exact case, because of smaller capillary pressure. The advantage of using P-scaled tables is that the curves for different densities and aquifer thicknesses can be calculated using only a single 1D table. The additional advantage of the P-K-scaled table is that varying rock properties can be handled without using additional tables (assuming the relationship (8)). The discrepancy in relative permeabilities between P-scaled table and the exact model can be explained by the fact that we can only compute table values for $h \leq H$, which corresponds approximately to $s \leq 0.7$. Therefore, a linear interpolation is used in the region $s > 0.7$, which corresponds to the case in which the plume height would extend beyond the depth of the aquifer.

5 Hysteresis effects

Residual trapping is a first-order effect for CO_2 migration that will generally lead to irreversibility, also known as hysteresis [40], in the vertical-equilibrium models, see, e.g., [27, 32]. In the models discussed herein, the residually trapped volume is determined by the difference between $S_{\max}$ and S_n, and this difference is inherently linked to hysteretic effects. In sharp-interface models, residual trapping is simple to handle because the local reconstruction of fluid distribution in the vertical columns of the underlying 3D model only has three valid saturation regimes: maximum CO_2 concentration, residual CO_2 concentration, or no CO_2. Hence, it is sufficient to track the height h of the CO_2 plume and the height $h_{\max}$ at which CO_2 has existed throughout the simulation history to define the vertical saturation distribution:

$$s_n(z) = \begin{cases} 1 - s_{w,r}, & z \leq h, \\ s_{n,r}, & h \leq z \leq h_{\max}, \\ 0, & h_{\max} \leq z. \end{cases} \tag{11}$$

Each of these zones has unique mobilities that are independent of the history if the hysteresis model used in the fine-scale 3D description only depends on s and the maximum value $s_{\max}$ seen over the saturation history. If we let λ_α^e denote the end-point of mobility curve λ_α and define

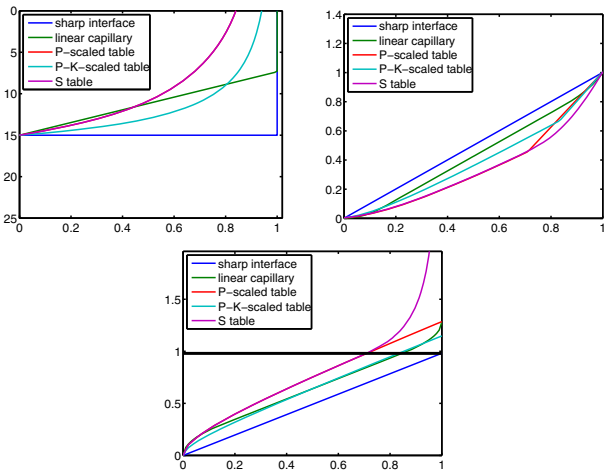


Fig. 6 The figure shows reconstructed saturation distributions for $h = 15$ m (*left*), upscaled relative permeability for CO_2 (*right*), and upscaled capillary pressure with the y-axis scaled by $g\Delta\rho H$ (*bottom*). The height of the given column is $H = 25$ m. (From: `plotVEReconstruction.m`)

$\Phi(z) = \frac{1}{H}\int_0^z \phi(z')\,dz'$ and likewise for $\mathbf{K}(z)$, the upscaled properties in Eq. 6 can be seen to follow a traditional hysteresis model that has a particular simple form:

$$\begin{aligned} S_{\mathrm{n}} &= \big[\Phi(h)(1-s_{w,r}) + (\Phi(h_{\max}) - \Phi(h))s_{\mathrm{n},r}\big]/\Phi(H) \\ S_{\max} &= \Phi(h_{\max})(1-s_{w,r})/\Phi(H), \\ \Lambda_{\mathrm{n}}\mathbf{K} &= \lambda_{\mathrm{n}}^{e}\mathbf{K}(h), \\ \Lambda_{w}\mathbf{K} &= \big[(\mathbf{K}(H) - \mathbf{K}(h_{\max}))\lambda_w(1) + (\mathbf{K}(h_{\max}) - \mathbf{K}(h))\lambda_{w,r}\big], \\ P_c &= gh\Delta\rho. \end{aligned} \tag{12}$$

While the hysteretic behavior of S_{n} and Λ_w is obvious from the expressions above and will appear explicitly in a pure h-formulation, the hysteretic behavior of Λ_{n} and P_c follows implicitly from the fact that h now also depends on $S_{\max}$. In particular, whereas P_c is a linear function of h, in the S formulation, it is a much more complex function $\hat{P}_c(S_n, S_{\max}, P)$.

The problem gets more complicated when the underlying model has a capillary fringe, since the shape of the fringe then generally will depend on previous profiles. If the fine-scale capillary pressure follows a hysteretic curve $\hat{p}_c(s_w, s_{w,\max})$, where $s_{w,\max} = 1 - s_{\max}$, the expressions in Eq. 6 for the averaged quantities are still valid when $s_w = s_{w,\max}$. In the general case, the fine-scale saturation is reconstructed as follows:

$$\begin{aligned} p_{c,\max}(s_w) &= \hat{p}_c(s_w, s_w) \\ s_{\max}(z) &= 1 - p_{c,\max}^{-1}(p_c(z; h_{\max})) \\ S_{\max} &= \frac{1}{H}\int_0^{h_{\max}} 1 - p_{c,\max}^{-1}(z)\,\mathrm{dz} \\ s_{\mathrm{n}}(z) &= \begin{cases} \hat{s}_{\mathrm{n}}\big(p_c(z;h), s_{w,\max}(z)\big), & z < h, \\ s_{\mathrm{n}}^{e}\big(s_{\max}(z)\big), & h < z < h_{\max}, \\ 0, & h_{\max} < z, \end{cases} \end{aligned} \tag{13}$$

where $\hat{s}_{\mathrm{n}}(\cdot,\cdot) = (1 - \hat{p}_c^{-1})(\cdot,\cdot)$ and the lower endpoint $s_{\mathrm{n}}^{e}(s_{\max})$ on the relative permeability curve emanating from $s_{\max}$ is an analogy to the residual saturation seen in the sharp-interface models. In the case of a general function $\hat{p}_c(s_w, s_{w,\max})$, the reconstruction in Eq. 13 must be computed on the fly by solving a set of nonlinear problems.

The simpler case of a vertically homogeneous aquifer and a plume that never extends to the full height of the aquifer was investigated recently by Gasda et al. [31], who concluded that the main hysteretic effects can be handled by simplified trapping models and that the specific form of the model is less important than capturing the reduced mobility and increased coarse-scale 'capillary pressure' of CO_2 for small plume heights. The effects of different fine-scale models are mainly captured if the simple hysteretic model accounts for the nonlinearity of the relative permeability caused by the capillary fringe and the correct amount of effective residually trapped CO_2. In the rest of the section, we will discuss simplified hysteresis models that enable efficient reconstructions based on tabulated values, much in the same way as discussed in Section 4.

5.1 Endpoint-scaling model

We start by assuming that all scanning curves in the fine-scale relative permeability and capillary pressure functions have the same functional form, which implies that the scanning curves are 'reversible' in the sense that the imbibition curve also models a subsequent drainage process. Moreover, we assume that the residual endpoint of each scanning curve for the relative permeability is defined as

$$s_{\mathrm{n}}^{e}(s_{\max}) = \frac{s_{\mathrm{n},r}}{1 - s_{w,r}} s_{\max} = C s_{\max}, \tag{14}$$

e.g., as illustrated in Fig. 7. If $\hat{k}_{r,\mathrm{n}}$ denotes the primary relative permeability curve for $s \in [0,1]$, then the scanning curve for $s \in [s_{\mathrm{n}}^{e}(s_{\max}), s_{\max}]$ is defined as $k_{r,\mathrm{n}}(s, s_{\max}) = \hat{k}_{r,\mathrm{n}}\big(\hat{s}_{\mathrm{eff}}(s)\big)$, where the rescaled saturation variable used to define the scanning curve reads

$$\hat{s}_{\mathrm{eff}}(s) = s_{\max}\frac{s - s_{\mathrm{n}}^{e}}{s_{\max} - s_{\mathrm{n}}^{e}} = s - \frac{C}{1-C}(s_{\max} - s) \tag{15}$$

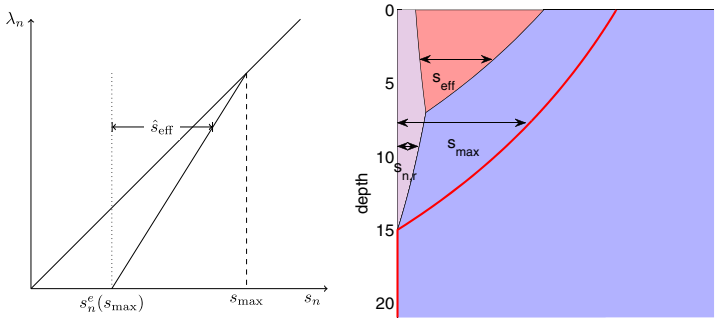


Fig. 7 *Left:* Illustration of a simple fine-scale relative permeability function with hysteresis effects, in which the scanning curve between $s_{\mathrm{n}}^{e}(s_{\max})$ and $s_{\max}$ has the same functional form as the primary curve between 0 and 1. *Right:* Reconstruction of vertical saturation distribution

for the case when $\hat{k}_r$ is linear as in Fig. 7. In the general case, we will use this as an approximation so that if we know $s(z) = 1 - s_w(z)$ from Eq. 5 and $s_{\max}(z)$ from Eq. 13, we can compute $s_{\text{eff}}(z)$ as follows

$$s_{\text{eff}}(z) = s(z) - \frac{C}{1-C}\big(s_{\max}(z) - s(z)\big), \tag{16}$$

which translates to the following expression for the averaged saturations,

$$S_{\text{eff}}(h, h_{\max}) = S(h, h_{\max}) - \frac{C}{1-C}\big(S_{\max}(h_{\max}) - S(h, h_{\max})\big). \tag{17}$$

From the assumptions of $\hat{p}_c(s_{\text{eff}})$, we can now find h, $h_{\max}$ and consequently $s_{\text{eff}}(z)$ and $s_{\max}(z)$. Then, we have all we need to reconstruct the vertical saturation distribution and compute the upscaled mobilities. In particular, the saturation distribution reads,

$$s_{\text{n}}(z) = \begin{cases} s(z), & z \le h, \\ C s_{\max}(z), & h \le z \le h_{\max} \\ 0, & h_{\max} \le z. \end{cases} \tag{18}$$

For $z < h$, the saturation that is already residually trapped is given by $s(z) - s_{\text{eff}}$. Including hysteretic effects does not change the capillary-fringe model for $h_{\max} = h$, and the new model reduces to the sharp-interface model in the case of a constant capillary pressure function. The model also has the properties that the residually trapped CO_2in a column will be $C S_{\max} = s_{\text{n},e} S_{\max}/(1-s_{w,e})$, which will be the case in any model in which the residual endpoint of the relative permeability is $s_{\text{n},e} s_{\max}/(1 - s_{w,e})$.

In the S-formulation, h and $h_{\max}$ are not known and must be computed from S, $S_{\max}$, and P. To compute h, we start by observing that since all scanning curves have the same functional form, the fine-scale capillary pressure will be a function of s_{eff}. Replacing s_n by s_{eff}, we can either use the relationship between $\hat{p}_c^{-1}$ and S_n given by Eq. 5 and 6 to form a nonlinear equation for h, or alternatively use a precomputed tabular that implements one of the (approximate) relationships discussed in Section 4. For instance, if we choose to use Eq. 7, h and $h_{\max}$ can be computed from the global pressure P and the precomputed, P-scaled table $\hat{S}_n(\Delta p) = gH\Delta\rho\, S_n(\Delta p)$ as follows

$$\begin{aligned} h &= \frac{1}{g\Delta\rho(P)}\, \hat{S}_n^{-1}\left(gH\Delta\rho(P) \cdot S_{\text{eff}}\right), \\ h_{\max} &= \frac{1}{g\Delta\rho(P)}\, \hat{S}_n^{-1}\left(gH\Delta\rho(P) \cdot S_{\max}\right). \end{aligned} \tag{19}$$

We can now insert h in Eq. 4 to compute the capillary pressure $p_c(z)$, from which we get $s_{\text{eff}}(z) = \hat{p}_c^{-1}(p_c(z))$, and similarly we get $s_{\max}(z)$ by inserting $h_{\max}$ instead of h. Inserting $s_{\text{eff}}(z)$ and $s_{\max}(z)$ into Eq. 16 and solving for $s(z)$, we finally can compute the vertical saturation distribution from Eq. 18.

5.2 Sharp-interface type model

Another model can be obtained if we assume that the capillary fringe has exactly the same shape as for the case without hysteresis. For this to be possible, the effective saturation used to evaluate the fine-scale relative-permeability curves has to have a drop from the maximum residually trapped CO_2 saturation to zero at h. In other words,

$$\begin{aligned} s(z) \quad &= \begin{cases} \hat{s}_n\big(p_c(z)\big), & z < h \\ s_{n,r}, & h < z < h_{\max} \\ 0, & h_{\max} \le z, \end{cases} \\ k_{r,\text{n}}(z) &= \begin{cases} \hat{k}_{r,\text{n}}(s(z)), & z < h, \\ 0, & h < z. \end{cases} \end{aligned} \tag{20}$$

Likewise, the only way one can have a region of constant residual saturation between h and $h_{\max}$ is if the capillary function $\hat{p}_c$ is constant for $s_n \in [0, \tilde{s}]$, where $\tilde{s} \ge s_{n,r}$. (In a sharp-interface model, $\tilde{s} = 1$ by definition.) The capillary function is therefore distinctly different from the one used for the model in Section 5.2.

In the new model, the averaged saturation is expressed as

$$S_n(h_{\max}, h) = S_{\text{eff}}(h) + \frac{h_{\max} - h}{H} s_{\text{n},r} \tag{21}$$

If we further assume that the relative permeabilities only depend on the effective residual saturation, and not on the maximum saturation, for values with $S > s_{\text{n}}^e$, the original fine-scale curves can be used to evaluate the averaged properties.

Because h and $h_{\max}$ are not known in the S-formulation, they must be computed from S, $S_{\max}$, and P. To this end, one can,for instance, use a P-scaled tabular (7) to give the following equation

$$S = S_n\left(g\Delta\rho(P) \cdot h\right) + \frac{h_{\max} - h}{H} s_{\text{n},r}. \tag{22}$$

This expression involves $h_{\max}$, which can be calculated beforehand using the same expression with $h = h_{\max}$.

As far as we understand, a model very similar to the one outlined above has previously been used in [24] to investigate CO_2 trapping mechanisms in the Johansen Formation from the Norwegian North Sea.

5.3 Comparison of models

To illustrate the how the hysteresis effects discussed above change the effective properties of the VE equations, we will consider two examples. In the first example, we revisit the setup from Example 1.

Example 2 Figure 8 illustrates hysteretic effects using the end-point scaling model applied to the basic setup from Example 1 for a case in which $h = 7$ m and $h_{\max} = 15$ m. Starting with the sharp-interface model without a capillary fringe, we see that the imbibition curves are linear but with a steeper inclination than the drainage curves. Similarly, in the other models, the primary drainage curves have been shifted and scaled to describe the imbibition and a possible subsequent drainage process. We notice that also in the hysteretic case, the effective curves are mostly linear except for small plume heights.

In the third example, we will illustrate how different choices for the underlying fine-scale capillary function can change the reconstructed saturation distribution and the effective parameters in the coarse-scale model. Specifically, we will construct a special capillary function that will enable us the compare the endpoint-scaling model Section 5.1 and the sharp-interface type model from Section 5.2.

Example 3 As explained in Section 5.2, the only way one can define a model that has a constant residual saturation between h and $h_{\max}$ is to have a capillary function that is constant on an interval. Using the same setup as Eq. 10 in Example 1, and setting $p = p_c - p_{c,i}$ so that $p = 0$ at $z = h$,

$$s_w(p) = \begin{cases} \max\left[\left(\frac{C}{C+p}\right)^2 - \tilde{s}, s_{w,r}\right], & p > 0, \\ 1, & p < 0. \end{cases} \tag{23}$$

This will give a capillary pressure curve

$$p_c(s_w) = \begin{cases} \infty, & s_w < s_{w,r}, \\ C(s_w + \tilde{s})^{-\frac{1}{2}} - 1 + p_{i,n}, & s_{w,r} < s_w < 1 - \tilde{s}, \\ p_{i,n}, & 1 - \tilde{s} < s_w. \end{cases} \tag{24}$$

We assume that $k_{rn}(s) = s^3$ and $H = 25$ m, as in the previous examples, and set $\tilde{s} = s_{r,n} = 0.2$, $C = 1.6 g \Delta\rho H$, and $\Delta\rho = 400$ kg/m^3. Figure 9 shows how the two different capillary functions and the two hysteresis models effect the reconstruction of fine-scale saturation. The corresponding effective properties are shown in Fig. 10.

Here, we notice in particular that the relative permeability for the sharp-interface model has a very strong dependence on the residual saturation parameter compared with the equivalent endpoint-scaling model with capillary pressure function (23). This is because the latter has a hysteresis model based on scanning curves, while the sharp-interface type model has not. With the endpoint-scaling framework it is also possible to have effective relative permeabilities that increases slower than linearly from $\Lambda_n = 0$ as shown in Fig. 10. For the sharp-interface type model, on the other hand, Λ_n has a finite and positive derivative at $\Lambda_n = 0$.

As noted before, a formulation in terms of primary variables h and $h_{\max}$ would allow general hysteresis models to be calculated explicitly on the fly. In principle, this can also be done from saturation variables as well, but then one has to solve nonlinear equations involving functions that are defined in terms of integrals. The main advantage of using the endpoint-scaling model is that with this model one can calculate all quantities in terms of 1D tables under the assumption of homogeneous columns, and that one can include K-scaling and pressure variations, provided that certain approximations are made if the plume extends to the bottom of the aquifer. We also notice that the effect of small-scale caprock undulations can be included in terms of an accretion layer model (see [8]) that uses $h_{\text{eff}} = h - h_{\text{trap}}$ to evaluate the effective relative permeability. The capillary pressure, however, is still computed using h.

At the end, we will use a simple example to illustrate the effect of dissolution on models including capillary fringe and small-scale geometrical trapping. For a discussion of the dissolution model itself, see [8] in which the dynamics

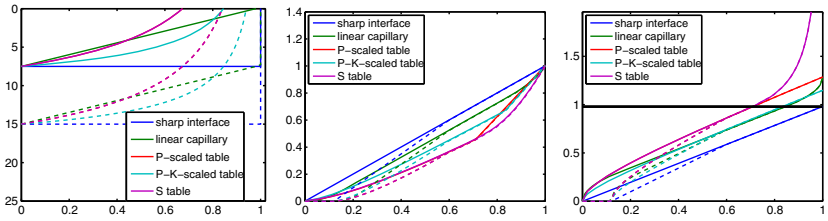


Fig. 8 The endpoint-scaling model applied to the basic setup from Example 1 as shown in Fig. 6. In the left plot, *solid lines* refer to $s_e(z)$ reconstructed from $h = 7$ m, while *dashed lines* correspond to $s_{\max}(z)$. In the middle and right plot, *solid lines* are the drainage curves, whereas *dashed lines* are the imbibition curves for $S_{\max} = \max(0.6, S)$. (From: `plotVEReconstruction.m`)

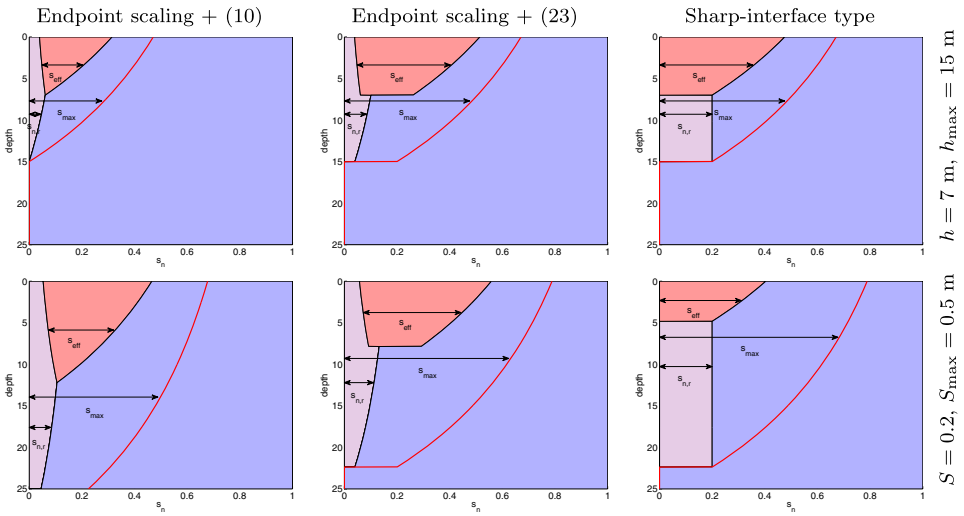


Fig. 9 The figure illustrates how different assumptions for the underlying capillary-pressure curve change the vertical average model. The capillary-pressure and relative-permeability curves for the given models are shown in Fig. 10. (From: `plotVEProfilesAndRelperms.m`)

of $s_{\max}$ is discussed for the sharp-interface model. Again, we assume that the effect of dissolution is to first to reduce $s_{\max}$, which means that with instantaneous dissolution (i.e., a black-oil type model), $s_{\max}(t)$ will always be equal to the maximum saturation value observed during the previous imbibition process. With a finite dissolution rate, on the other hand, $s_{\max}(t)$ may be smaller than the maximum value seen during imbibition. To simplify our computations, we also make the approximation that the reconstruction from s, $s_{\max}$ can be done as for the case without dissolution. The main difference between a sharp-interface and a capillary-fringe model is that the depth of the plume will be larger for the latter, and hence more CO_2 will be instantaneously dissolved.

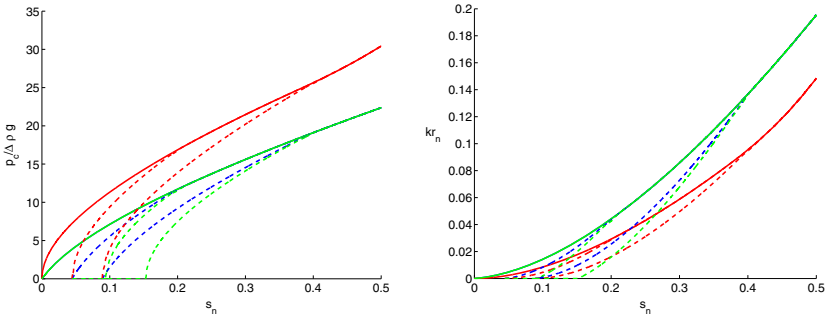


Fig. 10 Effective capillary pressure (*left*) and effective relative permeability (*right*) for the model three different models in Fig. 9. The *solid lines* are the primary imbibition curves and the *dashed lines* correspond to scanning curves for $S_{\max}$ equal 0.2 and 0.4, respectively. *Green curves* represent the sharp-interface type model, whereas *blue* and *red curves* represent the endpoint-scaling model with capillary pressure curves (23) and (10), respectively. (From: `plotVEProfilesAndRelperms.m`)

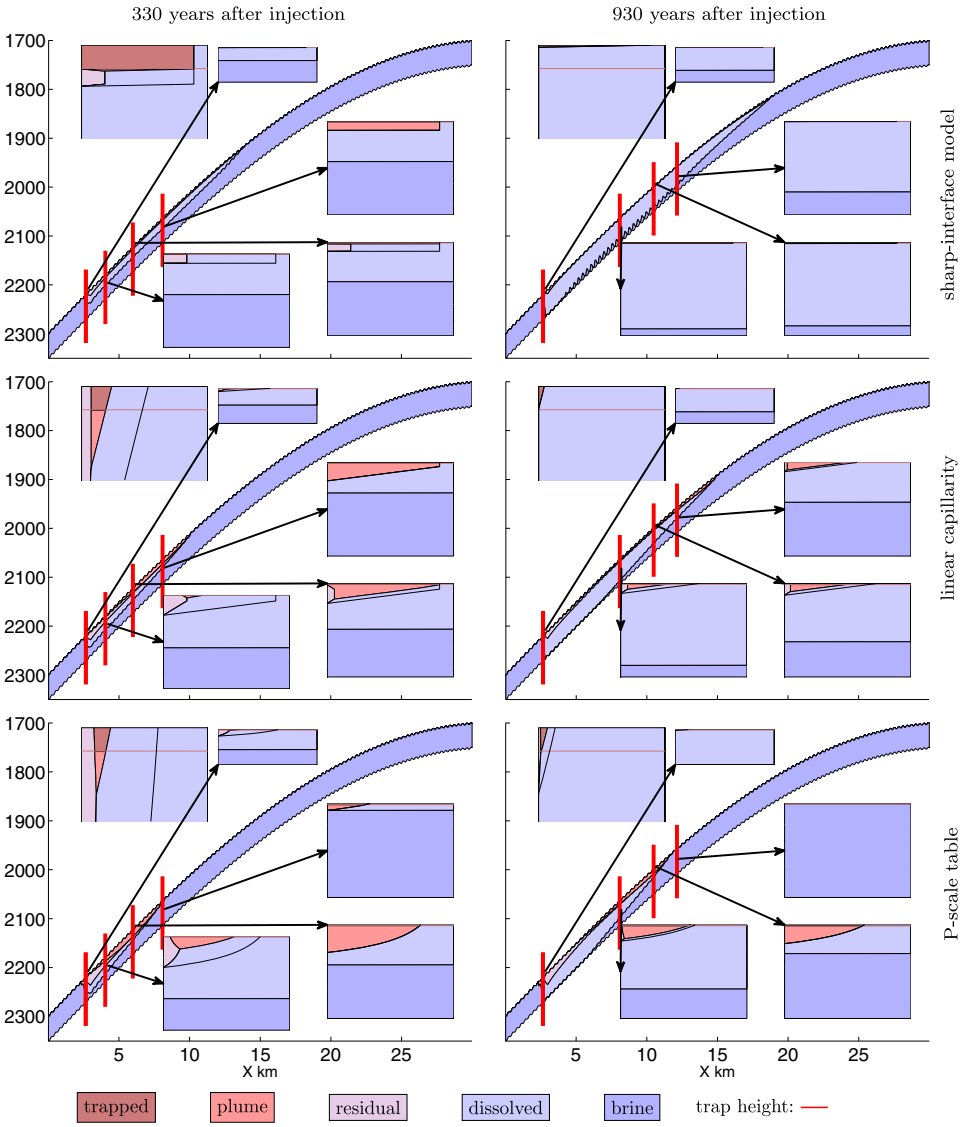


Fig. 11 The CO_2 distribution in global coordinates 330 and 930 years after injection ceased in the simple 1D sloping aquifer. The main figure shows the full aquifer depth for scale, whereas the inlets is the reconstruction of the plume in four different cells of the model. The inlet in the top left corner of each model is a zoom of the first reconstruction for the depth of four times the trapping height, which was 0.4 m. (From `dissolutionExampleCOMG2`)

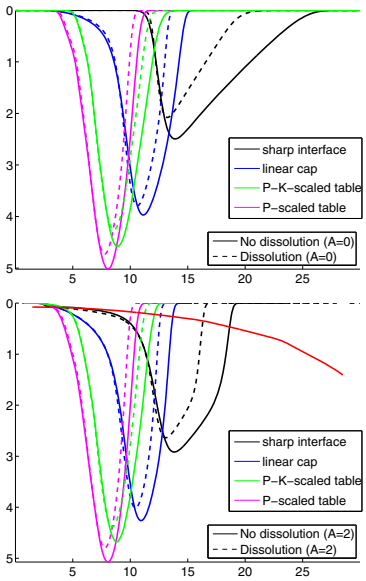


Fig. 12 The flowing CO_2 computed by different fluid models for the simple 1D sloping aquifer. Colors signify different approximate methods for evaluating effective properties, *dashed lines* are models with dissolution, and *solid lines* are models without. In the upper plot, the top-surface is smooth ($A = 0$) and in the lower plot it has small-scale undulations ($A = 2$ m). The *red line* shows the average height of the local traps (see [8, Fig. 10]). (From `dissolutionExampleCOMG2`)

Example 4 In this example, we consider a simple 1D model of a sealing caprock introduced in [8],

$$z_t = D - L_1 \sin\Big(\frac{x}{L_1}\Big)\tan\theta + A\sin\Big(\frac{2\pi x}{L_2}\Big), \tag{25}$$

where D is maximum depth, $\theta = 0.03$ is the initial tilt angel, $L_1 = 20$ km is the characteristic length of the antiform structure, and $L_2 = 0.3$ km is the wave-length L_2 m of the small-scale structures, which we assume have an amplitude $A = 0$ or $A = 2$ m. The density of CO_2 is assumed to follow a model taken from [41].

Figure 11 shows snapshots from three full simulations that include the coupled effect of dissolution and small-scale caprock undulations using three different models for fine-scale mobility and capillary pressure: sharp interface, linear capillary, and P-scaled table. We employ endpoint scaling for the hysteresis, which in the case of vanishing capillary fringe reduce to the sharp-interface hysteresis approach. In all simulations, we have used the dissolution rate taken from [8] for all models. In Fig. 12, we show the effective saturation, where we for completeness also have included results obtained using P-K-scaled tables. We see that the difference in the effective saturation for models with and without dissolution is much smaller at the trailing shock side. The reason for this is that the residual saturation has to be dissolved first, so the dynamics of the flowing part is relatively independent of the dissolution dynamics. However, dissolution clearly retards the imbibing front, and this effect is qualitatively the same for models with and without capillary fringe. Comparing the two subplots, we see that the capillary-fringe models are less retarded by small-scale trapping, and the actual amount of structurally trapped CO_2 is also much smaller. We also see that the endpoint of the trailing rarefaction follows the local volumes of the sub-scale traps (shown as a red line in the figure) for the sharp-interface model, but falls below for the capillary-fringe models. This effect of the capillary fringe can be identified by comparing the amount of CO_2 above the red line in the inlets of Fig. 11.

Example 5 To show the applicability of the models discussed herein, we compare the capillary fringe model with P-scaled table to a sharp-interface model in a large-scale, realistic setting. We use a simulation grid of the Utsira aquifer in the Norwegian North Sea to simulate the injection and migration of CO_2 in a long-term perspective. We assume a simple setup in which CO_2 is injected into the aquifer at two separate sites, indicated with black dots on the upper left plot in Fig. 13. Each site injects 150 megatons of CO_2 over a period of 50 years, followed by a 1000-year migration period. During this period, some CO_2 will be permanently trapped in residual form or within pockets (structural traps) of the caprock surface. The trapping state as a function of time is displayed in the two diagrams on the bottom row of Fig. 13. The positions of the corresponding free plumes are shown on the upper row. We observe that the sharp-interface model here results in a more irregularly-shaped plume as a result of increased gravity fingering, whereas the capillary fringe model spreads out more evenly. This can be explained by increased plume heights in this model (because of lower average saturation in each vertical column), which leads to a stronger upscaled capillary force that drives plume expansion in all directions.

The 2D grid used for the two VE simulations contains 23,393 cells, and each simulation ran within less than 5 min on a standard workstation PC. A full 3D simulation with a vertical resolution of 100 cells would thus have required 2.3 million cells, which in our experience leads to at least a factor ten larger runtimes.

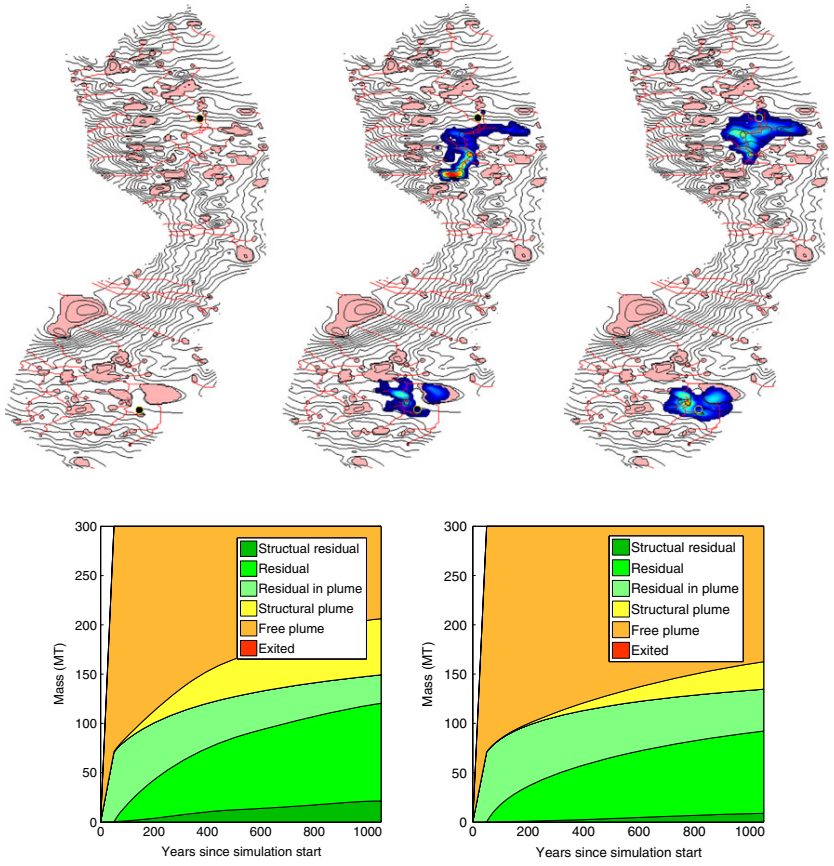


Fig. 13 Impact of different upscaled CO_2 properties on a large-scale injection and migration scenario into the Utsira aquifer. *Upper left:* Topographical map of the Utsira caprock, with structural traps and spill paths indicated in *red* and injection points indicated with *black dots*. *Upper middle and right:* location of the free-flowing plume after 1000 years of migration using a sharp-interface (*middle*) and capillary fringe (*right*) model. Maximum plume thicknesses are 28.3 m (sharp interface) and 18.8 m (capillary fringe). Lower row: Corresponding breakdown of total injected CO_2 into various states of trapping as a function of time. Left plot represents the simulation with a sharp interface, right plot with a capillary fringe. "Structural" trapping refers to the part of the plume contained within the structural traps of the caprock

6 Concluding remarks

In the paper, we have presented a general, fully-implicit, and mass-conservative framework for VE models on unstructured grids that can be used to simulate the combined large-scale and long-term effects of structural, residual, and solubility trapping. The framework is implemented as part of the free and open-source code `MRST-co2lab`, which also includes the capabilities of constructing grid models from GIS data or the industry-standard corner-point format. As such, the software can be applied to perform highly resolved simulations of potential injection sites.

In the paper and in `MRST-co2lab`, we mainly use model formulations that are as close as possible to standard

black-oil formulations used by the petroleum industry. This is a pragmatic choice, motivated by the fact that if VE models are formulated in the black-oil framework, one can:

- easily include the VE models into existing commercial and academic simulators developed for this framework and potentially develop hybrid models that combine VE and 3D models in different parts of the domain,
- extend the VE models with the wide range of physical effects that are described in the literature using the black-oil framework, and
- utilize robust and reasonably efficient numerical methods developed over the past four decades. As discussed in more detail in [8], these methods are particularly suitable for accurately computing steady states that correspond to various kinds of trapping in the long-term transient migration of CO_2.

Nevertheless, if one considers VE models from a pure computational point of view, it follows from the derivation and discussion above that the reconstruction of the fine-scale fluid distribution is more computationally efficient if the model is formulated in terms of $h, h_{\max}$. At present, we do not consider the computational gain to be worth the implementational cost and the added modeling complexity of using different primary unknowns for VE and 3D models. Likewise, an implementation of a general code in $h, h_{\max}$ with possible extensions to higher order hysteresis models [32], is left for future work. However, most parts of the current open-source code will be reusable also for such codes formulated in $h, h_{\max}$.

If CO_2 storage becomes a large industry and sufficiently detailed models can be obtained, we believe that a fit-to-purpose simulator with on-the-fly numerical calculation of all the fine-scale properties would be a useful tool to estimate the impact of the approximations made in our work to make a tractable numerical code. Although the numerical reconstruction of the vertical fluid distribution is computationally expensive, the update inside each 2D cell, which corresponds to a column of cells in the underlying 3D model, is independent of the other 2D cells and thus trivial to parallelize. A prototype implementation that uses GPUs to accelerate concurrent parts of the algorithm has already been developed [34]. Preliminary results obtained with this simulator indicates that assuming a sharp interface, as discussed in [8], tends to overestimate the retardation effects that structural trapping has on plume migration.

Acknowledgments The work was funded in part by Statoil ASA and the Research Council of Norway through grants no. 199878 (Numerical CO_2 laboratory) and 215641 (MatMoRA-II).

References

1. Eiken, O., Ringrose, P., Hermanrud, C., Nazarian, B., Torp, T.A., Høier, L.: Lessons learned from 14 years of {CCS} operations: Sleipner, in salah and snøhvit. Energy Procedia **4**, 5541–5548 (2011). doi:10.1016/j.egypro.2011.02.541. 10th International Conference on Greenhouse Gas Control Technologies
2. SINTEF ICT: The MATLAB reservoir simulation toolbox: numerical CO_2 laboratory (2014). http://www.sintef.no/co2lab
3. The MATLAB Reservoir Simulation Toolbox: version 2015a. http://www.sintef.no/MRST/ (2015)
4. Lie, K.-A., Krogstad, S., Ligaarden, I.S., Natvig, J.R., Nilsen, H., Skaflestad, B., Open-source, M.A.T.L.A.B.: implementation of consistent discretisations on complex grids. Comput. Geosci. **16**, 297–322 (2012). doi:10.1007/s10596-011-9244-4
5. Krogstad, S., Lie, K.-A., Møyner, O., Nilsen, H.M., Raynaud, X., Skaflestad B.: MRST-AD—an open-source framework for rapid prototyping and evaluation of reservoir simulation problems (2015)
6. Lie, K.-A.: An introduction to reservoir simulation using MATLAB: User guide for the Matlab reservoir simulation toolbox (MRST), SINTEF ICT. http://www.sintef.no/Projectweb/MRST/publications. 1st Edition (2014)
7. Nilsen, H.M., Lie, K.-A., Møyner, O., Andersen, O.: Spill-point analysis and structural trapping capacity in saline aquifers using MRST-co2lab. Computers & Geoscience **75**, 33–43 (2015). doi:10.1016/j.cageo.2014.11.002
8. Nilsen, H.M., Lie, K.-A., Andersen, O.: Robust simulation of sharp-interface models for fast estimation of CO_2 trapping capacity in large-scale aquifer systems. Comput. Geosci. to appear
9. Nilsen, H.M., Lie, K.-A., Andersen, O.: Analysis of trapping capacities in the Norwegian North Sea using MRST-co2lab. Computers & Geoscience **79**, 15–26 (2015). doi:10.1016/j.cageo.2015.03.001
10. Lie, K.-A., Nilsen, H.M., Andersen, O., Møyner, O.: A simulation workflow for large-scale CO_2 storage in the Norwegian North Sea, Computational Geosciences
11. Andersen, O., Lie, K.-A., Nilsen, H.M.: An open-source toolchain for simulation and optimization of aquifer-wide co2 storage, Energy Procedia
12. Halland, E.K., Mujezinović, J., Riis, F. (eds.): CO_2 Storage Atlas: Norwegian Continental Shelf, Norwegian Petroleum Directorate, P.O. Box 600, NO-4003 Stavanger, Norway (2014). http://www.npd.no/en/Publications/Reports/Compiled-CO2-atlas/
13. Class, H., Ebigbo, A., Helmig, R., Dahle, H.K., Nordbotten, J.M., Celia, M.A., Audigane, P., Darcis, M., Ennis-King, J., Fan, Y., Flemisch, B., Gasda, S.E., Jin, M., Krug, S., Labregere, D., Beni, A.N., Pawar, R.J., Sbai, A., Thomas, S.G., Trenty, L., Wei, L.: A benchmark study on problems related to CO2 storage in geologic formations. Comput Geosci **13**(4), 409–434 (2009). doi:10.1007/s10596-009-9146-x

14. Nordbotten, J.M., Flemisch, B., Gasda, S.E., Nilsen, H.M., Fan, Y., Pickup, G.E., Wiese, B., Celia, M.A., Dahle, H.K., Eigestad, G.T., Pruess, K.: Uncertainties in practical simulation of CO2 storage. Int. J. Greenh. Gas Control. **9**(0), 234–242 (2012). doi:10.1016/j.ijggc.2012.03.007
15. Nordbotten, J.M., Celia, M.A.: Geological storage of CO2: Modeling approaches for large-scale simulation. Wiley, Hoboken (2012)
16. Bear, J.: Dynamics of fluids in porous media, Dover books on physics and chemistry, Dover publications, Incorporated (1988). http://books.google.no/books?id=lurrmlFGhTEC
17. Martin, J.C.: Some mathematical aspects of two phase flow with application to flooding and gravity segregation. Prod. Monthly **22**(6), 22–35 (1958)
18. Coats, K.H., Nielsen, R.L., Terune, M.H., Weber, A.G.: Simulation of three-dimensional, two-phase flow in oil and gas reservoirs. Soc. Pet. Eng. J. Dec., 377–388 (1967)
19. Martin, J.C.: Partial integration of equation of multiphase flow. Soc. Pet. Eng. J. Dec, 370–380 (1968)
20. Coats, K.H., Dempsey, J.R., Henderson, J.H.: The use of vertical equilibrium in two-dimensional simulation of three-dimensional reservoir preformance. Soc. Pet. Eng. J. Mar, 68–71 (1971)
21. Yortsos, Y.: A theoretical analysis of vertical flow equilibrium. Transp. Porous Media **18**(2), 107–129 (1995). doi:10.1007/BF01064674
22. Court, B., Bandilla, K.W., Celia, M.A., Janzen, A., Dobossy, M., Nordbotten, J.M.: Applicability of vertical-equilibrium and sharp-interface assumptions in CO2 sequestration modeling. Int. J. Greenh. Gas Control **10**, 134–147 (2012). doi:10.1016/j.ijggc.2012.04.015
23. Nilsen, H.M., Herrera, P.A., Ashraf, M., Ligaarden, I., Iding, M., Hermanrud, C., Lie, K.-A., Nordbotten, J.M., Dahle, H.K., Keilegavlen, E.: Field-case simulation of CO2-plume migration using vertical-equilibrium models. Energy Procedia **4**(0), 3801–3808 (2011). doi:10.1016/j.egypro.2011.02.315
24. Gasda, S.E., Nordbotten, J.M., Celia, M.A.: Application of simplified models to CO2 migration and immobilization in large-scale geological systems. Int. J. Greenh. Gas Control **9**, 72–84 (2012). doi:10.1016/j.ijggc.2012.03.001
25. Andersen, O., Gasda, S., Nilsen, H.M.: Vertically averaged equations with variable density for CO_2 flow in porous media. Transp. Porous Media **107**, 95–127 (2015). doi:10.1007/s11242-014-0427-z
26. Gasda, S.E., Nordbotten, J.M., Celia, M.A.: Vertically-averaged approaches to CO2 injection with solubility trapping. Water Resour. Res. **47**, W05528 (2011). doi:10.1029/2010WR009075
27. Nordbotten, J.M., Dahle, H.K.: Impact of the capillary fringe in vertically integrated models for CO_2 storage. Water Resour. Res. **47**(2), W02537 (2011). doi:10.1029/2009WR008958
28. Nilsen, H.M., Syversveen, A.R., Lie, K.-A., Tveranger, J., Nordbotten, J.M.: Impact of top-surface morphology on CO2 storage capacity. Int. J. Greenh. Gas Control **11**(0), 221–235 (2012). 10.1016/j.ijggc.2012.08.012
29. Gasda, S.E., Nilsen, H.M., Dahle, H.K., Gray, W.G.: Effective models for CO_2 migration in geological systems with varying topography. Water Resour. Res. **48**(10). doi:10.1029/2012WR012264
30. Gasda, S.E., Nilsen, H.M., Dahle, H.K.: Impact of structural heterogeneity on upscaled models for large-scale CO_2 migration and trapping in saline aquifers. Adv. Water Resour. **62**(Part C (0)), 520–532 (2013). doi:10.1016/j.advwatres.2013.05.003
31. Gasda, S.E., du Plessis, E., Dahle, H.K.: Upscaled models for modeling CO2 injection and migration in geological systems. In: Bastian, P., Kraus, J., Scheichl, R., Wheeler, M. (eds.) Simulation of Flow in Porous Media, Vol. 12 of Radon Series on Computational and Applied Mathematics, De Gruyter, pp. 1–38. Berlin, Boston (2013). doi:10.1515/9783110282245.1
32. Doster, F., Nordbotten, J.M., Celia, M.A.: Hysteretic upscaled constitutive relationships for vertically integrated porous media flow. Comput. Visual. Sci. **15**, 147–161 (2012). 10.1007/s00791-013-0206-3
33. Ligaarden, I.S., Nilsen, H.M.: Numerical aspects of using vertical equilibrium models for simulating CO_2 sequestration. In: Proceedings of ECMOR XII–12th European Conference on the Mathematics of Oil Recovery, EAGE, Oxford, UK, p. 2010
34. Seternes, G.: Simulations of co2 migration with a fully-integrated ve model on the gpu, Master's thesis, Norwegian University of Science and Technology. https://daim.idi.ntnu.no/masteroppgave?id=12404 (2015)
35. Singh, V., Cavanagh, A., Hansen, H., Nazarian, B., Iding, M., Ringrose, P.: Reservoir modeling of CO_2 plume behavior calibrated against monitoring data from Sleipner, Norway. In: SPE Annual Technical Conference and Exhibition, 19-22 September 2010, Florence, Italy, 2010, SPE 134891-MS. doi:10.2118/134891-MS
36. Bandilla, K.W., Celia, M.A., Leister, E.: Impact of model complexity on CO_2 plume modeling at sleipner. Energy Procedia **63**, 3405–3415 (2014). doi:10.1016/j.egypro.2014.11.369. 12th International Conference on Greenhouse Gas Control Technologies, GHGT-12
37. Gasda, S.E., Nordbotten, J.M., Celia, M.A.: Vertical equilibrium with sub-scale analytical methods for geological C02 sequestration. Comput. Geosci **13**(4), 469–481 (2009). doi:10.1007/s10596-009-9138-x
38. Andersen, O., Nilsen, H.M., Lie, K.-A.: Reexamining CO_2 storage capacity and utilization of the Utsira Formation. In: ECMOR XIV –14^{th} European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8-11 September 2014, EAGE (2014). doi:10.3997/2214-4609.20141809
39. Lie, K.-A., Nilsen, H.M., Andersen, O., Møyner, O.: A simulation workflow for large-scale CO_2 storage in the Norwegian North Sea. In: ECMOR XIV– 14^{th} European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8-11 September 2014, EAGE (2014). doi:10.3997/2214-4609.20141877
40. Juanes, R., Spiteri, E.J., Orr Jr., F.M., Blunt, M.J.: Impact of relative permeability hysteresis on geological CO_2 storage. Water Resour. Res. **42**, W12418. doi:10.1029/2005WR004806
41. Span, R., Wagner, W.: A new equation of state for carbon dioxide covering the fluid region from triple-point temperature to 1100 K at pressures up to 800 MPa. J. Phys. Chem. Ref. Data **25**(6), 1509–1597 (1996)

Paper V

Analysis of CO_2 Trapping Capacities and Long-Term Migration for Geological Formations in the Norwegian North Sea using MRST-co2lab

Halvor M. Nilsen, Knut-Andreas Lie, Odd A. Andersen
Computers & Geosciences, **79**, (2015)

Computers & Geosciences 79 (2015) 15–26

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Analysis of CO_2 trapping capacities and long-term migration for geological formations in the Norwegian North Sea using MRST-co2lab

Halvor Møll Nilsen, Knut-Andreas Lie *, Odd Andersen

SINTEF ICT, Department of Applied Mathematics, P.O. Box 124 Blindern, N-0314 Oslo, Norway

ARTICLE INFO

Article history:
Received 1 September 2014
Received in revised form
12 December 2014
Accepted 2 March 2015
Available online 4 March 2015

Keywords:
CO_2 storage
Sleipner and Utsira
Injection sites
Rate optimization

ABSTRACT

MRST-co2lab is a collection of open-source computational tools for modeling large-scale and long-time migration of CO_2 in conductive aquifers, combining ideas from basin modeling, computational geometry, hydrology, and reservoir simulation. Herein, we employ the methods of MRST-co2lab to study long-term CO_2 storage on the scale of hundreds of megatonnes. We consider public data sets of two aquifers from the Norwegian North Sea and use geometrical methods for identifying structural traps, percolation-type methods for identifying potential spill paths, and vertical-equilibrium methods for efficient simulation of structural, residual, and solubility trapping in a thousand-year perspective. In particular, we investigate how data resolution affects estimates of storage capacity and discuss workflows for identifying good injection sites and optimizing injection strategies.

1. Introduction

The net European CO_2 emissions from energy industries, manufacturing, and production totaled 1.95 Gt in 2011 according to the UNFCCC database.[1] Geological sequestration of a significant share of this CO_2 would require injecting hundreds of megatonnes underground annually. Altogether, the resulting storage requirement would be (at least) two orders of magnitude larger than the ongoing Sleipner injection or the planned White Rose project in the UK.

Sedimentary basins offshore Norway contain a number of saline aquifers with large volumes of pore space potentially usable for CO_2 storage. The Norwegian Petroleum Directorate has released two CO_2 Storage Atlases (Halland et al., 2011, 2012) that explore large-scale CO_2 storage for a number of aquifers. In total, twenty-seven geological formations have been grouped into aquifers whose qualities are assessed with regard to CO_2 storage potential. Similar atlases have been compiled in other parts of the world (Lewis et al., 2008; Natural Resources Canada, 2012; U.S. Department of Energy, 2012; Cloete, 2010; Bradshaw et al., 2011). See also EU GeoCapacity (2009) for a summary of sites and storage capacities for European countries.

Pressure buildup is often the main effect that limits storage capacity in closed or low-conductive aquifers, see e.g., Lindeberg et al. (2009) and Thibeau and Mucha (2011). For open aquifers with good hydraulic conductivity, leakage risk due to long-term CO_2 migration could be a larger concern and one must therefore determine the amounts of CO_2 retainable by different trapping mechanisms. In most relevant scenarios, the injected CO_2 has lower density than the surrounding formation fluid, and will form a buoyant plume that might migrate long distances below a sealing, sloping caprock. Over time, some CO_2 will be retained in structural and stratigraphic traps (structural trapping), be trapped as small droplets between rock grains (residual trapping), dissolve into the formation water (solubility trapping), or react with rock minerals and become permanently trapped (mineral trapping). Simulating the trapping processes constitutes a challenging multiscale problem that is best attacked using a range of different computational approaches.

In Nilsen et al. (2015a–c), we describe an ensemble of simulation tools that can be used to simulate likely outcomes of large-scale, long-term migration processes and estimate capacity for structural, residual, and solubility trapping. These tools are implemented using a high-level scripting language and made publicly available as a separate module (SINTEF ICT, 2014) of the open-source Matlab Reservoir Simulation Toolbox (MRST) (MRST, 2014; Lie et al., 2012). Herein, we demonstrate how these tools can be combined to provide alternative estimates of storage capacities. First, a set of simple geometrical/percolation type methods are employed to identify traps, accumulation areas, and spill paths (Nilsen et al., 2015a) to provide upper bounds on the overall capacity for structural trapping. We analyze how the resulting

* Corresponding author.
E-mail addresses: Halvor.M.Nilsen@sintef.no (H. Møll Nilsen), Knut-Andreas.Lie@sintef.no (K.-A. Lie), Odd.Andersen@sintef.no (O. Andersen).

[1] http://unfccc.int/di/FlexibleQueries.do Query parameters: Europe27, 2011, net emissions excluding land use, CO_2 only, total for categories 1.A.1 (Energy Industries) and 1.A.2 (Manufacturing Industries and Construction)

http://dx.doi.org/10.1016/j.cageo.2015.03.001
0098-3004/

estimates depend on the data resolution and then use vertical equilibrium (VE) models (Nordbotten and Celia, 2012; Gasda et al., 2009, 2011, 2012a,b, 2013; Nilsen et al., 2011; Nordbotten and Dahle, 2011; Doster et al., 2012), whose efficient implementation is discussed in Nilsen et al. (2015b,c), to provide practical and realistic estimates for representative storage scenarios in which CO_2 is injected at a finite rate. In VE models, the flow of a thin CO_2 plume is approximated in terms of its thickness underneath the top zeal of the aquifer to obtain a 2D simulation model. Although this reduces the dimension of the model, important information of the heterogeneities in the underlying 3D medium is preserved and the errors resulting from the VE assumption are in many cases significantly smaller than those seen in (overly) coarse 3D simulation models. The presented examples are based almost entirely on publicly available data sets, and complete scripts can be downloaded and used under the GNU Public License version 3.

2. Impact of model resolution

The accuracy of geological models used to represent large saline aquifers will in most cases be questionable. First of all, there is a general lack of accurate data: seismic surveys are not as dense as for petroleum reservoirs, core data from drilling are scarce, etc. Secondly, because of the large spatial areas involved, standard 3D flow simulations can typically only be performed on relatively coarse models to be computationally tractable, see discussion in Nordbotten et al. (2012). If we think of the top seal as an undulating surface, any oscillation with a wave length shorter than twice the cell size cannot be represented. Decreasing the resolution will typically remove a great number of smaller structural traps and tend to underestimate structural trapping capacity and retardation effects.

In the CO_2 Storage Atlas (Halland et al., 2011), twenty-one geological formations have been individually assessed and grouped into saline aquifers that can be considered candidates for CO_2 injection. Using information from the accompanying data sets, we were able to construct grid models and estimate structural trapping capacity for fourteen different sand volumes (Nilsen et al., 2015a). The atlas data sets cover large areas and are primarily intended for mapping. The extracted grid models are therefore comparatively coarse, with typical lateral resolutions of 500 or 1000 m. Moreover, inaccuracies are also introduced during the data integration process when constructing the simulation grids, as explained in Nilsen et al. (2015a). In this section, we will discuss the impact of model resolution in more detail for two of the fourteen aquifers considered in Nilsen et al. (2015a).

2.1. The Johansen formation

The deep saline Johansen aquifer (Akervoll and Bergmo, 2009; Bergmo et al., 2009; Eigestad et al., 2009; Sundal et al., 2013) is located below the Troll field on the Southwest coast of Norway. The aquifer has an estimated theoretical storage capacity in the range of 1–2 billion tonnes of CO_2 and was proposed as a storage site for CO_2 to be captured from gas power plants at Mongstad and Kårsø.

In Nilsen et al. (2015a), we outline how the potential for structural trapping predicted by the atlas model of Johansen deviated significantly from two low-resolution (sector and field) models previously developed for simulation purposes (Eigestad et al., 2009). Herein, we investigate the effect of model resolution by generating six grid realizations using raw data from the CO_2 Storage Atlas, see Fig. 1. The first realization is the full data set with 500 m lateral resolution, the second is coarsened by a factor two in each lateral direction, the third by a factor three, and so on. All grids are fairly coarse compared to typical simulation grids. All major traps appear inside the domain and hence the estimates of trapping are not significantly affected by how the computational algorithm determines traps against the perimeter, which here is assumed to be open.

Fig. 2 and Table 1 show the result of an analysis of the structural trapping capacity for the six different grids. Note that several smaller traps are removed as the coarsening increases, which can be observed statistically by noting that the mean of the trap volume quickly increases as the smaller traps are smoothed away. Total trapping volume also changes as we coarsen the model: first, the volume increases as the largest traps become slightly larger due to the lower resolution. However, as the resolution further decreases, the total volume shrinks as smaller traps are removed entirely.

The general effect of coarsening is well studied within reservoir simulation, but the problem of CO_2 injection is special in that it contains a light CO_2 phase trapped above a nearly immobile water or hydrocarbon phase. Small-scale traps not only increase the volume available for structural trapping, but can significantly divert spill paths and retard plume migration and hence increase the volumes that can be injected and safely contained within the boundaries of an aquifer (Nilsen et al., 2012; Syversveen et al., 2012; Gasda et al., 2012b). The geometry of the top surface can therefore in many cases have an effect on the total amount of trapped CO_2 that is larger than the effect of fine-scale permeability variations. In Nilsen et al. (2015b), we outlined several effective models that account for the retardation effect from unresolved caprock undulations. Data that can be used to estimate small-scale undulations in the top seal will usually not be available. In the next subsection, whoever, we will discuss a case in which such data *is* available.

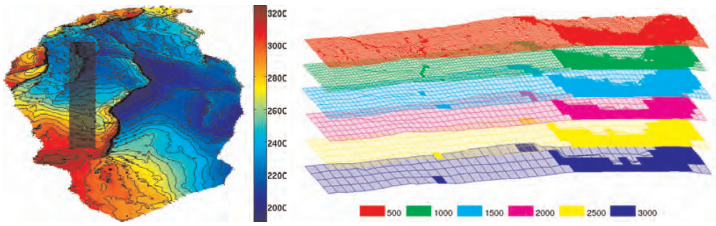


Fig. 1. Models of the Johansen aquifer from the CO_2 Storage Atlas covering an area of 45.8×101.2 km^2. The left plot shows the depth map in meter for the model with full lateral resolution of 500 m. The plot to the right shows the top surface for the 5×50 km^2 subregion marked in gray in the left plot for six different lateral resolutions. Cells that are inside traps are marked in solid color.

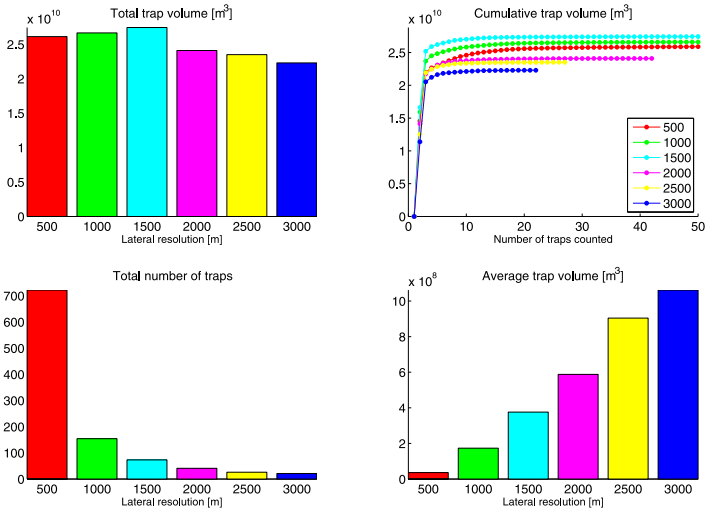


Fig. 2. Impact of grid resolution on structural trapping capacity for the Johansen Formation. For the cumulative plot, the traps have been sorted by volume in descending order.

2.2. The Utsira formation

The world's first commercial CO_2 storage project started in 1996 at the Sleipner West field on the Norwegian Continental Shelf. Here, CO_2 that is a byproduct from natural gas production is injected into the neighboring Utsira Formation. The annual injection rate is approximately one metric megatonne of CO_2 per year. Six seismic surveys acquired between 1999 and 2008 indicate how the injected CO_2 has accumulated under a series of nine geological horizons (Boait et al., 2012), and there is no evidence so far of CO_2 leaking out of the formation.

Here, we will use geological models that have been developed for the ninth and upper horizon in the area around the injection site to illustrate and estimate the impact that data resolution has on the estimated trapping capacity for the whole Utsira Formation. These models are:

- IEAGHG: Sleipner model provided by IEAGHG (International Energy Agency, 2012), 50 m resolution;
- GHGT: in-house model used in Nilsen et al. (2011), same resolution as the IEAGHG model but covers a larger region;
- INHOUSE: in-house model from Statoil which covers the same region as the IEAGHG model, but with a 12.5 m resolution.

Comparing the structural trapping capacities derived from these models with the corresponding figure derived from the coarser (500 m resolution) Utsira model constructed from atlas data, we can estimate the amount of fine-scale structural trapping capacity that is not captured by the latter model.

Fig. 3 compares the IEAGHG model with the corresponding region from the atlas model. What is seen as a smooth surface without any local maximum in the coarse atlas model will in the IEAGHG model contain a large number of structural traps of varying sizes. Altogether, the IEAGHG model predicts an average small-scale bulk trap volume of 560 l/m^2. However, this simple analysis does not separate between truly sub-scale traps and those large enough to be reflected in the atlas grid. Indeed, part of the structural capacity described by the detailed model could still be represented by the coarser model. To avoid double-counting when estimating the amount of structural capacity only present in the detailed model, we base the analysis on the *difference* surface, obtained by subtracting the height variations of the atlas model from the IEAGHG model as shown to the left in Fig. 4. We thereby obtain the fine-scale surface that only represents relative depth variations not resolved in the atlas model. This gives an average of 210 l of unresolved trapping volume per m^2. Assuming that these local undulations are representative for the rest of the Utsira model, we predict the additional geometric trapping capacity from the fine-scale grid to be 5.1×10^9 m^3 across the model, which amounts to 30.5% of the bulk volume of traps estimated directly from the original model.

The impact of the additional fine-scale structure relative to a flat, horizontal surface will likely exaggerate the local trapping effects; the volume of local traps obviously becomes smaller if the residual surface is imposed on an inclined, planar surface. To assess this effect, we give the residual surface a global tilt and compute the average volume of unresolved traps per surface area as a function of the angle and direction of inclination. The resulting function is shown to the right in Fig. 4. We notice that the function is not symmetric, and that the maximum amount of trapping occurs for a nonzero tilt. This can likely be attributed to an inconsistency in the overall inclination angle of the high and low-resolution models. To compensate, we shift the maximum point to the origin. By assuming that the amount of sub-resolution trapping does not vary much across the formation, this function, which we refer to as the *sub-scale trapping function*, can be interpreted as an estimate of sub-scale trapping capacity per surface area as a function of local tilt and be used as input to the effective models outlined in Nilsen et al. (2015b).

The unresolved small-scale trapping potential for the whole Utsira Formation can now be estimated as follows: for each cell in the atlas model, we calculate the local tilt angle and direction, and

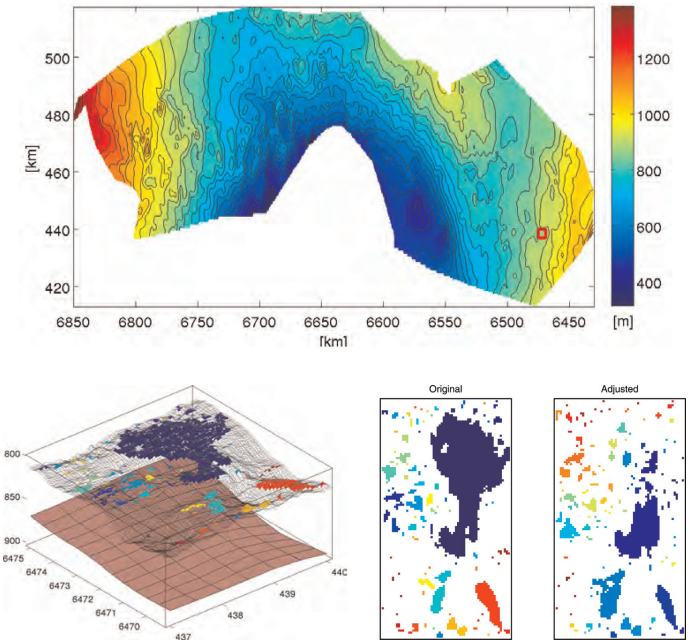


Fig. 3. Estimation of sub-resolution trapping in the Utsira Formation. The first plot shows the depth map of the Utsira atlas model with the IEAGHG model of Sleipner marked as a small red box. The next plot shows the IEAGHG surface (50 m) with color-coded traps plotted above the low-resolution Utsira model (500 m) of the same area. The last plots show structural traps on the original IEAGHG model (left) and on a version in which the height variations of the low-resolution Utsira model have been eliminated (right). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

determine the corresponding unresolved trapping capacity by evaluating the sub-scale trapping function and multiplying by cell area. Doing this for all cells in the Utsira model and summing up, we obtain a global estimate of sub-scale trapping. We have constructed sub-scale trapping functions based on residual surfaces obtained by subtracting the atlas grid from the aforementioned IEAGHG, GHGT, and INHOUSE data sets. In addition, we constructed a sub-scale trapping function using a version of the IEAGHG top surface in which all details with resolution below 500 m were removed by smoothing with a Gaussian kernel. By computing the residual surface (and sub-scale trapping function) from one data set only, we obtain a trapping function that is not affected by inconsistencies between different data sets.

The results assuming a constant or slope-dependent amount of subscale trapping for each data set are presented in Table 2. From these figures, we estimate that the amount of sub-scale trapping not resolved in the atlas model is in the range of 13–20%. The large difference between the constant and the slope-dependent estimates shows that it is important to properly account for the interplay between the local slope and the steepness of the small-scale undulations in the caprock. How this interplay impacts upscaled relative permeabilities is discussed in detail in Gasda et al. (2013).

3. Sleipner: upscaled injection operation

We now use the Utsira aquifer model obtained from the CO_2 Storage Atlas to investigate the long-term fate of CO_2 for a hypothetical upsized operation at Sleipner. We consider an injection rate of 10 Mt per year (approximately ten times the actual injection rate), for an injection period of 50 years, followed by a 3000 year migration period. We run three simulations, which all include residual trapping, structural trapping, and sub-scale trapping (as estimated from the combined Sleipner and Utsira data sets in Section 2.2), but which differ in their treatment of solubility trapping. The first simulation does not include dissolution effects, the second simulation considers dissolution to be instantaneous in any vertical column where CO_2 is present, and the third simulation models a constant *rate* of dissolution, as explained in Nilsen et al. (2015b) and Gasda et al. (2011). Solubility of CO_2 in brine is assumed to be 53 kg/m^3 for the two latter simulations (taken from Chadwick et al., 2008), and the dissolution rate in the third simulation is set to 0.44 kg/m^2 per year. The simulations are performed using a fully implicit VE simulator based on a sharp interface model, implemented using automatic differentiation. CO_2 density and viscosity values are functions of local pressure and temperature, computed using (Bell et al., 2014). Linear compressibility is assumed for rock (10^{-5} bar^{-1}) and brine (4.3×10^{-5} bar^{-1}). Residual saturation for brine and CO_2 is respectively set to 0.11 and 0.21, as suggested in Singh et al. (2010). We consider a uniform rock porosity of 21.1%, as inferred from Halland et al. (2011).

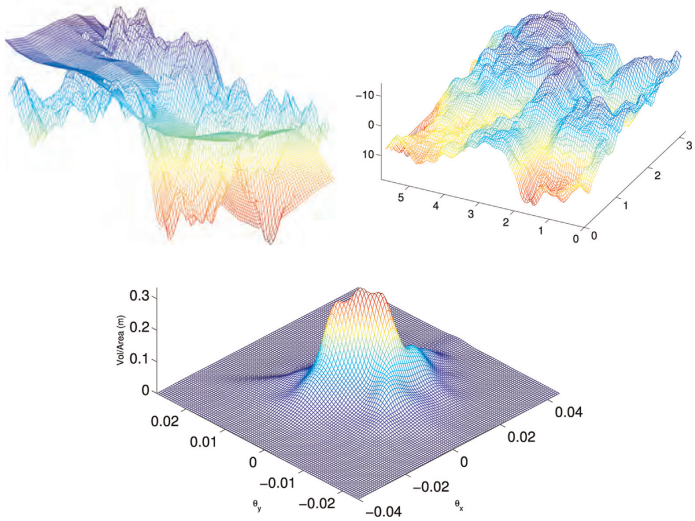


Fig. 4. Estimation of small-scale trap volumes not resolved in the atlas model. *Top-left*: fine-scale interpolant from atlas data imposed on top of the IEAGHG model (scales exaggerated vertically), see also lower-left plot in Fig. 3. *Top-right*: the corresponding residual surface. *Bottom:* average volume of sub-resolution traps per unit area as a function of the average tilt angle of the coarse surface.

Table 1
Impact of grid resolution on structural trapping for the Johansen formation.

Resolution (m)	# traps	Bulk volume (m^3)	Avg. volume (m^3)
500	722	2.61e+10	3.62e+07
1000	154	2.67e+10	1.73e+08
1500	73	2.75e+10	3.76e+08
2000	41	2.41e+10	5.88e+08
2500	26	2.35e+10	9.04e+08
3000	21	2.23e+10	1.06e+09

Table 2
Estimates of unresolved small-scale geometrical trapping as a percentage of resolved geometrical trapping for the atlas model of Utsira. Each column represents the result of using a particular estimate of the trapping function, based on the difference between the high-resolution and low-resolution surfaces indicated in the header. The slope-dependent estimate is computed taking local tilt into account, whereas the other estimate is obtained by assuming a constant tilt everywhere that maximizes sub-scale trapping for the local fine-scale model.

High-resolution Low-resolution	IEAGHG Utsira	GHGT Utsira	INHOUSE Utsira	IEAGHG IEAGHG[1]
Slope-dependent (%)	17.7	19.5	17.2	13.6
Constant (%)	55.3	59.4	55.9	45.3

[1] smoothed version of grid.

3.1. Initial analysis of structural trapping potential

We first assess how much CO_2 we can expect to store in structural traps from the Sleipner injection point. Fig. 5 presents a map of structural traps at Utsira, with the location of the Sleipner injection indicated. Altogether, these traps represent a combined pore volume of 3.55 km^3, which is 0.44% of the total pore volume of the aquifer model. Previous studies have arrived at roughly

Fig. 5. Map over structural traps and connecting spill paths for the Utsira Formation.

comparable figures. Chadwick et al. (2004) estimate a total of 1.8 km^3 of pore space within structural closures of Utsira, extrapolated from analysis of 3D seismic data around the Sleipner area, whereas Bøe et al. (2002) arrive at a value of 1.10 km^3, based on

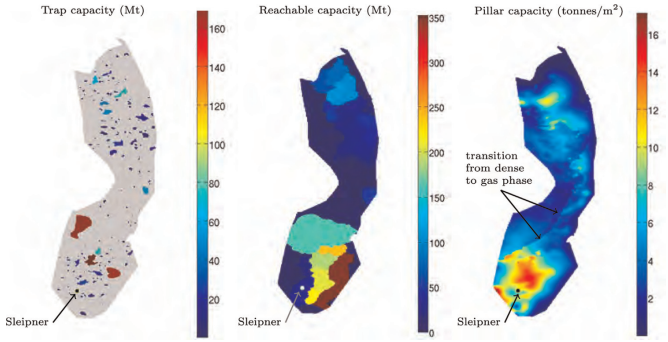


Fig. 6. Analysis of trapping capacity for the Utsira Formation. *Left*: individual structural traps, color-coded by estimated CO_2 mass trapping capacity. *Middle*: combined capacity of upstream traps that can be reached along a spill path from each point in the aquifer. The total structural capacity of the aquifer is 1.13 Gt (Andersen et al., 2014). *Right*: total theoretical CO_2 retaining capacity per square meter, taking all trapping mechanisms (structural, residual, solubility) into account. The total containing capacity of the whole aquifer is 112 Gt (Andersen et al., 2014). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

multiplication of certain assumed ratios. The left plot of Fig. 6 indicates the capacity of each identified structural trap in terms of CO_2 mass. The *cumulative* structural trapping along the length of the spill path from an injection point and to the top of the formation is presented in the middle plot. According to this figure, the cumulative trap capacity reachable from the Sleipner site is close to 30 Mt. On the other hand, the injection point is located close to a region with much higher reachable capacity (215 Mt). Since the real flow of CO_2 will be far from infinitesimal, and initially driven primarily by viscous forces, we expect that a considerable amount of CO_2 will end up there as well. If we add up the figures for the two regions, we conclude that we might to reach up to 245 Mt of structural capacity from the Sleipner injection point (not counting subscale trapping), or 49% of the total injected CO_2 during the operation. However, if we also consider the other trapping mechanisms (residual, solubility), we can hope to retain a significantly higher amount. The right plot of Fig. 6 is a map of the estimated total trapping capacity per lateral square meter of the aquifer, taking structural, residual, and solubility trapping into account. This estimate is highly theoretical, since it considers an (unobtainable) one hundred percent sweep efficiency and complete saturation of dissolved CO_2 in formation water. However, the plot still indicates that the Sleipner injection point is located close to a region that is able to hold large amounts of CO_2 by a combination of trapping mechanisms. A detailed discussion of the assumptions and parameters behind the construction of this figure is presented in Andersen et al. (2014).

3.2. Simulation results

The outcomes of the three simulations are presented in Figs. 7–9. For each simulation, three snapshots of the CO_2 distribution are presented, corresponding to year 50 (injection end), year 1085, and year 3050. We also present an inventory that details the various forms in which CO_2 is trapped as a function of time. The diagram uses a color coding that goes from dark green for volumes that are safely trapped to red for volumes that have escaped across the perimeter of the domain. In the diagram, and in the following discussion, we use the term (*movable*) *CO_2 plume* for regions in which the CO_2 saturation is higher than the residual saturation. When the CO_2 is present at residual or lower saturation, it is considered to be residually trapped. To distinguish volumes that can move and volumes that cannot, the mobile CO_2 plume is divided into a free part that may continue to propagate in the upslope direction and a residual part that is destined to be left behind after imbibition as the movable plume propagates upslope. Likewise, the structurally trapped volumes are divided into a free part that may potentially leak through imperfections in the

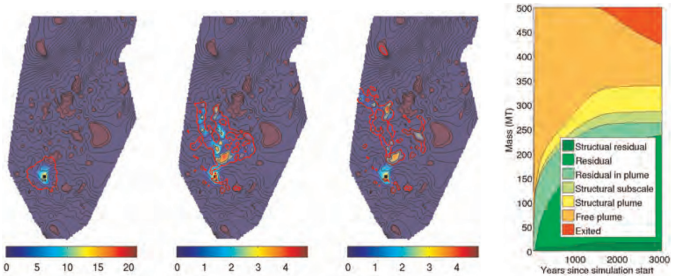


Fig. 7. Simulation of single-point injection into the Utsira aquifer without dissolution effects. Snapshots of CO_2 distribution after 50, 1085, and 3050 years and diagram presenting historical trapping distribution. Outline of the remaining movable plume is traced in red, and overall CO_2 content of each vertical column indicated with color (unit: tonnes per lateral square meter). Structural traps are indicated in purple. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

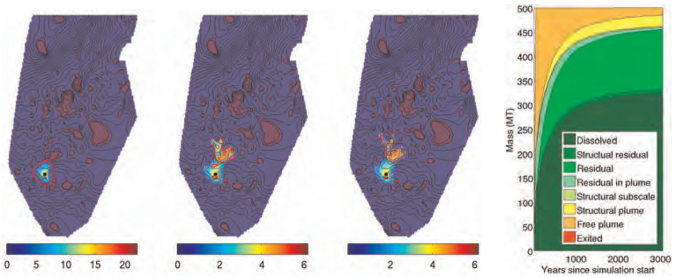


Fig. 8. Simulation of single-point injection into the Utsira aquifer with instantaneous dissolution. Snapshots of CO_2 distribution after 50, 1085, and 3050 years. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

caprock and a residual part that is permanently immobilized and cannot escape the trap. The inventory also accounts for volumes that are immobilized within small traps not resolved by the simulation grid. Fig. 10 provides a conceptual illustration and Table 3 summarizes the terminology.

In Fig. 7, we note that at the end of simulation, approximately 75 Mt (15%) of the injected CO_2 has leaked from the domain. The onset of this leakage occurs after approximately 1400 years. After this period, little additional CO_2 becomes structurally or residually trapped, since flow is now predominantly along established pathways. This also means that most of the remaining 280 Mt of free CO_2 is likely to leak in the future. We note that at the end of simulation, approximately 70 Mt of CO_2 has become structurally trapped at the macro scale ("Structural residual" and "structural plume"). This means that in addition to the 30 Mt of reachable structural capacity identified in our initial analysis, approximately 40 Mt of CO_2 has been structurally trapped in the neighboring spill system, with an additional 25 Mt of CO_2 trapped in subscale traps ("structural subscale") that are not resolved on the grid. Altogether, the injection utilizes approximately 6% of the structural capacity of the resolved traps and 11–16% of the subscale trapping capacity estimated in Table 2. The reason these numbers are different is that the plume only connects a specific subset of all the structural traps, whereas subscale trapping will take place in all the area contacted by the plume. By and large, though, the most important trapping mechanism here is residual trapping, which accounts for roughly half of all injected CO_2 at the end of the simulation period.

In Figs. 8 and 9, we can see that solubility trapping quickly ends up dominating all other trapping mechanisms in the two simulations that include this effect. However, there are significant differences in end results between these two models. In the case of instant dissolution (Fig. 8), the full impact of dissolution is present from the start. As a result, the movable CO_2 plume never grows large, and as it moves it is quickly dissolved and does not migrate far. On the other hand, since the brine below the plume is saturated with CO_2 at all times, no additional dissolution occurs in areas where the plume remains present. On the trapping distribution diagram, we thus see a notably reduced growth of the dissolved component after some 1000 years, as migration gradually stagnates.

The effect of rate-dependent dissolution, as presented in Fig. 9, is notably different. Here, dissolution does not become the dominant effect until after approximately a thousand years, meaning that the CO_2 manages to spread much farther. As a result of the larger spatial extent of plume migration, a larger amount brine is exposed to CO_2 here than in the instantaneous dissolution model, allowing for a larger total amount of CO_2 to be dissolved before saturation is reached. Moreover, we observe that the residual saturation (green) never grows much despite significant plume migration, as it is constantly depleted due to the ongoing dissolution.

4. Utsira: large-scale industrial injection

The amount of CO_2 theoretically retainable in the Utsira Formation by structural and residual trapping is estimated to be orders of magnitude above the amounts currently injected at Sleipner (Andersen et al., 2014). In the present example, we simulate a large-scale operation in which up to 1.5 Gt of CO_2 is

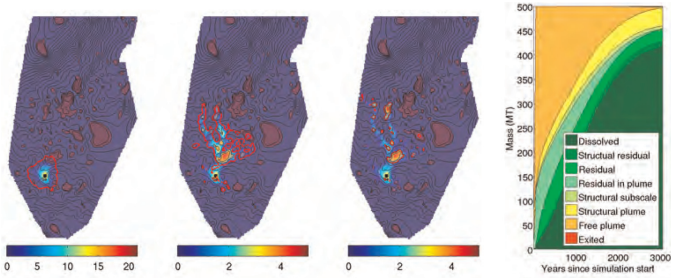


Fig. 9. Simulation of single-point injection into the Utsira aquifer with rate-dependent dissolution. Snapshots of CO_2 distribution after 50, 1085, and 3050 years. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

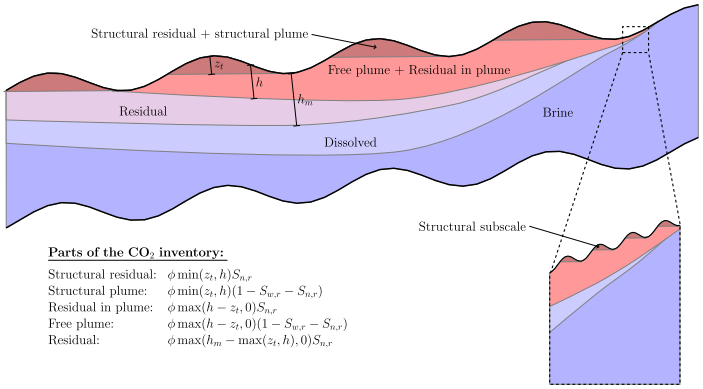


Fig. 10. Schematic of a vertical section of an aquifer identifying the various trapping states mentioned in Table 3. Here, ϕ is the pore volume, $S_{n,r}$ is the residual CO_2 saturation, $S_{w,r}$ is the irreducible water saturation, z_t is the depth of the trap (or zero if there is not trap), h is the depth of the movable plume, and h_m is the largest depth at which CO_2 has been observed. The right inlet visualizes the small-scale undulations in the top surface that are not resolved by the large-scale model, thus accounted for as "subscale trapping" in our inventory.

Table 3
Explanation of terminology used in the CO_2 inventory of Figs. 7–9.

State	Explanation
Dissolved	CO_2 trapped by dissolution into formation brine
Structural residual	CO_2 that is both structurally and residually trapped
Residual	Residually trapped CO_2 outside free plume and structural traps
Residual in plume	CO_2 still in the free-flowing plume, but destined to be left behind after imbibition
Structural subscale	CO_2 trapped in caprock structures too small to be represented by the grid
Structural plume	structurally but not residually trapped CO_2
Free plume	CO_2 that is still free to migrate (i.e. part of the plume that is neither residually nor structurally trapped)
Exited	CO_2 that has left the simulated domain

injected by means of ten separate injection sites over a fifty year period, after which we track 3000 years of migration. Injector locations are chosen to maximize utilization of available structural trapping, with optimal injection rates subsequently determined using a nonlinear optimization approach. For the simulation, we use the same fully implicit numerical model and parameters as in Section 3. We include structural, residual, and subscale trapping, but not the effect of dissolution.

4.1. Choosing injection locations

The Utsira Formation being a high-permeability, open aquifer, we assume pressure buildup to be of minor concern, and choose injection sites solely based on reachable structural trapping, using the rapid, greedy algorithm described in Nilsen et al. (2015a). Within target catchment areas, injection locations are chosen as far away as possible from the formation boundary. The result is presented in Fig. 11, with wells numbered according to the order they were chosen by the algorithm. We note two primary clusters of wells, one in the north and one in the south. Only a single well is located in the narrow middle region, which provides only a small amount of structural trapping, as apparent from Fig. 6.

Fig. 11. Choice of injection sites for large-scale utilization of the Utsira aquifer.

4.2. Setting injection rates

Since reachable structural capacity for each injection site has already been identified, division by total injection period gives an initial suggestion of injection rates. This estimate does not take residual trapping into account, nor does it acknowledge that CO_2 may spill out of the intended regions during injection and migration. More optimal injection rates can be obtained by taking these effects into account. To this end, we use a nonlinear optimization approach, made practical by our ability to run multiple,

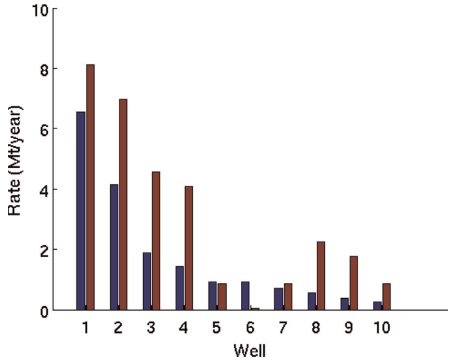


Fig. 12. Initial (blue) and optimized (red) injection rates. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

rapid simulation using the VE framework (Nilsen et al., 2015b). For the optimization, we define an objective function that equals the combined total amount of CO_2 injected, minus the amount of CO_2 leaked by the end of the simulation period weighted by a factor ten to strongly penalize injection of volumes that will leak back out. To compute the value of the objective function for a given set of rates, a full VE simulation is carried out. The gradient of the objective function can then be obtained by an adjoint method (Raynaud et al., 2014; Jansen, 2011), and the optimization problem is solved iteratively with gradient-based method. As a starting point for the optimization algorithm, we use injection rates obtained from the estimates of reachable structural capacity, as described above. We will refer to these rates as the "initial rates", and those obtained from optimization as the "optimized rates". These are presented in Fig. 12. As can be seen from this figure, injection rates for most sites are adjusted significantly upwards after optimization, primarily attributable to the additional effect of residual trapping. On the other hand, the injection rate of site 6 has been adjusted down to almost zero. This is further discussed below.

4.3. Simulation results

We first consider the scenario using the initial (unoptimized) injection rates. The result is presented in the left plot of Fig. 13 (trapping distribution over time) and the upper row of Fig. 14, presented similarly to Section 3. The total amount injected in this scenario is 887 Mt, which equals the estimated reachable trapping capacity from the injection sites. As can be seen from Fig. 13, less than half of the injected CO_2 ends up in macro-scale structural traps ("structural residual" and "structural plume"), since significant amounts are retained by residual and subscale trapping. The leakage at the end of the simulation period is negligible, but approximately 10% of the CO_2 remains mobile and can potentially leak in the future. The simulation snapshots in Fig. 14 illustrate how the CO_2 migration is generally upwards towards the western boundary, with accumulation in encountered traps. Notably, a large part of the CO_2 injected from site 6 does not end up in the neighbor trap, but flows upwards along the steep slope towards the west.

The right plot of Fig. 13 presents the trapping distribution over time for the simulation with optimized injection rates. In this simulation, a total of 1.52 Gt has been injected, still with negligible leakage at the end of the simulation period. The relative importance of the different trapping mechanisms remain comparable to the unoptimized case, but the relative amount of mobile CO_2 after 3000 years is approximately twice as large as before (approximately 300 Mt). A longer simulation period would allow a large part of this amount to eventually exit the domain. However, the objective function used in the optimization algorithm does not account for future developments, and does not penalize presence of mobile CO_2. What is considered "optimal" injection rates will therefore depend both on how much leakage is tolerated and on the considered time span. The corresponding simulation snapshots are presented in the lower row of Fig. 14. We note that the migrating plumes have formed some established pathways towards the formation boundaries, where future leakage will take place. We also note that the amount injected from site 6 has been drastically reduced. As such, the amount of leaked CO_2 from this site has been reduced to a minimum, but the neighbor trap has ended up under-utilized. This is due to the placement of the injection site within the trap catchment area. As previously mentioned, the algorithm used to choose well sites favors positions at the edges of the targeted catchment area in order to maximize distance to the outer boundary. As a consequence, since real flow is neither infinitesimal nor purely gravity driven, some amount of CO_2 will spill out of the intended region. For site 6 in our example, this means that a large quantity of CO_2 takes the alternative path westwards rather than flowing into the intended trap. Since leakage is heavily penalized by our objective function, the result is

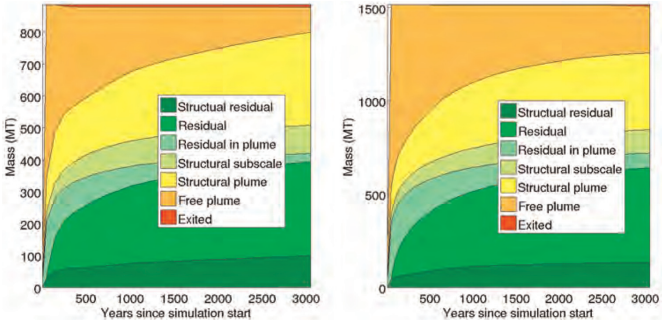


Fig. 13. Historical trapping distribution of CO_2 for the scenarios presented in Section 4. The left diagram presents the history for the unoptimized case and the right diagram for the optimized case.

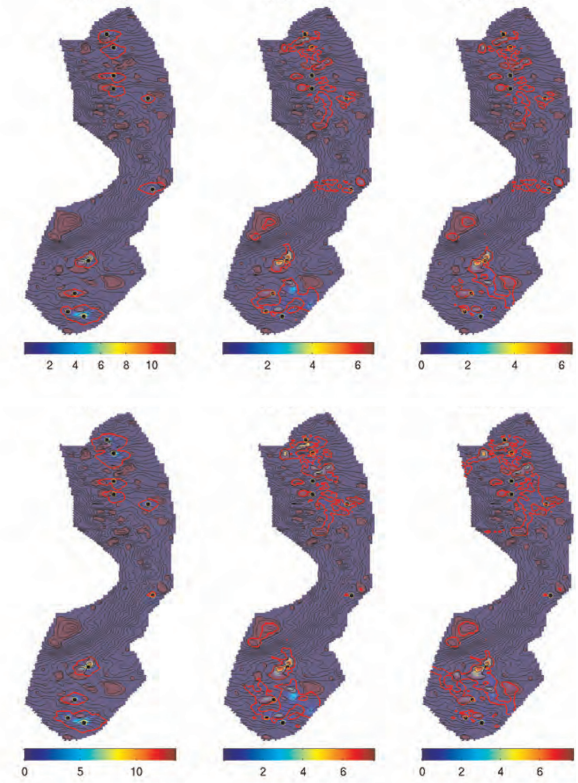


Fig. 14. Simulation snapshots of the CO_2 distribution after 50, 1050, and 3000 years for the two injection scenarios presented in Section 4. The upper row presents the initial case with injection rates determined without flow simulation, while the lower row presents the optimized case. Outline of the remaining movable CO_2 plume is traced in red, and overall CO_2 content of each vertical column indicated with color (unit: tonnes per lateral square meter). Structural traps indicated in purple. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

that injection at this site is drastically scaled down. A more sophisticated algorithm for site selection would take into account the estimated radial extent of the plume after injection, and position the injection point far enough within the intended catchment area to minimize spill along unintended pathways.

As mentioned, the selection of sites are based on optimizing reachable structural capacity, assuming that pressure buildup is not a critical issue. To verify this assumption, we end our analysis by examining the overpressure attained during injection for the scenario with optimized injection rates, defined as the difference between initial (fluid-static) pressure and maximum pressure observed during the vertical-equilibrium simulation. We find that overpressure is highest at the earliest simulated time step, where it reaches 2.53 MPa. As can be seen in Fig. 15, this happens around the northernmost injection site, which is located in the deeper and thinner end of the formation. By assuming a lithostatic pressure gradient of 17 MPa/km (Singh et al., 2010), we estimate the Utsira overburden pressure to range from 4.8 MPa to 23 MPa depending on depth. We therefore conclude that in our simulated scenario, the overpressure from injection will at all times remain well below the overburden pressure. This conclusion would, however, strongly depend on the assumptions made about the aquifer boundaries, which were considered fully open. If we reduce the transmissibility of the aquifer boundary to mimic a situation where fluids expelled from the simulation domain have to pass through another 100 km of sandstone before reaching a hydrostatic pressure domain, our simulation produces a significantly higher overpressure of 4.12 MPa, now reached towards the end of the injection period. Similarly, running the same injection scenario with the assumption of fully closed boundaries yields an overpressure of 6.75 MPa. It is clear that any simulation of large-scale CO_2 storage at Utsira that intends to produce real figures on pressure buildup would require a valid model of the larger aquifer surroundings, and its effect on lateral fluid flow across aquifer boundaries as well as pressure-induced diffuse leakage of brine through the caprock.

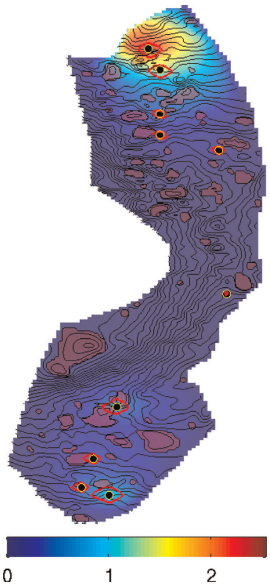


Fig. 15. Overpressure at early injection stage for scenario with optimized injection rates for the Utsira aquifer (unit: MPa).

5. Concluding remarks

In this paper we have applied tools available in MRST-co2lab to process and analyze data sets available from the CO_2 atlases published by the Norwegian Petroleum Directorate. Our analyses include estimates on structural trapping potential, VE simulations to track CO_2 state and plume development, potential impact of subscale features and of dissolution, identification of good injection sites, and optimization of injection rates.

The use of VE models produces rapid results relative to the size of the targeted simulation problems. The low computational demand of such models is in part due to the reduced dimensionality of the simulated domain (from 3D to 2D) and in part due to weaker coupling of physical flow mechanisms. Although the reduction in dimensionality represents a significant simplification of reality, VE models still provide good results in many cases, and can take into account a range of physical phenomena. In the examples above, this includes hysteresis, subscale caprock variations, dissolution, and variable CO_2 density and viscosity as function of local temperature and pressure.

We strongly believe that rapid modeling tools can play an important role when evaluating capacity and performance of potential CO_2 storage sites, for designing usage scenarios, and for interpreting monitoring data after a site has become operative. The large scales involved and the limited availability and resolution of data means that obtaining definite forecasts from numerical simulations will likely remain an impossible task. However, the availability of rapid modeling tools makes it possible to extensively explore the unknown parameter space and thereby develop a good understanding of the various ways a given scenario *might* evolve, and identify scenario-specific important factors. The availability of rapid simulation tools also enables practical optimization of operational as well as physical parameters.

MRST-co2lab is published as an open-source module within MRST (MRST, 2014; Lie et al., 2012). Recognizing the importance of reproducible science, we have also made the numerical code behind the examples in this paper and its three predecessors publicly available (SINTEF ICT, 2014). The emphasis in MRST-co2lab is on rapid prototyping and exploration of ideas, and we believe it will support further advancement in the field both by presenting helpful tools for analyzing data sets, as well as providing a framework that allows researchers to quickly implement and test their ideas in terms of working code.

Acknowledgments

The work was funded in part by Statoil ASA and the Research Council of Norway through Grant nos. 199878 (Numerical CO_2 laboratory) and 215641 (MatMoRA-II).

References

Akervoll, I., Bergmo, P., 2009. A study of Johansen formation located offshore Mongstad as a candidate for permanent CO_2 storage. In: European Conference on CCS Research, Development and Demonstration, 10–11 February 2009, Oslo, Norway.

Andersen, O., Nilsen, H.M., Lie, K.-A., 2014. Reexamining CO_2 storage capacity and utilization of the Utsira Formation. In: ECMOR XIV – 14th European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8–11 September 2014, EAGE. http://dx.doi.org/10.3997/2214-4609.20141809.

Bell, I.H., Wronski, J., Quoilin, S., Lemort, V., 2014. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library CoolProp. Ind. Eng. Chem. Res. 53 (6), 2498–2508. http://dx.doi.org/10.1021/ie4033999.

Bergmo, P.E.S., Lindeberg, E., Riis, F., Johansen, W.T., 2009. Exploring geological storage sites for CO_2 from Norwegian gas power plants: Johansen formation. Energy Procedia 1 (1), 2945–2952. http://dx.doi.org/10.1016/j.egypro.2009.02.070.

Boait, F.C., White, N.J., Bickle, M.J., Chadwick, R.A., Neufeld, J.A., Huppert, H.E., 2012. Spatial and temporal evolution of injected CO_2 at the Sleipner Field, North Sea, J. Geophys. Res. B 117 (B3). http://dx.doi.org/10.1029/2011JB008603.

Bøe, R., Magnus, C., Osmundsen, P.T., Rindstad, B.I., 2002. CO_2 Point Sources and Subsurface Storage Capacities for CO_2 in Aquifers in Norway. Geological Survey of Norway, NGU Report 2002.10. URL: ⟨http:www.ngu.no/FileArchive/101/2002_010_skjerm.pdf⟩.

Bradshaw, B.E., Spencer, L.K., Lahtinen, A.-L., Khider, K., Ryan, D.J., Colwell, J.B., Chirinos, A., Bradshaw, J., Draper, J.J., Hodgkinson, J., McKillop, M., 2011. An assessment of Queensland's CO_2 geological storage prospectivity-The Queensland CO_2 geological storage atlas. Energy Procedia 4 (0), 4583–4590. http://dx.doi.org/10.1016/j.egypro.2011.02.417.

Chadwick, R., Zweigel, P., Gregersen, U., Kirby, G., Holloway, S., Johannessen, P., 2004. Geological reservoir characterization of a CO_2 storage site, the Utsira sand, Sleipner, northern North Sea. Energy 29 (9–10), 1371–1381. http://dx.doi.org/10.1016/j.energy.2004.03.071.

Chadwick, A., Arts, R., Bernstone, C., May, F., Thibeau, S., Zweigel, P., 2008. Best practice for the storage of CO_2 in saline aquifers – Observations and guidelines from the SACS and CO2STORE projects, Vol. 14 of British Geological Survey Occasional Publication, British Geological Survey, Nottingham, UK. URL ⟨http://nora.nerc.ac.uk/2959/⟩.

Cloete, M., 2010. Atlas on Geological Storage of Carbon Dioxide in South Africa. Technical Report. Council for Geoscience, Johannesburg, South Africa. URL ⟨http://www.sacccs.org.za/wp-content/uploads/2010/11/Atlas.pdf⟩.

Doster, F., Nordbotten, J.M., Celia, M.A., 2012. Hysteretic upscaled constitutive relationships for vertically integrated porous media flow. Comput. Vis. Sci. 15, 147–161. http://dx.doi.org/10.1007/s00791-013-0206-3.

Eigestad, G., Dahle, H., Hellevang, B., Riis, F., Johansen, W., Øian, E., 2009. Geological modeling and simulation of CO_2 injection in the Johansen formation. Comput. Geosci. 13 (4), 435–450. http://dx.doi.org/10.1007/s10596-009-9153-y.

EU GeoCapacity, Assessing European Capacity for Geological Storage of Carbon Dioxide, WP2 report: Storage capacity, 2009. URL ⟨http://www.geology.cz/geocapacity/publications⟩.

Gasda, S.E., Nordbotten, J.M., Celia, M.A., 2009. Vertical equilibrium with sub-scale analytical methods for geological Co_2 sequestration. Comput. Geosci. 13 (4), 469–481. http://dx.doi.org/10.1007/s10596-009-9138-x.

Gasda, S.E., Nordbotten, J.M., Celia, M.A., 2011. Vertically-averaged approaches to CO_2 injection with solubility trapping. Water Resour. Res. 47, W05528. http://dx.doi.org/10.1029/2010WR009075.

Gasda, S.E., Nordbotten, J.M., Celia, M.A., 2012a. Application of simplified models to CO_2 migration and immobilization in large-scale geological systems. Int. J.

Greenh. Gas Control 9, 72–84. http://dx.doi.org/10.1016/j.ijggc.2012.03.001.
Gasda, S.E., Nilsen, H.M., Dahle, H.K., Gray, W.G., 2012. Effective models for CO_2 migration in geological systems with varying topography, Water Resour. Res. 48 (10). http://dx.doi.org/10.1029/2012WR012264.
Gasda, S.E., Nilsen, H.M., Dahle, H.K., 2013. Impact of structural heterogeneity on upscaled models for large-scale CO_2 migration and trapping in saline aquifers. Adv. Water Resour. 62 (Part C (0)), 520–532. http://dx.doi.org/10.1016/j.advwatres.2013.05.003.
Halland, E.K., Johansen, W.T., Riis, F. (Eds.), 2011. CO_2 Storage Atlas: Norwegian North Sea, Norwegian Petroleum Directorate, P.O. Box 600, NO–4003 Stavanger, Norway. URL ⟨http://www.npd.no/no/Publikasjoner/Rapporter/CO2-lagringsatlas/⟩.
Halland, E.K., Johansen, W.T., Riis, F. (Eds.), 2012. CO_2 Storage Atlas: Norwegian Sea, Norwegian Petroleum Directorate, P.O. Box 600, NO-4003 Stavanger, Norway. URL ⟨http://www.npd.no/en/Publications/Reports/CO2-storage-altas-Norwegian-Sea/⟩.
International Energy Agency, Sleipner Benchmark Model, 2012. URL ⟨http://www.ieaghg.org/index.php?/2009112025/modelling-network.html⟩.
Jansen, J.D., 2011. Adjoint-based optimization of multi-phase flow through porous media – a review. Comput. Fluids 46 (SI (1)), 40–51. http://dx.doi.org/10.1016/j.compfluid.2010.09.039.
Lewis, D., et al., 2008. Assessment of the Potential for Geological Storage of Carbon Dioxide for the Island of Ireland. Technical Report. Sustainable Energy Ireland, Environmental Protection Agency, Geological Survey of Northern Ireland, and Geological Survey of Ireland.
Lie, K.-A., Krogstad, S., Ligaarden, I.S., Natvig, J.R., Nilsen, H.M., Skaflestad, B., 2012. Open source MATLAB implementation of consistent discretisations on complex grids. Comput. Geosci. 16, 297–322. http://dx.doi.org/10.1007/s10596-011-9244-4.
Lindeberg, E., Vuillaume, J.-F., Ghaderi, A., 2009. Determination of the CO_2 storage capacity of the utsira formation. Energy Procedia 1 (1), 2777–2784. http://dx.doi.org/10.1016/j.egypro.2009.02.049.
The MATLAB Reservoir Simulation Toolbox, Version 2014. ⟨http://www.sintef.no/MRST/⟩. May 2014.
Natural Resources Canada, Mexican Ministry of Energy, and U.S. Department of Energy, The North American Carbon Storage Atlas, 2012. URL ⟨http://www.nacsap.org/⟩.
Nilsen, H.M., Herrera, P.A., Ashraf, M., Ligaarden, I., Iding, M., Hermanrud, C., Lie, K.-A., Nordbotten, J.M., Dahle, H.K., Keilegavlen, E., 2011. Field-case simulation of CO_2-plume migration using vertical-equilibrium models. Energy Procedia 4 (0), 3801–3808. http://dx.doi.org/10.1016/j.egypro.2011.02.315.
Nilsen, H.M., Syversveen, A.R., Lie, K.-A., Tveranger, J., Nordbotten, J.M., 2012. Impact of top-surface morphology on CO_2 storage capacity. Int. J. Greenh. Gas Control 11 (0), 221–235. http://dx.doi.org/10.1016/j.ijggc.2012.08.012.
Nilsen, H.M., Lie, K.-A., Møyner, O., Andersen, O., 2015a. Spill-point analysis and structural trapping capacity in saline aquifers using mrst-co2lab. Comput. Geosci. 75, 33–43. http://dx.doi.org/10.1016/j.cageo.2014.11.002.
Nilsen, H.M., Lie, K.-A., Andersen, O., 2015b. Robust Simulation of sharp-Interface Models for Fast Estimation of CO_2 Trapping Capacity. URL ⟨http://folk.uio.no/kalie/papers/co2lab-2.pdf⟩.
Nilsen, H.M., Lie, K.-A., Andersen, O., 2015c. Fully Implicit Simulation of Vertical-equilibrium Models with Hysteresis and Capillary Fringe. URL ⟨http://folk.uio.no/kalie/papers/co2lab-3.pdf⟩.
Nordbotten, J.M., Celia, M.A., 2012. Geological Storage of CO_2: Modeling Approaches for Large-Scale Simulation. John Wiley & Sons, Hoboken, New Jersey.
Nordbotten, J.M., Dahle, H.K., 2011. Impact of the capillary fringe in vertically integrated models for CO_2 storage. Water Resour. Res. 47 (2), W02537. http://dx.doi.org/10.1029/2009WR008958.
Nordbotten, J.M., Flemisch, B., Gasda, S.E., Nilsen, H.M., Fan, Y., Pickup, G.E., Wiese, B., Celia, M.A., Dahle, H.K., Eigestad, G.T., Pruess, K., 2012. Uncertainties in practical simulation of CO_2 storage. Int. J. Greenh. Gas Control 9 (0), 234–242. http://dx.doi.org/10.1016/j.ijggc.2012.03.007.
Raynaud, X., Krogstad, S., Nilsen, H.M., 2014. Reservoir management optimization using calibrated transmissibility upscaling. In: ECMOR XIV – 14th European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8–11 September 2014, EAGE. http://dx.doi.org/10.3997/2214-4609.20141864.
Singh, V., Cavanagh, A., Hansen, H., Nazarian, B., Iding, M., Ringrose, P., 2010. Reservoir modeling of CO_2 plume behavior calibrated against monitoring data from Sleipner, Norway. In: SPE Annual Technical Conference and Exhibition, 19–22 September 2010, Florence, Italy, SPE 134891-MS. http://dx.doi.org/10.2118/134891-MS.
SINTEF ICT, The MATLAB Reservoir Simulation Toolbox: Numerical CO_2 Laboratory, October 2014. URL ⟨http://www.sintef.no/co2lab⟩.
Sundal, A., Nystuen, J.P., Dypvik, H., Miri, R., Aagaard, P., 2013. Effects of geological heterogeneity on CO_2 distribution and migration – a case study from the Johansen Formation, Norway. Energy Procedia 37, 5046–5054. http://dx.doi.org/10.1016/j.egypro.2013.06.418, gHGT-11.
Syversveen, A.R., Nilsen, H.M., Lie, K.-A., Tveranger, J., Abrahamsen, P. 2012. A study on how top-surface morphology influences the storage capacity of CO_2 in saline aquifers. In: Abrahamsen, P., Hauge, R., Kolbjørnsen, O. (Eds.), Geostatistics Oslo 2012, Quantitative Geology and Geostatistics, vol. 17, 2012, Springer, Netherlands, pp. 481–492. http://dx.doi.org/10.10.1007/978-94-007-4153-9.
Thibeau, S., Mucha, V., 2011. Have we overestimated saline aquifer CO_2 storage capacities? Oil Gas Sci. Technol.—Rev. IFP Energies N. 66 (1), 81–92. http://dx.doi.org/10.2516/ogst/2011004.
U.S. Department of Energy, Office of Fossil Energy, The 2012 United States Carbon Utilization and Storage Atlas, 4th Edition, 2012. URL ⟨http://www.netl.doe.gov/research/coal/carbon-storage/atlasiv⟩.

Paper VII

Reexamining CO_2 Storage Capacity and Utilization of the Utsira Formation

Odd A. Andersen, Halvor M. Nilsen, Knut-Andreas Lie
ECMOR XIV - 14th European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8-11 September 2014

ECMOR XIV

Mo P29

Reexamining CO2 Storage Capacity and Utilization of the Utsira Formation

O.A. Andersen* (SINTEF ICT), H.M. Nilsen (SINTEF ICT) & K.A. Lie (SINTEF ICT)

SUMMARY

In this work we provide estimates of CO2 storage capacity of the Utsira Formation using the recently provided datasets from the Norwegian Petroleum Directorate, taking CO2 density variation into account. We also investigate strategies on how to realize as much as possible of this potential in large-scale injection scenarios. We base our study on the assumption that the limiting factor for CO2 storage at Utsira is the efficient use of available trapping mechanisms, with pressure buildup being of secondary concern. We consider the trapping mechanisms considered most important for the medium-to-long term (structural, residual and dissolution trapping), using a combination of modeling tools based on the Matlab Reservoir Simulation Toolbox (MRST).

Introduction

The Utsira Formation is a large saline aquifer covering some 26.000 square kilometers (Chadwick et al., 2012), consisting of a more than a hundred meter thick column of high-permeability sand. The formation has been used as a target for industrial-scale CO_2 sequestration since 1996, where the injected CO_2 is a by-product from natural gas extraction at the nearby Sleipner Vest field (Baklid et al., 1996). Since then, approximately 1 Mt of CO_2 has been injected per year. While this storage operation is unique in its kind, the involved quantity falls far short of the storage potential of the site, and remains insignificant compared to the quantities necessary for CCS to be a significant part of an European greenhouse gas mitigation scenario. For instance, net European CO_2 emissions from energy industries, manufacturing and construction (comprising the industries for which CCS can be considered at least nominally possible) totaled 1.95 Gt in 2011 (UNFCCC, 2011). If CCS with geological storage were to handle a significant amount of this, hundreds of megatonnes of CO_2 would need to be injected annually into geological formations for permanent storage, orders of magnitude above current operations at Utsira.

Several capacity figures for the Utsira aquifer have been published in previous literature. A summary table in (Thibeau and Mucha, 2011) provides an overview, where estimated total capacity ranges from 0.3 Gt (Chadwick et al., 2008) to 20–60 Gt (Lindeberg et al., 2009). The large variation in published estimates is due to different assumptions made, in particular to what is considered the main limiting factors. If the aquifer is considered closed, or with low hydraulic connectivity, then pressure buildup becomes the main concern. The estimates in (Lindeberg et al., 2009; Thibeau and Mucha, 2011) are based on the prevention of excessive pressure buildup, and the 60 Gt estimated figure from (Lindeberg et al., 2009) is arrived at by assuming extensive production of formation water to alleviate pressure buildup from CO_2 injection into a closed system. On the other hand, if the aquifer is considered open with good hydraulic connectivity, leakage risk due to long-term CO_2 migration typically becomes the main limiting factor on storage capacity. In that case, the central issue defining storage capacity is the amount of CO_2 the storage site can retain over time by various trapping mechanisms. The most immediately available trapping mechanism is stratigraphic or structural trapping, where the CO_2 migration is blocked by low-permeability rock barriers. The lower figure of 0.3 Gt cited above only considers this type of trapping. Other available trapping mechanisms include residual trapping, dissolution and mineral trapping, whose full potential cannot be efficiently realized, or happen over much longer time spans.

In 2011, the Norwegian Petroleum Directorate released a CO_2 Storage Atlas of the North Sea (Halland et al., 2011), providing detailed models of twenty-one formations having the potential for CO_2 storage. In the CO_2 Storage Atlas, the Utsira Formation is classified as an open aquifer and the corresponding model includes caprock depth and thickness values over the full range of the aquifer in the Norwegian sector. The storage capacity of the combined Utsira–Skade system is estimated to be 16 Gt, with a prospectivity of 0.5–1.5 Gt. Pressure buildup at the site has been studied by Chadwick et al. (2012), who conclude that available pressure monitoring data suggests very minor pressure increases (perhaps 0.1 MPa) away from the injection point, consistent with little or no flow compartmentalization. Herein, we employ functionality from the CO_2-module of an open-source software (Lie et al., 2012; MRST) to reexamine the total storage capacity of the aquifer as described by the data set from the CO_2 Storage Atlas, and look at possible injection strategies for large-scale operations. While our study is simplistic in several aspects (e.g. assuming homogeneous rock properties and fully open boundaries), it sketches a possible approach for preliminary exploration of possible scenarios and estimation of realized trapping capacity.

Method

In (Nilsen et al., 2014c) we describe how we generate an aquifer model of Utsira from the data provided in (Halland et al., 2011), using functionality available in the open-source Matlab Reservoir Simulation Toolbox (MRST). In addition to providing simple access to the CO_2 Storage Atlas data, the next release

of `MRST-co2lab` will include column-based storage estimates, geometrical analysis of the caprock (Nilsen et al., 2014c), vertical equilibrium simulators (Nilsen et al., 2014a,b), and optimization of well placement and injection rates (Lie et al., 2014).

Data and assumptions

In the present work, we use the atlas model of Utsira together with parameters from (Singh et al., 2010; Holloway et al., 2004; Chadwick et al., 2008), as listed in Table 11 to estimate volumetric trapping capacities. These parameters suggest a temperature at the depth of the current injection point of 40.2°C, in the middle of the suggested range of 36°C to 46°C from (Chadwick et al., 2008). We further use the equation-of-state for CO_2 specified by Span and Wagner (1996) to estimate local CO_2 densities and compute trapping capacity in terms of mass. By lack of more detailed data, rock porosity and residual saturations are assumed constant across the aquifer.

Table 1 *Parameter values used to estimate trapping capacities for the Utsira Formation.*

Parameter	Value	Unit	Reference
Sea depth	80	m	Holloway et al. (2004)
Injection depth	1012	m	Singh et al. (2010)
Thermal gradient	35.6	°C/km	Singh et al. (2010)
Seabed temperature	7	°C	Singh et al. (2010)
Residual water saturation	0.11		Singh et al. (2010)
Residual CO_2 saturation	0.21		Singh et al. (2010)
Rock porosity	0.36		Singh et al. (2010)
Water density	1020	kg/m^3	Singh et al. (2010)
CO_2 solubility in brine	53	kg/m^3	Chadwick et al. (2008)

From the conclusions in (Chadwick et al., 2012) and the classification of the formation as an open aquifer (Halland et al., 2011), we base our capacity estimate on the assumption that the ability of the formation to ultimately retain injected CO_2 is the main limiting factor. We assume that any overpressure from the injection operation will dissipate over time, so that the long-term pressure regime is hydrostatic. We further assume long-term thermal equilibrium, with ambient temperature given by the thermal gradient and reference (seabed) temperature. Structural traps are identified and their volume assessed by geometric analysis of the caprock shape using the algorithms described in (Nilsen et al., 2014c). For dissolution trapping, the value for CO_2 solubility is taken from (Chadwick et al., 2008).

We consider the storage capacity of Utsira to be the maximum quantity of CO_2 that can be ultimately retained by structural, residual and dissolution trapping. Although the formation pore space can accommodate even larger quantities of CO_2, any amount in excess of this retaining capacity will migrate and ultimately escape the domain boundaries, potentially leaking back to the surface.

Total trapping capacity of a vertical column

If we approximate the formation geometry as a grid of vertical pillars, we can compute the trapping capacity of each such pillar separately, and add up to obtain the total capacity estimate. The height H of a given pillar can be written $H = h_1 + h_2$, where $h_1 (\geq 0)$ represents the part contained within a structural trap, as illustrated in Figure 1.

For the part of the column that is within a structural trap, the full retaining capacity Q_1 is reached when the pore space is maximally saturated with CO_2 and the remaining brine contains the maximal amount of dissolved CO_2:

$$Q_1 = Ah_1\phi\left[s_{w,r}c_{max} + (1 - s_{w,r})\rho_{co_2}\right]. \tag{1}$$

Here, A is the area of a lateral cross-section of the pillar, ϕ the rock porosity, $s_{w,r}$ the residual water

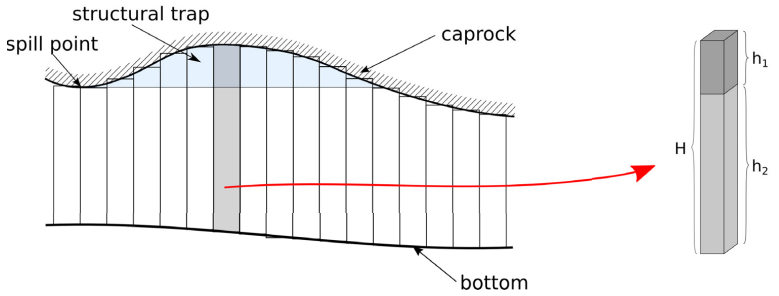


Figure 1 *Illustration of how the aquifer volume is divided into a mesh of pillars. Left: aquifer cross section. Right: a single pillar.*

saturation, ρ_{co_2} the CO_2 density, and c_{max} the maximum mass of CO_2 that can be dissolved per volume of formation water. We consider ϕ and c_{max} to be constants as given in Table 1, whereas ρ_{co_2} is a function of temperature and pressure.

Similarly, the part of the column not contained within a structural trap is at full retaining capacity Q_2 when all pore space contains the residual amount of CO_2 and all remaining brine contains the maximal amount of dissolved CO_2

$$Q_2 = Ah_2\phi\left[(1-s_{n,r})c_{max} + s_{n,r}\rho_{co_2}\right], \tag{2}$$

where $s_{n,r}$ represents the residual saturation of CO_2.

We now use superscript i to denote a given pillar. The total retaining capacity of pillar i is

$$Q^i = Q_1^i + Q_2^i \tag{3}$$

and the total estimate for the aquifer becomes

$$Q = \sum_i Q^i = \sum_i \left(D^i + S^i + R^i\right) = D + S + R, \tag{4}$$

where

$$\begin{aligned} D^i &= c_{max}\left[h_1^i s_{w,r} + h_2^i(1-s_{n,r})\right]A\phi, \\ S^i &= \rho_{co_2}^i\left[h_1^i(1-s_{w,r}) + h_2^i s_{n,r}\right]A\phi, \\ R^i &= \rho_{co_2}^i h_2^i s_{n,r} A\phi. \end{aligned}$$

Here, $D = \sum_i D^i$, $S = \sum_i S^i$, and $R = \sum_i R^i$ represent total CO_2 mass in the aquifer that can be retained by dissolution, structural trapping, and residual trapping, respectively.

Potential storage capacity of the Utsira formation

At the finest scale, our digital Utsira model is discretized using approximately one hundred thousand vertical pillars, each with a lateral cross-section of 500×500 m. Applying formula (3) on each pillar and summing up, we obtain the figures presented in Table 2. From this calculation, the estimated total trapping capacity of the aquifer surpasses 112 Gt, 69% of which is attributed to residual trapping. The aquifer has an estimated 1.13 Gt of structural trapping capacity, which is only 1% of the total figure.

Using values for the individual pillars, we construct a map of local storage capacity of the Utsira Formation (Figure 2, left). On this map, we note a wide range in local trapping capacities. Some pillars

Table 2 *Trapping capacity of the Utsira aquifer.*

Type	Value [Gt]	% of total
Structural	1.13	1.0
Residual	77.05	68.7
Dissolution	34.06	30.3
Total	112.24	100

are able to retain more than 4 Mt of CO_2 whereas others retain practically nothing. The highest local capacities are seen across a large region in the south. Another region with moderately high trapping capacities is located in the north. High storage capacities are associated with thick parts of the aquifer, particularly in the presence of structural traps and at depths where CO_2 is in a dense phase. A large part of the Utsira aquifer is so shallow that CO_2 will be in gas phase at equilibrium, resulting in significantly lower storage capacities. The boundary between regions with gas and liquid CO_2 can clearly be seen in the figure as a sharp, arch-shaped discontinuity that cuts into the aquifer from southwest and exits in the northwest.

For most open aquifers there will be no practical way of utilizing all, or even a significant fraction of the full storage capacity as defined above. Although the capacity of structural traps can in theory be fully exploited by strategic positioning of injection sites, efficient use of residual and dissolution trapping is hard to achieve. Because of density differences, CO_2 and brine will tend to separate after injection into separate volumes, with the CO_2 volume on top. The amount of residual trapping realized in a storage scenario will therefore depend on the sweep efficiency, i.e., the thickness of the migrating CO_2 plume compared to the local height of the aquifer. As the plume migrates and spreads out by buoyancy forces, it will also gradually thin out. Efficient use of residual trapping therefore mostly happens relatively close to the injection point, where the CO_2 plume still retains much of its initial thickness. Likewise, since dissolution of CO_2 into brine happens only gradually, dissolution trapping can improve the overall CO_2 retention rate at a given location only in the long-term presence of otherwise mobile CO_2.

In the middle diagram of Figure 2 we have plotted the combined capacity of pillars reachable by infinitesimal, purely gravity-driven migration originating from a given point in the model. We refer to this value as "total reachable capacity". In other words, if a small rate of CO_2 is continuously injected at some point in the aquifer, the reachable capacity value associated with that point is the total trapping capacity of all pillars encountered as the CO_2 migrates upwards following the steepest slope of the caprock. Whenever a structural trap is encountered, all pillars associated with that trap are considered reachable, since the CO_2 will accumulate there and spread out until the entire trap area is covered. The result is a plot with large, relative uniform regions associated with spill regions ("watersheds") connected with individual structural traps. This plot can be used as a simple heuristic to choose CO_2 injection locations associated with migration pathways that have high trapping potential. This should only be seen as an imperfect approach, in that it only considers infinitesimal flow, does not take sweep efficiency or migration speed (associated with caprock steepness) into account, and depends on the resolution of the aquifer model.

If we only consider structural trapping when computing reachable capacities, we end up with the right plot in Figure 2. We refer to the values thus obtained as representing "reachable structural capacity". The division of the aquifer into distinct regions is even stronger than for the middle figure, as any point associated with the spill region of a given trap has exactly the same value of reachable structural capacity. Although this plot disregards residual and dissolution trapping, it is more precise in what it covers, since structural capacity can in theory be fully utilized. When choosing an injection site with a goal of maximizing structural trapping only, "reachable structural capacity" thus provides clearer guidance than "total reachable capacity" does.

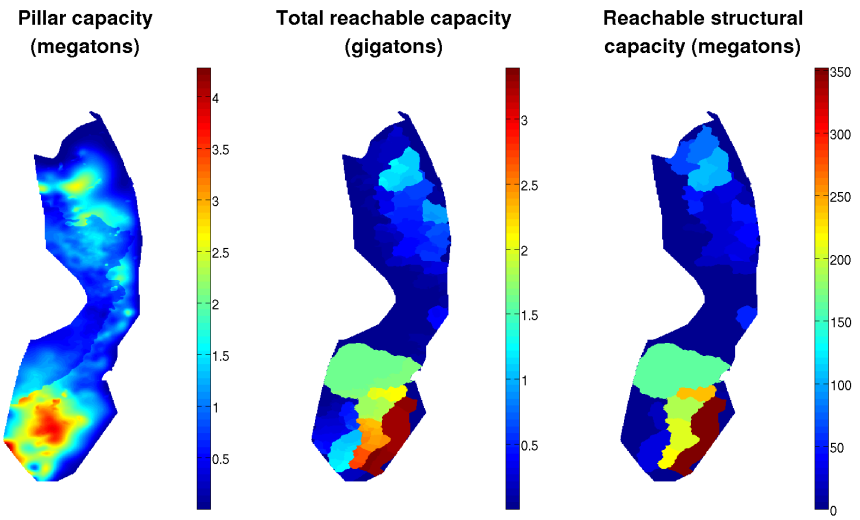


Figure 2 *Left: trapping capacity of individual vertical pillars in the Utsira model. Middle: combined trapping capacity reachable by gravity-driven migration from a given point. Right: combined structural trapping capacity reachable by gravity-driven migration from a given point.*

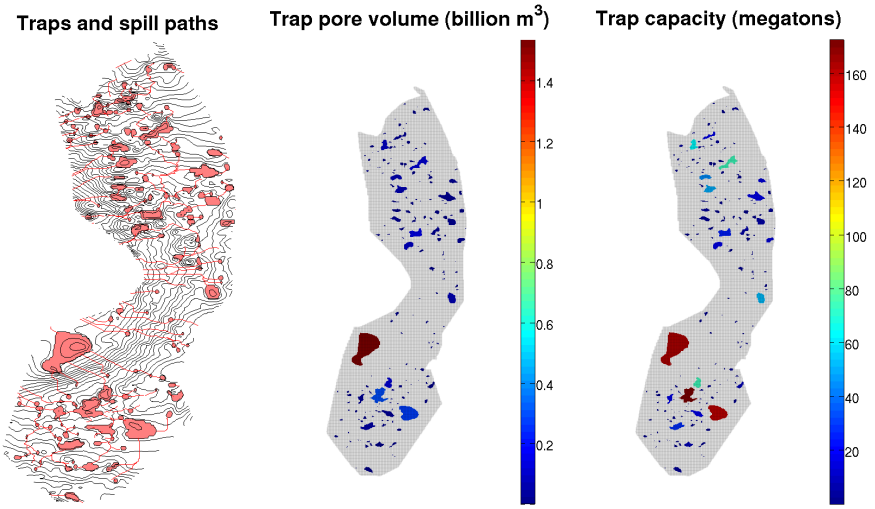


Figure 3 *Left: topographic map of the Utsira caprock, with structural traps and spill paths indicated. Middle: structural traps colored by pore volume. Right: structural traps colored by trapping capacity in mass terms.*

Mapping the structural traps of Utsira

Although structural traps only provide a very small fraction of the theoretical Utsira trapping capacity, they are important since they provide the most readily exploitable trapping mechanism, and because they slow down migration of the advancing CO_2 plume. A wide trap will make a narrow plume spread out, thus also facilitating dissolution.

In the left plot of Figure 3, we see a topographical map of the Utsira caprock with all traps and spill paths are traced out, as identified by a purely geometrical analysis. We observe an intricate, hierarchical system where multiple deeper traps spill into shallower ones along tributaries defined by spill paths. The largest traps are found in the southern parts of the aquifer. The general spill direction is from the east upwards into the shallower regions of the west.

In the middle plot of Figure 3, we have colored all the traps by total pore volume. We note that by this measure, the single largest trap by far is located in the southwest, with a pore volume of around 1.5 billion cubic meters. However, this does not translate directly into actual trapping capacity, presented in the rightmost plot of Figure 3. On this diagram, we see that the trap with the largest structural trapping capacity is now found in the middle of the southern region, where several other traps also gain prominence. The difference is due to varying CO_2 density. Indeed, the largest trap in terms of pore volume is located in a part of the aquifer where CO_2 is expected to be in gas phase, thus significantly reducing its structural trapping capacity.

Injection strategies

In practice, it will not be possible to exploit more than a small fraction of the total Utsira retaining capacity. However, our estimates serve as a theoretical upper bound on the quantity of CO_2 that can be stored at Utsira, and provide a reference for comparison when exploring possible injection scenarios. In the present section we identify and assess a few such scenarios. We choose injection locations and rates with the goal of maximizing the ultimately retained amount of CO_2. To do this, we employ a combination of spill-point analysis, "rapid" simulations and nonlinear optimization, all of which is functionality provided by MRST.

Defining scenarios

For our purposes, an injection scenario is defined by (1) the location of a given number of injection sites; and (2) the rates injected at each site over time. Our task is therefore to choose favorable injection sites and optimal injection schedules to maximize realized storage capacity and minimize leakages. In reality, there will naturally be logistical and other factors further constraining these choices, but we do not consider them here. However, such restrictions could be integrated in the proposed workflow by adding constraints and modifying the objective function of the optimization routine presented below.

Since structural traps provide the most immediately available and easiest to realize containment mechanism, a reasonable strategy would be to choose injection sites based on how much structural trapping capacity they can reach. To do this, we first map out the hierarchical system of structural traps, connecting spill paths and spill regions, and compute the storage capacity of each trap in mass terms. We can then apply a greedy algorithm to identify the set of injection point with the combined highest reachable structural capacity, adjusted to avoid double-counting (Lie et al., 2014). Since reachable structural capacity of all points within a given spill region is identical (cf. Figure 2, right), we use distance from model boundary as an additional selection criterion. In reality, flow will not be infinitesimal, so injecting CO_2 too close to the model boundary will often lead to significant leakage out of the domain even if the associated spill region leads into an internal trap.

Assuming that most of the injected CO_2 will migrate along the path predicted by caprock topography,

Table 3 *Simulation parameter values for optimization of injection rates*

Parameter	Value	Unit	Reference
Injection period	50	year	
Migration period	3000	year	
W (importance of leakage)	10		
brine viscosity	$8 \cdot 10^{-4}$	Pascal second	Singh et al. (2010)
CO_2 viscosity	$6 \cdot 10^{-5}$	Pascal second	Singh et al. (2010)
rock permeability	2	Darcy	Singh et al. (2010)

the reachable structural capacity figures provide immediate estimates on how much CO_2 can be injected at each site. Dividing these quantities by the total injection period results in relatively conservative injection rates compared to total expected trapping potential, since we expect additional trapping from capillarity and dissolution. The additional amount trapped by these mechanisms is hard to estimate a priori, but can be obtained from numerical simulation of the whole injection and migration phases. Vertical equilibrium (VE) modeling (Nilsen et al., 2014a) provides us with a tool to carry out simplified long-term simulations rapidly while respecting the most relevant physics. The ability to run multiple rapid simulations enables us to compute optimal injection schedules using nonlinear optimization. In this approach, the (fixed or time-dependent) injection rates at each site represent the variables of the problem, and we define an objective function that we seek to optimize. We here choose the objective function to be the combined total amount of CO_2 injected at all sites, minus the total amount of CO_2 escaped by the end of the simulation period multiplied by some weighting factor W. The use of this weighting factor enables us to specify the relative importance to be given to leakage compared to total amount injected – a high value of W means little leakage is tolerated, and optimal injection rates will end up being lower. We solve the resulting optimization problem using a steepest descent method with gradients obtained by an adjoint method (Raynaud et al., 2014; Jansen, 2011) in combination with automatic differentiation. We optimize for scenarios in which the initial injection period is 50 years followed by a migration period of 3000 years. The weighting factor W and other relevant simulation parameters used are listed in Table 3 (with the exception of those already listed in Table 1).

Even though the selection of injection sites is based solely on the maximization of structural trapping, the subsequent optimization of injection schedules allows us to take other trapping mechanisms into account. In principle, these would include at least residual and dissolution trapping. In the following examples however, we consider only structural and residual trapping for optimization of schedules, i.e., we disregard dissolution. There are two reasons for this. Firstly, including dissolution in VE simulations incur significant computational cost that would lead to very long runtimes for the optimization algorithm. This problem could partly be tackled by more fine-tuned algorithmic design, possibly on parallel hardware, but is a topic for future work. Secondly, while figures for CO_2 solubility in brine can be derived for Utsira conditions (cf. Table 1), little information exist on the effective dissolution *rates*. Theoretical studies indicate that if and when convective mixing between CO_2-saturated and unsaturated brine occurs, the subsequent dissolution rate will remain relatively constant until dissolution ends (Nordbotten and Celia, 2012). As the Utsira aquifer has a very high permeability, it is possible that significant convective mixing will happen. On the other hand, rate and onset time will likely be highly sensitive to the presence and vertical permeability of intra-reservoir mudstone barriers, which are observed at the Sleipner injection site. In the one-well example presented below, we carry out an additional simulation that includes dissolution trapping to show the potential impact. For simulation with dissolution, we use a rate of 0.44 kg per square meter per year. This is similar to the value used in (Gasda et al., 2011), which was there chosen by reference to theoretical studies. The impact of dissolution shown in the first example below should therefore be only understood as a qualitative illustration.

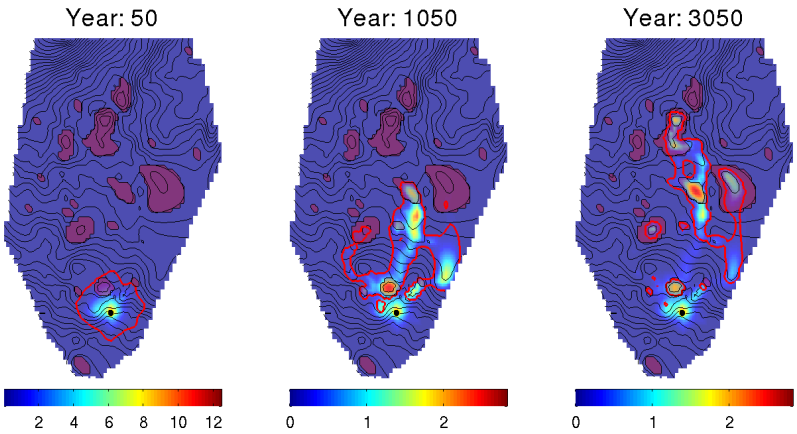


Figure 4 *Simulation of Scenario 1 with unoptimized injection rate. CO_2 plume outlined in red, structural traps overlaid in purple. Total vertical integrated CO_2 content indicated with color (unit: tonnes per lateral square meter).*

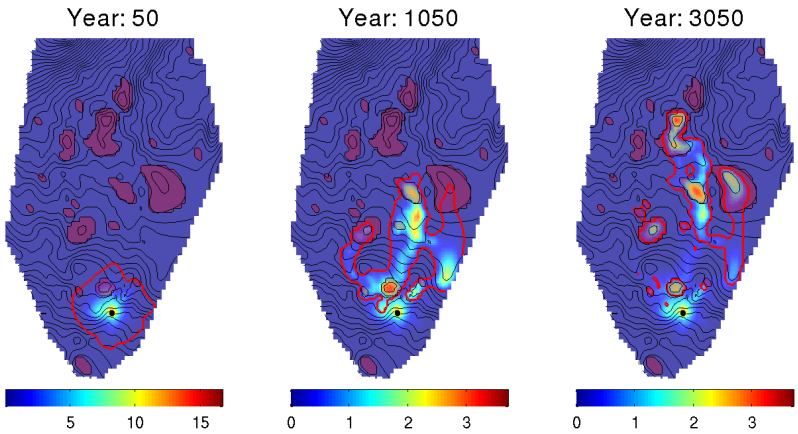


Figure 5 *Simulation of Scenario 1 after optimization of injection rate. CO_2 plume outlined in red, structural traps overlaid in purple. Total vertical integrated CO_2 content indicated with color (unit: tonnes per lateral square meter).*

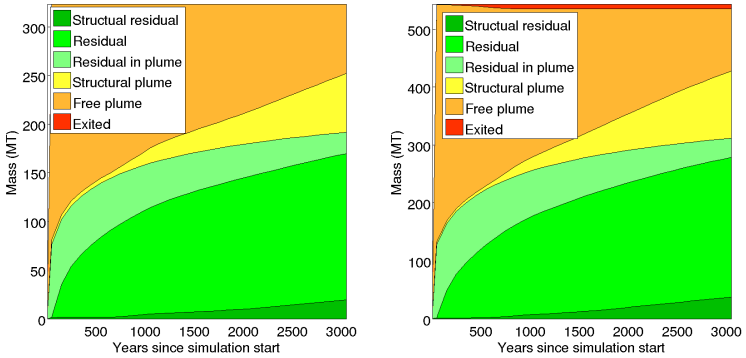


Figure 6 *Distribution of injected CO_2 in Scenario 1 as a function of time. Left diagram represents injection with unoptimized rates and right diagram with optimized rates.*

Scenario 1: Injection from a single site

In this example, we seek a good single-site injection scenario. Using reachable structural capacity as site selection criterion, we identify a location at the very south of the aquifer, indicated by a black dot in Figure 4 (only the part of the aquifer relevant for the subsequent simulation is shown). The reachable structural capacity at this point is 323 Mt, which translates into an injection rate (before optimization) of 6.46 Mt per year for 50 years. The outcome of a VE simulation of this scenario is presented in Figure 4, where the evolution of the CO_2 plume during the migration phase can be clearly seen. The moving plume leaves behind a trail of residually trapped CO_2. This is clearly visible on the color plot, which indicates total CO_2 content per square meter. We see that after 3000 years of migration, part of the CO_2 plume is located within structural traps but some is still migrating. This can also be seen in the left diagram of Figure 6, which shows the time-dependent distribution of injected CO_2 among various states of trapping, flow and leakage. We see that after 3000 years, free-flowing CO_2 outside traps ("Free plume") still represents a significant fraction of the injected quantity. The amount of CO_2 that has reached structural traps is indicated by "structural plume" and "structural residual", together making up approximately 25% of the total. Since the total amount injected equals the estimated reachable structural capacity, it is clear that most of this capacity has not been reached. This is partly due to the fact that migration has not yet ended, but another important factor is the large quantity of CO_2 that has become structurally trapped along the way ("Residual" and "Residual in plume"). It is clear from the figure that choosing an injection rate only based on reachable structural trapping can seriously underestimate actual realizable trapping capacity.

We now compute an optimal injection schedule as previously described. The outcome suggests a total amount of 543 Mt, injected at an almost-constant rate over the 50 year timespan. This is approximately 68% higher than for the unoptimized schedule. We re-run our scenario with this optimized schedule. The resulting migration is illustrated in Figure 5. We observe that the plume is now somewhat thicker, but has not migrated any further than in the previous case. However, more of the CO_2 has reached the trap at the northernmost tip of the plume, as can be read from the color plot. The right diagram in Figure 6 presents the corresponding CO_2 trapping distribution plot. We see that although the total quantity injected is significantly higher than for the unoptimized case, the relative share of each trapping mechanism remains roughly the same. In addition, we can see a thin red sliver at the top, which represents CO_2 that has escaped the domain (e.g., leaked). We note that the leakage happened during a period of some 800 years starting shortly after injection stop. From the previous plume plot (Figure 5) it is clear that this leakage occurred as the migrating plume brushed against the eastern border. The total amount escaped through the vertical boundaries is 8.5 Mt or 1.57% of the total injected quantity.

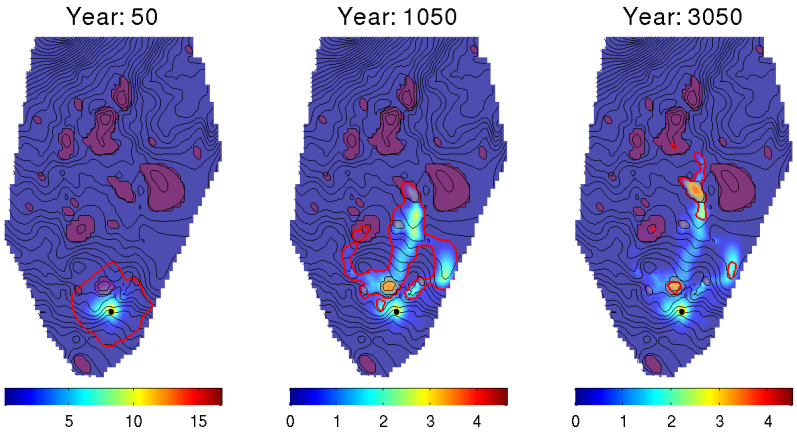


Figure 7 *Simulation of Scenario 1 after optimization of injection rate and with dissolution trapping included. CO_2 plume outlined in red, structural traps overlaid in purple. Total vertical integrated CO_2 content indicated with color (unit: tonnes per lateral square meter).*

To assess the potential impact of dissolution, we re-run the scenario with dissolution trapping enabled. We first do this for the optimized injection rates from the previous simulation. The resulting plume evolution is presented in Figure 7, where we note a dramatic difference from the case without dissolution. After 3000 years, only a few small isolated pockets of the CO_2 plume remain, and some traps have not been reached at all. From Figure 8, we see that most of the CO_2 has indeed dissolved in the brine. The total leakage has also been reduced to 0.7 Mt, or 0.13% of the total, which is too small to be visible on the figure.

We run the scenario once again, this time increasing the injection rate by 50%, for a total of 815 Mt over 50 years. The outcome is presented in Figure 9 and the right diagram of Figure 8. We see that while the plume reaches a bit farther this time, most of the CO_2 still ends up dissolved, and the amount that escapes across the vertical boundaries (7.8 Mt, or 0.96% of the total) is still lower than the original case without dissolution. This experiment suggests that by ignoring dissolution trapping, we run the risk of severely underestimating the amount of CO_2 that can will be trapped for a given scenario. Case-specific knowledge on dissolution rates at Utsira thus remains an important unknown in any attempt to predict long-term plume migration and realized storage capacity.

Scenario 2: Injection from multiple sites

By using multiple injection sites, a larger part of the aquifer can be reached. In this example, we define a scenario with ten sites. Site locations and injection schedules are chosen using the same approach as the previous example. The chosen sites can be seen as black dots on the plots in Figure 10. In addition to the injection site from the previous example, three more sites have been added to the southern region. The northern region is covered with five injection sites, whereas one injection site has been included in the narrow corridor connecting the two regions. This connecting corridor is relatively thin, steeply inclined, shallow, and contains few structural traps. It is thus of limited use for CO_2 storage, but the presence of a small trap to the northeast of the injection site led to the placement of the middle injection site.

Figure 10 shows the result of simulating the scenario before optimization of schedules. The total injected amount over the 50 year injection period is 894 Mt. As we can see on the corresponding distribution plot (Figure 11, left), this time almost 50% of the injected CO_2 reaches the structural traps during the simulation period. Another 40% ends up residually trapped, while approximately 10% either remains in the free part of the plume, or has escaped (1.45%, or 13 Mt, for the latter).

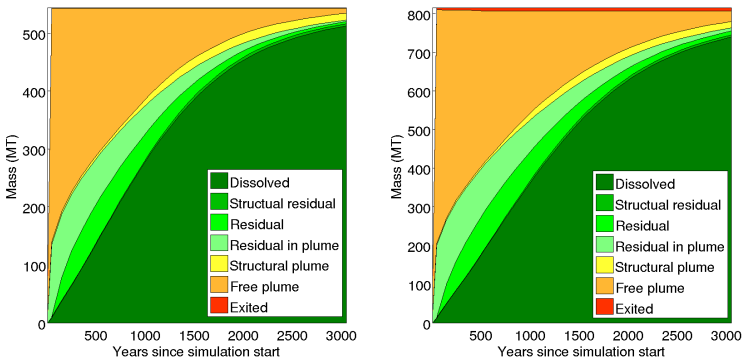


Figure 8 *Distribution of injected CO_2 in Scenario 1 as a function of time, including dissolution. Left diagram represents injection with the optimized schedule, and right diagram injection with 50% higher rates.*

After optimization of injection schedules, the total amount to be injected increases to 1598 Mt, an almost 80% increase. However, as can be seen on the resulting plume evolution plots on Figure 12, this increase is not evenly distributed among the injection sites. In particular, we note that the injection site in the middle region has been completely shut off. The reason is the leakage associated with injecting at this location. In the simulation involving *unoptimized* schedules, this site did manage to partly fill the small neighbor trap. However, during the injection process a significant amount of CO_2 was pushed the other way. This quantity quickly escaped up the steep slope towards the west during the migration period and left the aquifer domain. As the objective function used in the optimization process puts a large emphasis on preventing leakage ($W = 10$), the end result for this injection site was to be completely shut off. The right diagram of Figure 11 presents the distribution plot for this simulation. Out of 1.6 Gt of CO_2 injected, only 19 Mt, or 1.19%, has leaked. This is less than for the unoptimized case. However, we note that at 3050 years, around 20% of injected CO_2 is still freely flowing. If the simulation period had been further extended, parts of this quantity could still leak, forcing the optimization algorithm to reduce the injection rates. In other words, optimal use of the aquifer also depends on the time horizon considered.

Scenario 3: Optimizing for an array of injection sites

In the previous example, we noted that the optimization of injection schedules can cause some injection sites to become completely inactive. This observation leads to an alternative strategy for positioning injection sites. Instead of carefully selecting injection locations based on optimal use of structural traps, we can initially consider a large, evenly distributed array of injection sites that covers the whole domain of the aquifer. We then use the optimization of injection rates to identify sites that end up with zero or very small injection rates, and eliminate these sites. This approach treats all trapping mechanisms equally in the site selection process (although we have here excluded dissolution trapping, as explained above).

To test this approach at Utsira, we start with an initial, regular array of injection locations, measuring 99 km from east to west, 408 km from north to south, and consisting of 8 x 16 separate sites. We then remove all sites whose positions do not lie within the footprint of the aquifer, as well as all sites positioned over parts of the aquifer where CO_2 would be in a gaseous state. The location of the remaining sites can be seen as black dots in Figure 13. The initial, unoptimized schedule is defined by specifying a very large amount (4 Gt) to be injected over 50 years, evenly distributed across all sites. Although this amount is clearly in excess of what we expect to be able to store, we put the responsibility on the optimization algorithm to scale down rates where necessary to obtain an optimal set of schedules.

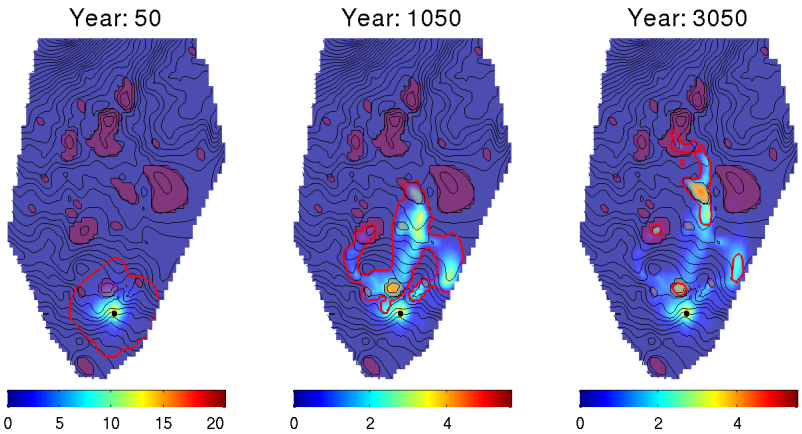


Figure 9 *Simulation of Scenario 1 with injection rate 50% higher than the optimized value, and with dissolution trapping included. CO_2 plume outlined in red, structural traps overlaid in purple. Total vertical integrated CO_2 content indicated with color (unit: tonnes per lateral square meter).*

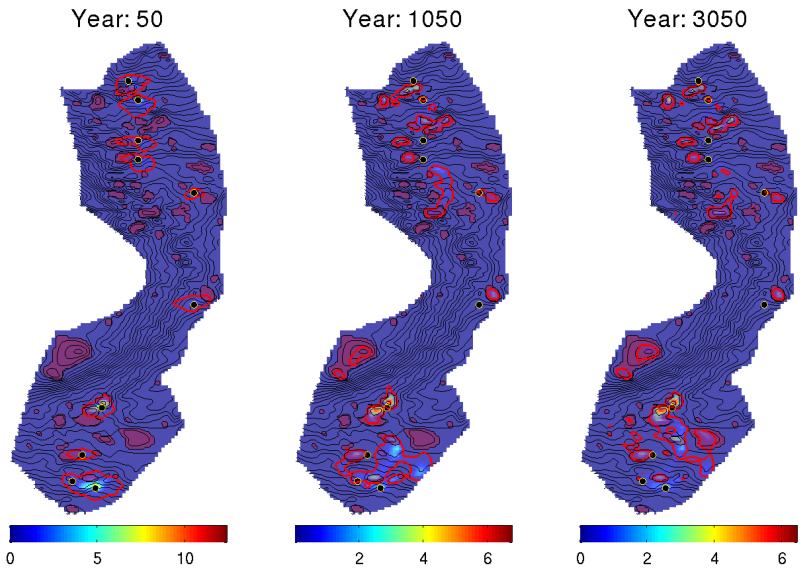


Figure 10 *Simulation of Scenario 2 with unoptimized injection rate. CO_2 plume outlined in red, structural traps overlaid in purple. Total vertical integrated CO_2 content indicated with color (unit: tonnes per lateral square meter).*

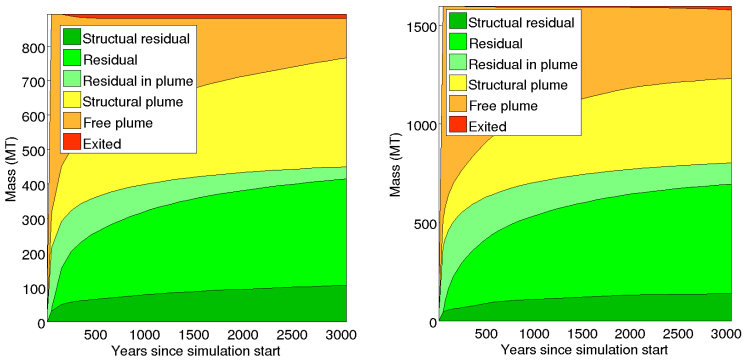


Figure 11 *Distribution of injected CO_2 in Scenario 2 as a function of time. Left diagram represents injection with unoptimized rates and right diagram with optimized rates.*

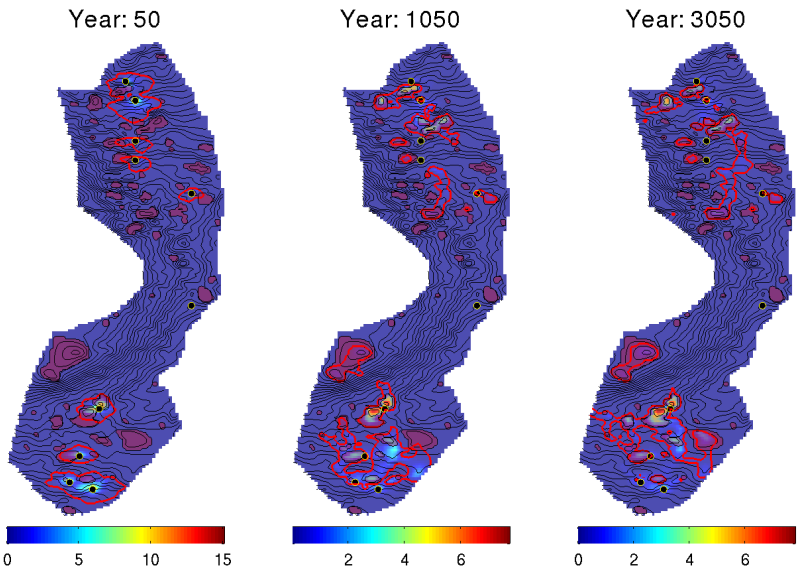


Figure 12 *Simulation of Scenario 2 after optimization of injection rate. CO_2 plume outlined in red, structural traps overlaid in purple. Total vertical integrated CO_2 content indicated with color (unit: tonnes per lateral square meter).*

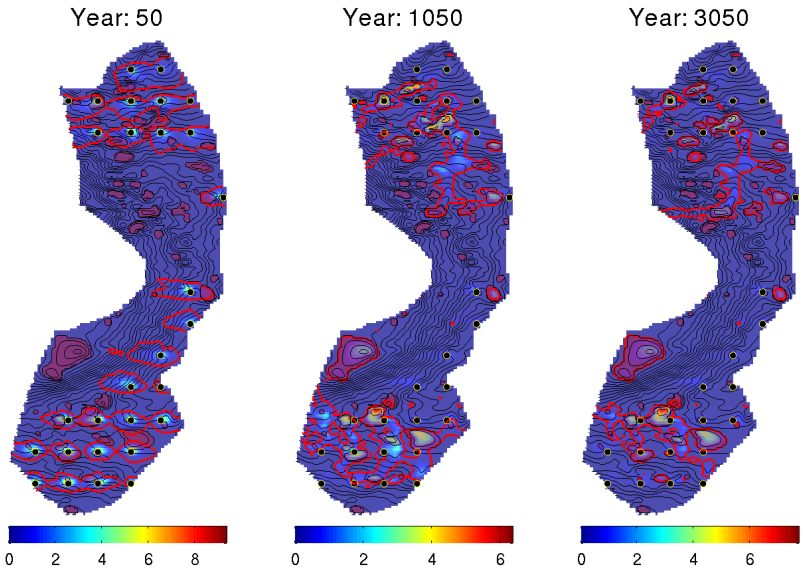


Figure 13 *Simulation of Scenario 3 with unoptimized injection rate. CO_2 plume outlined in red, structural traps overlaid in purple. Total vertical integrated CO_2 content indicated with color (unit: tonnes per lateral square meter).*

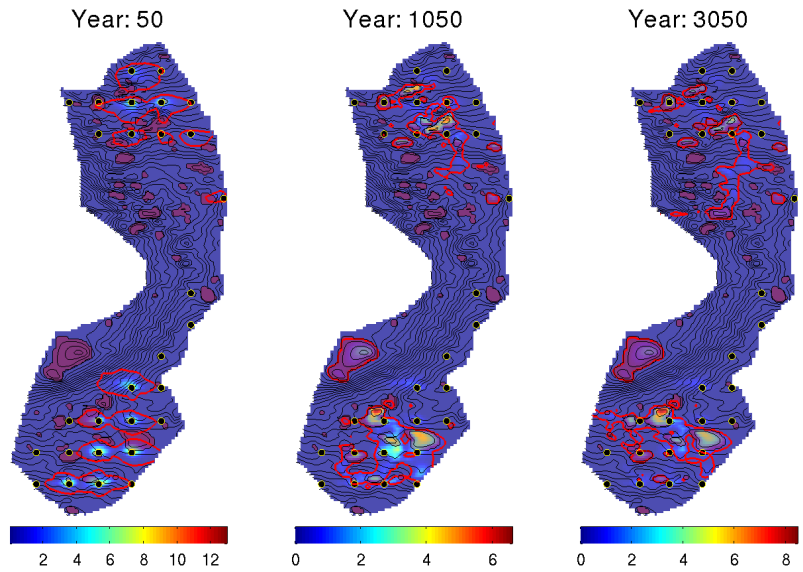


Figure 14 *Simulation of Scenario 3 after optimization of injection rate. CO_2 plume outlined in red, structural traps overlaid in purple. Total vertical integrated CO_2 content indicated with color (unit: tonnes per lateral square meter).*

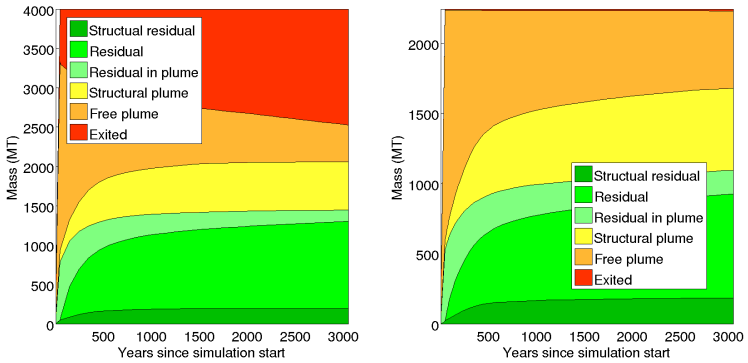


Figure 15 *Distribution of injected CO_2 in Scenario 3 as a function of time. Left diagram represents injection with unoptimized rates and right diagram with optimized rates.*

Running the simulation with the unoptimized schedules produces the results shown in Figure 13 and the left diagram of Figure 15. From the latter, we see that at the end of the injection period a large amount (almost 40%) of the injected CO_2 has leaked, as could be expected. Nevertheless, almost 2 Gt has been retained, predominantly as residual saturation. The large quantity of residually trapped CO_2 can be understood as a result of "flooding" the aquifer with large amounts of CO_2. Although much of it will leak back out, the size of the plume means that larger volumes of rock are reached, thus leaving behind more residually trapped CO_2. However, after approximately 1000 years the remaining free CO_2 primarily flows along previously visited pathways, and the amount of residually trapped CO_2 ceases to increase.

In Figure 14 and the right diagram of Figure 15, we see the result of simulating the scenarios with the optimized schedules. From the left plot of Figure 14 we note that only 14 of the initial 29 wells have actually injected CO_2, the rest being shut off. The shut-off sites include most of the sites along the eastern border, as well as some of the higher-lying sites in the northern and southern domains. Notably, no active injection site remains in the middle region. Using the optimized schedules, a total of 2.24 Gt is injected into the aquifer, with only 15.5 Mt (0.69%) escaped after 3000 years. From the distribution plot we see that while the total amount of realized structural capacity is approximately the same as in the unoptimized case, significantly less structural capacity has been realized. There is also a large quantity of free-flowing CO_2 remaining in the aquifer, so the potential for future leakage remains. Again, we see that optimal utilization of the aquifer will depend on the time horizon involved.

Conclusion

Using available published information about the Utsira Formation, we estimate a maximal retaining capacity of 112 Gt of CO_2 for the aquifer. Of this, structural trapping constitutes 1.13 Gt, which lies wihtin the range of 0.5–1.5 Gt for the estimated prospectivity of the aquifer according to the Norwegian Petroleum Directorate. In addition, we estimate that the aquifer is capable of retaining 77 Gt of CO_2 by residual trapping and 34 Gt by dissolution into the remaining formation water. The combined estimate only provides a theoretical upper bound, as there is no practical way to achieve more than a small fraction of this capacity for a real scenario.

We have demonstrated two possible approaches for defining good injection scenarios, illustrated by a few simulated examples. In the most ambitious scenario tested, a total of more than 2.2 Gt of CO_2, injected from fourteen separate injection sites, was ultimately retained in the aquifer by structural and residual trapping alone, with only negligible volumes leaking across the boundaries of the aquifer model, during the 3000 year simulation period.

If the rate of CO_2 dissolution into brine at Utsira proves to be significant, the realisable storage potential of the aquifer could be significantly larger than we get from estimates neglecting this effect, as illustrated in our first scenario. However, at present, the role of dissolution for CO_2 storage at Utsira remains unclear. Another uncertain factor that can have significant impact on results is the exact division of the aquifer into regions where CO_2 is in gas and in dense phase. The boundary of this region is here only inferred based on the assumed temperature gradient and hydrostatic pressure. It should also be mentioned that the geometric analysis of the caprock that identifies traps and spill regions can be sensitive to small changes, so a thorough analysis should take resolution, precision, and level of noise in the input data into account. We would also like to underline that the lack of detailed petrophyical data is a signinficant cause of uncertainty. However, the methods we propose work equally well with heterogeneity and can be used to study this uncertainty if more data becomes available.

Finally, we emphasize that the amount of realizable trapping capacity in an open aquifer such as Utsira will highly depend on the time horizon considered as well as the total amount of leakage tolerated during the injection and migration processes.

Acknowledgements

We thank Xavier Raynaud og Stein Krogstad for valuable contributions to the fully implicit solver and the adjoint calculations in MRST which we reused.

This work was founded by the MatMoRa-II project, Contract no. 21564 and Numerical CO_2 laboratory, Contract 199878 sponsored by the Research Council of Norway and Statoil ASA.

References

Baklid, A., Korbøl, R. and Owren, G. [1996] Sleipner Vest CO2 disposal, CO2 injection into a shallow underground aquifer. *SPE Annual Technical Conference and Exhibition, 6-9 October, Denver, Colorado*, doi: 10.2118/36600-MS, sPE-36600-MS.

Chadwick, A., Arts, R., Bernstone, C., May, F., Thibeau, S. and Zweigel, P. [2008] *Best practice for the storage of CO2 in saline aquifers – Observations and guidelines from the SACS and CO2STORE projects*, vol. 14 of *British Geological Survey Occasional Publication*. British Geological Survey, Nottingham, UK.

Chadwick, R.A., Williams, G.A., Williams, J.D.O. and Noy, D.J. [2012] Measuring pressure performance of a large saline aquifer during industrial-scale co2 injection : the Utsira Sand, Norwegian North Sea. *Int. J. Greenh. Gas Control*, **10**, 374–388, doi:http://dx.doi.org/10.1016/j.ijggc.2012.06.022.

Gasda, S.E., Nordbotten, J.M. and Celia, M.A. [2011] Vertically-averaged approaches to CO2 injection with solubility trapping. *Water Resources Research*, **47**, W05528, doi:10.1029/2010WR009075.

Halland, E.K., Johansen, W.T. and Riis, F. (Eds.) [2011] *CO_2 Storage Atlas: Norwegian North Sea*. Norwegian Petroleum Directorate, P. O. Box 600, NO–4003 Stavanger, Norway.

Holloway, S., Chadwick, A., Lindeberg, E., Czernichowski-Lauriol, I. and Arts, R. [2004] Best practice manual from SACS–Saline aquifer CO2 storage project. Tech. rep., Statoil Research Center, Trondheim, Norway, IEA Greenhouse Gas R&D Programme, Schlumberger Research, European Commission.

Jansen, J.D. [2011] Adjoint-based optimization of multi-phase flow through porous media - a review. *Computers & Fluids*.

Lie, K.A., Krogstad, S., Ligaarden, I.S., Natvig, J.R., Nilsen, H.M. and Skaflestad, B. [2012] Open source MATLAB implementation of consistent discretisations on complex grids. *Comput. Geosci.*, **16**, 297–322, ISSN 1420-0597, doi:10.1007/s10596-011-9244-4.

Lie, K.A., Nilsen, H.M., Andersen, O. and Møyner, O. [2014] A simulation workflow for large-scale CO_2 storage in the norwegian north sea. *ECMOR XIV – 14th European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8-11 September 2014*, EAGE.

Lindeberg, E., Vuillaume, J.F. and Ghaderi, A. [2009] Determination of the CO2 storage capacity of the utsira formation. *Energy Procedia*, **1**(1), 2777–2784, doi:10.1016/j.egypro.2009.02.049.

MRST [2014] The MATLAB Reservoir Simulation Toolbox, version 2014a. `http://www.sintef.no/MRST/`.

Nilsen, H.M., Lie, K.A., Andersen, O. and Møyner, O. [2014a] `MRST-co2lab`: sharp-interface models for fast estimation of trapping capacity. *submitted*.

Nilsen, H.M., Lie, K.A., Andersen, O. and Møyner, O. [2014b] `MRST-co2lab`: vertical-equilibrium models with hysteresis and capillary fringe. *submitted*.

Nilsen, H.M., Lie, K.A., Møyner, O. and Andersen, O. [2014c] `MRST-co2lab`: spill-point analysis and structural

trapping capacity in saline aquifers. *submitted*.
Nordbotten, J.M. and Celia, M.A. [2012] *Geological Storage of CO_2: Modeling Approaches for Large-Scale Simulation*. John Wiley & Sons, Hoboken, New Jersey.
Raynaud, X., Krogstad, S. and Nilsen, H.M. [2014] Reservoir management optimization using calibrated transmissibility upscaling. *ECMOR XIV – 14th European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8-11 September 2014*, EAGE.
Singh, V., Cavanagh, A., Hansen, H., Nazarian, B., Iding, M. and Ringrose, P. [2010] Reservoir modeling of CO_2 plume behavior calibrated against monitoring data from Sleipner, Norway. *SPE Annual Technical Conference and Exhibition, 19-22 September 2010, Florence, Italy*, doi:10.2118/134891-MS, sPE 134891-MS.
Span, R. and Wagner, W. [1996] A new equation of state for carbon dioxide covering the fluid region from triple-point temperature to 1100 K at pressures up to 800 MPa. *J. Phys. Chem. Ref. Data*, **25**(6), 1509–1597.
Thibeau, S. and Mucha, V. [2011] Have we overestimated saline aquifer CO2 storage capacities? *Oil Gas Sci. Technol. – Rev. IFP Energies nouvelles*, **66**(1), 81–92, doi:10.2516/ogst/2011004.
UNFCCC [2011] `http://unfccc.int/di/FlexibleQueries.do`, query parameters: Europe27, 2011, net emissions excluding land use, CO2 only, total for categories 1.A.1 (Energy Industries) and 1.A.2 (Manufacturing Industries and Construction).

Paper VIII

An Open-Source Toolchain for Simulation and Optimization of Aquifer-Wide CO_2 Storage

Odd A. Andersen, Knut-Andreas Lie, Halvor M. Nilsen
Energy Procedia, **86**, (2016)

VIII

Available online at www.sciencedirect.com

ScienceDirect

Energy Procedia 86 (2016) 324 – 333

Energy Procedia

The 8th Trondheim Conference on CO_2 Capture, Transport and Storage

An Open-Source Toolchain for Simulation and Optimization of Aquifer-Wide CO_2 Storage

Odd Andersen[a]*, Knut-Andreas Lie[a], Halvor Møll Nilsen[a]

[a]*SINTEF ICT, Department of Applied Mathematics, P.O.Box 124 Blindern, N-0314 Oslo, Norway*

Abstract

Planning and execution of large-scale, aquifer-wide CO_2 storage operations will require extensive use of computer modelling and simulations. The required computational tools will vary depending on the physical characteristics of the targeted aquifer, the stage of the project, and the questions asked, which may not always be anticipated in advance. In this paper, we argue that a one-size-fits-all simulation tool for the modelling CO_2 storage does not exist. Instead, we propose an integrated toolchain of computational methods that can be used in a flexible way to set up adaptive workflows. Although a complete toolchain will require computational methods at all levels of complexity, we further argue that lightweight methods play a particularly important role in addressing many of the relevant questions. We have implemented a number of such simplified methods in MRST-co2lab, a separate module to the open-source MATLAB Reservoir Simulation Toolbox (MRST). In particular, MRST-co2lab contains percolation-type methods that within seconds can identify structural traps and their catchment areas and compute spill paths and rough estimates of trapping capacity. The software also offers two-phase simulators based on vertical-equilibrium assumptions that can forecast structural, residual, and solubility trapping in a thousand-year perspective and are orders of magnitude faster than traditional 3D simulators. Herein, we apply these methods on realistic, large-scale datasets to demonstrate their capabilities and show how they can be used in combination to address optimal use of open aquifers for large-scale storage.

Peer-review under responsibility of the Programme Chair of the 8th Trondheim Conference on CO_2 Capture, Transport and Storage

Keywords: CO2 storage; modeling workflow; open-source software; simulation; vertical equilibrium; spill-point analysis; nonlinear optimization; migration; Bravo Dome; thermal effects

* Corresponding author. Tel.: +47-91785380;
E-mail address: odd.andersen@sintef.no

Peer-review under responsibility of the Programme Chair of the 8th Trondheim Conference on CO_2 Capture, Transport and Storage
doi:10.1016/j.egypro.2016.01.033

1. Introduction

If carbon capture and storage is to play a significant role in the mitigation of European greenhouse gas emissions, hundreds of megatonnes of CO_2 must be injected annually into geological formations for permanent storage. Operations at this scale represent a considerable scale-up from current practice and experience, and will require extensive mapping, analysis, planning, and monitoring of the selected storage sites to maintain sufficient confidence in their large-scale and long-term storage properties. The spatial and temporal scales involved are huge: tens of thousands of square kilometers and potentially thousands of years of migration for a typical large aquifer. Computer simulations will be indispensable in this process, from initial screening and ranking, through strategic injection planning, to day-to-day operation and subsequent long-term monitoring. However, each stage of analysis requires different computational methods and tools. Tools for rapid estimation of storage capacities and injection properties will play a key role in the initial exploration stage, whereas more advanced simulation tools will be needed to address actual utilization and realization of available storage potential. A large number of simulations are needed to explore the range of likely outcomes since data even in the best case will be limited and uncertain, and it is therefore necessary to minimize computational costs, particularly when forecasting long-term outcomes, with CO_2 migrating for thousands of years after operations have ceased. Computationally inexpensive model that can be gradually refined will also enable the use of mathematical optimization tools to guide injection planning and operations. During the operational phase, more detailed and computationally costly 3D models are necessary to study the local and short-term interplay among pressure buildup and multiphase, geomechanical, thermal, and geochemical effects, and can also be used to perform continuous integration of new data.

Potential storage sites will vary in operational constraints and in physical characteristics [1] such as seal quality, caprock shape and inclination, aquifer boundaries, rock properties, pressure regimes, thermal conditions, geochemistry, and impact of effects at the pore-scale like fluid relative permeabilities, capillary pressure, and dissolution into brine. Computer modelling and analysis should therefore be adapted on a case-by-case basis to adequately account for the relevant physical effects.

To address the considerations above, we have developed an integrated, open-source collection of software tools for investigating CO_2 storage at the basin-scale. These tools can be used in isolation but are developed to work together using common data structures, plotting and support routines to analyze large-scale aquifer systems in terms of storage capacities or to optimized utilization in terms of pressure buildup, leakage risk, and long-term migration and trapping. The open-source MATLAB Reservoir Simulation Toolbox [3] contains a separate module, MRST-co2lab [2], which emphasizes simplified models, for two reasons. First, whereas conventional 3D simulation is important to understand local development during the operational phase, such capabilities are already available in well-established simulators [4,5,6] and in other modules of MRST [27]. Secondly, computationally lightweight approaches have the ability to rapidly run a large number of simulations on large spatial and temporal domains, which not only enables more extensive exploration of uncertainties, but also makes tasks such as scenario specifications and continuous integration of new data amenable to nonlinear optimization algorithms [7]. An additional advantage is the higher degree of user interactivity compared with what is possible using full 3D models. Finally, we remark that all examples presented in the following are based on real aquifer models provided by the Norwegian Petroleum Directorate (NPD) [8] and the University of Texas at Austin [9].

2. Methods based on spill-point analysis of the aquifer caprock

The movement of CO_2 within a saline aquifer is driven by viscous, gravity and capillary forces. Viscous forces dominate flow close to the injection point, but diminish by distance and become negligible at some point compared to gravity and capillary effects if background flow is not present. After injection ceases, pressure buildup around the injection point will gradually dissipate, further reducing the role of pressure-driven CO_2 flow. When flow is primarily driven by gravity and capillary forces, simulations based on invasion-percolation theory can often provide a good description of CO_2 migration, as demonstrated for Sleipner [10].

Similar results can be obtained by assuming infinitesimal flow rate so that only gravity forces are taken into account. Because of the considerable density difference, the injected CO_2 will form a separate plume that starts to migrate upwards following the steepest ascent of the local caprock slope. When a local maximum is reached, CO_2

will start to accumulate and fill up the surrounding area. As CO_2 accumulates beneath the caprock in this *structural trap*, the local CO_2/water contact is pushed further down until the contact plane reaches a depth at which an alternative upward direction exists. CO_2 will then stop accumulating and any further inflow of CO_2 will exit the trap from the encountered *spill-point* and continue to migrate upwards along the steepest ascent until another local maximum is encountered, and so on. The expected migration from any given point in the aquifer can be predicted by analyzing caprock topography. This *spill-point analysis* provides the location, size, and shape of structural traps, spill-point(s) and spill paths in the upslope direction, and catchment areas (*spill regions*) associated with structural traps.

The modelled process has an analog in hydrology, where water catchment areas lead into lakes, which again spill out into rivers, leading to new lakes downstream, etc. Likewise, just as a detailed numerical simulation of fluid flow is not required to predict where rainfall eventually flows and accumulates, a spill-point analysis can give a good idea of where injected CO_2 will end up. The method also provides valuable information for assessment of storage capacity, as structural traps in the caprock represent the most easily exploitable trapping capacity in an open aquifer. On the other hand, spill-point analysis cannot forecast other trapping mechanisms (residual, dissolution), nor does it take into account non-geometric information and heterogeneous rock properties.

In MRST-co2lab, we have implemented two separate algorithms to run spill-point analysis on discrete, quadrangular 2D-meshes that either use cell centroids or cell vertices to representing the caprock surface [11]. Whereas the former is consistent with the way simulation grids are interpreted in a numerical simulation, the latter produces slightly more precise results. Each algorithm identifies all trap cells, spill regions and spill-point depths, as well as topological information such as connectivity between traps and the exact spill paths ("rivers") followed. Both algorithms are very fast, typically requiring only a few seconds for a grid with 100.000 cells. Once this information has been computed, the exact spill path and traps that would be reached from any hypothetical CO_2 injection point in the aquifer can be instantly computed and presented to the user in an interactive visualization tool. MRST also offers other lightweight tools for computing connected volumes, communication patterns, and timelines for fluid transport in heterogeneous 3D models (see [28]), but these are not discussed herein.

As an example, we have applied spill-point analysis on a simulation grid of the Statfjord formation, a large 150 by 300 km aquifer off the western coast of Norway at a depth of 1700-9000 m, identified by the NPD as a potential CO_2 storage site [8]. Using MRST-co2lab, we have constructed a simulation grid from the data provided by NPD as shown n the top plot of Figure 1. The location of the largest identified structural trap is displayed in dark brown, while the corresponding spill region that extends well beyond the trap itself is shown in light brown. The lower plot shows a complete system of structural traps and spill paths, superposed on a topographic map of the caprock. An

Fig. 1. Structural traps in the Statfjord formation. *Left:* Topographic map of caprock, with traps and spill paths. *Middle:* 3D plot with largest trap and associated spill region highlighted. *Right:* Traps coloured according to CO2 storage capacity (megatonnes).

intricate system of “rivers” and “lakes” can be observed. In some locations, injected CO_2 will migrate through a whole chain of structural traps before reaching the uppermost (eastern) boundary of the formation. The lower right plot reports estimates of the total trapping capacity each structural trap, obtained by multiplying the identified trap volumes with assumed porosity for the rock and density values of CO_2 that depend on local temperature and pressure conditions (assuming hydrostatic pressure and a constant temperature gradient). We note a large variation in trapping capacity, with three particularly large traps in the northern half. The largest has a capacity of 1.45 Gt, which is about the same as the two second-largest combined. Adding together all traps, the total storage capacity in structural traps for the aquifer is just short of 7 Gt. However, these figures are sensitive to inaccuracies in the caprock topography as well as our assumed values of rock porosity, local temperature, and pressure conditions. Altogether, the total computational time spent on the trap analysis was 4 seconds, less than the time it took to load and prepare the grid data from disk.

3. Methods based on a vertical-equilibrium assumption

As noted above, the use of spill-point analysis can rapidly provide rough capacity estimates and establish migration patterns that are useful for addressing several questions related to CO_2 storage. Nonetheless, to forecast the combined long-term effect of structural, residual, and solubility trapping, more comprehensive numerical simulations are required. State of the art conventional 3D simulators are highly sophisticated, but are not designed to handle grid models covering tens of thousands of square kilometres and simulations with timespans extending thousands of years. Moreover, under most relevant aquifer conditions, density and viscosity differences between CO_2 and the resident brine will quickly lead to phase segregation and fingering phenomena, in which a buoyant plume of CO_2 migrates as a thin wedge between the caprock and the brine zone below. Simulation grids with very high vertical resolution are therefore needed to capture the behaviour of the thin CO_2 layer, which increases computational costs and imposes severe time-step restrictions. Sufficient lateral resolution is also needed to model caprock undulations and their effect on gravity-driven flow. Computational times therefore quickly become prohibitive for addressing questions that require a large number of separate simulations.

In recent years, vertical-equilibrium (VE) models originating from the 1950’s when computing resources were scarce have been reintroduced and adapted to the CO_2 storage problem to answer the associated computational challenges [12]. The key assumption for such models is that the fluid phases are in vertical hydrostatic equilibrium and fully segregated at all times. The injected CO_2 is assumed to form a separate plume above the brine phase, which has proved to be an excellent approximation for the aspect ratios and time scales generally considered. A number of studies, both on benchmark problems and real models, have shown that results from VE simulations compare well with full 3D ones [13, 14, 15, 16]. VE models not only reduce the dimensionality and the number of grid cells, but also contribute to reduce coupling and differences in time constants in the flow model. The lower plot in Figure 2 shows a 3D reservoir model grid of the Sleipner injection site whose general shape is close to that of a thin and almost flat sheet. The third dimension, with a vertical resolution of 34 grid cells, can only be discerned up close, and it is clear that the long-term migration primarily will take place in the lateral direction. A 2D grid representing the same domain with a much smaller number of cells, can be used for VE simulations that preserve the

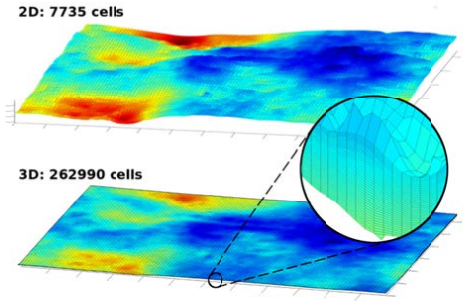


Fig. 2. Simulation grid model of the Sleipner injection site. Top: 2D model. Bottom: 3D model with correct aspect ratio.

fluid distribution profile, relative permeabilities, and rock heterogeneities from the 3D model in an integrated sense. Such VE models are particularly well suited to study large-scale, long-term migration scenarios, addressing questions related to capacity estimation, pressure build-up and the trapping state of CO_2 over time (inventories). The ability to rapidly simulate a large number of scenarios is very useful for assessing plausible outcomes in the face of scarce data, for interpretation/integration of new data in an evolving model, and for optimizing injection scenarios and operational parameters.

Another advantage of VE models is their flexibility in modelling physical effects. Although the basic formulation is simple, it can easily be extended to include a wide range of more advanced phenomena that affect two-phase flow. Recent literature has demonstrated the inclusion and impact of effects such as capillarity [17, 18], residual saturations and hysteresis [12, 19], dissolution and convective mixing [19, 20], subscale caprock topography [21, 22, 23], compressibility [24], brine leakage through caprock [12], geomechanics [25], and thermal effects [26].

In MRST-co2lab, we have implemented VE models that support most of the effects mentioned above, including capillarity, residual trapping, subscale caprock topography, dissolution, compressibility and heat transport, with geomechanical coupling currently under development. In addition, the software contains tools to create vertically integrated simulation grids from 3D stratigraphic models or depth and thickness maps [8], as well as fluid objects with density, viscosity, and enthalpy that depend on temperature and pressure. A finite-volume, fully-implicit numerical scheme is used to discretize the VE equations. The discretization is implemented using automatic differentiation [27], which provides the associated Jacobians of the system with no additional programming effort and hence enables rapid prototyping of new fluid models, as well as providing the necessary information for adjoint-based optimization, as exemplified in Section 4.To demonstrate the use of VE models with additional physical effects, we present two examples. Figure 3 shows the potential impact of CO_2 dissolution for a 1000 year migration scenario following the injection of CO_2 at three separate locations of the Bryne aquifer [8]. The upper left plot

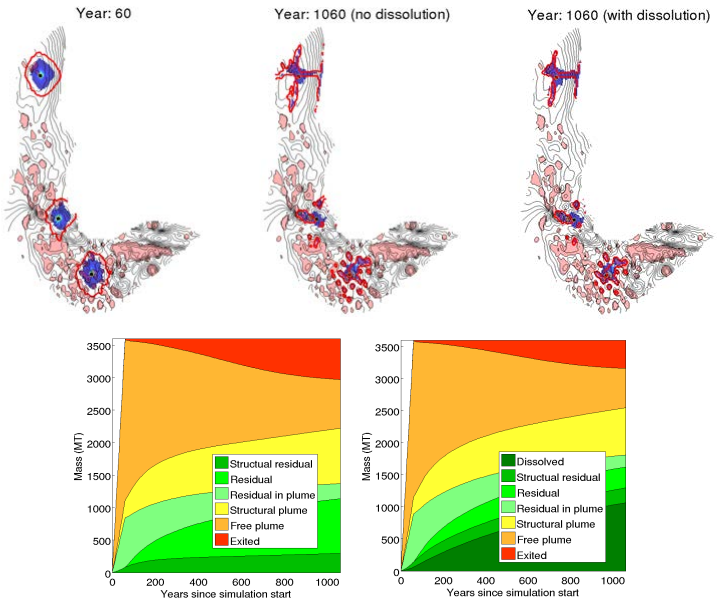


Fig. 3. Simulation of CO2 migration in the Bryne aquifer. Top row: Presence of CO_2 in aquifer just after injection end (left), after 1000 years, assuming no dissolution (middle), and after 1000 years, including dissolution (right). Red contour indicates edge of free-flowing CO_2 plume. Regions containing significant amounts of CO_2 (whether or not mobile) shown in blue. Bottom row: Trapping state of injected CO_2 as a function of time for the scenario without dissolution (left) and with dissolution (right).

shows a topographic map of the aquifer after 3.5 Gt has been injected over a 60-year period. Structural traps are indicated in pink, wells shown as black dots, the mobile CO_2 plume front traced in red, and regions with significant amounts of CO_2 (mobile or not) are coloured in blue. Whereas the two southern wells are positioned in regions with many large structural traps, the northernmost well is situated in a slope with few traps. The two following plots show the state of the aquifer after 1000 years of migration, simulated without and with inclusion of CO_2 dissolution. In both cases, most of the CO_2 injected in the two southern wells collects in nearby structural traps, with the remaining mobile plumes being smaller when dissolution is included. A larger impact can however be seen for the northernmost plume, which mostly does not collect in traps, but spreads out, moves upslope, and exits the simulated domain through the open eastern boundary.

When dissolution is included, the mobile CO_2 plume shrinks faster, more CO_2 is locally retained, and as a consequence, there is less migration out of the domain. The state of injected CO_2 at any point in time during migration is summarized in the two inventory plots in the second row of Figure 3. The coloured zones represent trapping states, with CO_2 exited across domain boundaries in red, the free and migrating plume shown in orange, and CO_2 immobilized by structural, residual, or solubility trapping shown in shades of green and yellow. We see on the rightmost plot how the impact of dissolution grows with time, having dissolved almost 30 percent of all injected CO_2 by the end of the simulation. Migration across boundaries is also significantly reduced compared to the case without dissolution – this is largely due to the impact on migration from the northernmost injection site.

In Figure 4, we show a very different example, where the basic VE model has been extended with energy conservation and conduction of heat. The scenario considered here is the intrusion of hot CO_2 of magmatic origin into a saline aquifer, assumed to have led to the creation of the present-day Bravo Dome natural CO_2 field in New Mexico [9]. In the figure, we show the end result of modelling 1000 years of continuous CO_2 intrusion from a point source close to the western boundary. For this scenario, the entering CO_2 has a temperature of 300 ºC, whereas ambient temperature is set to 20 ºC. Heat is transported with the CO_2 and diffused through the rock. Fluid flow limited to the aquifer itself and modelled using a VE formulation, whereas diffusion of heat is modelled on an extended grid that includes the over/underburden. The resulting discrete equations are solved as a fully coupled system. From the plots, we can see that whereas CO_2 spreads out widely in the aquifer, most of the introduced heat remains relatively close to the injection site. It should however be noted that this example is purely intended to demonstrate our modelling approach, and the choice of parameters should not be taken to accurately represent the historical event, which is still being studied.

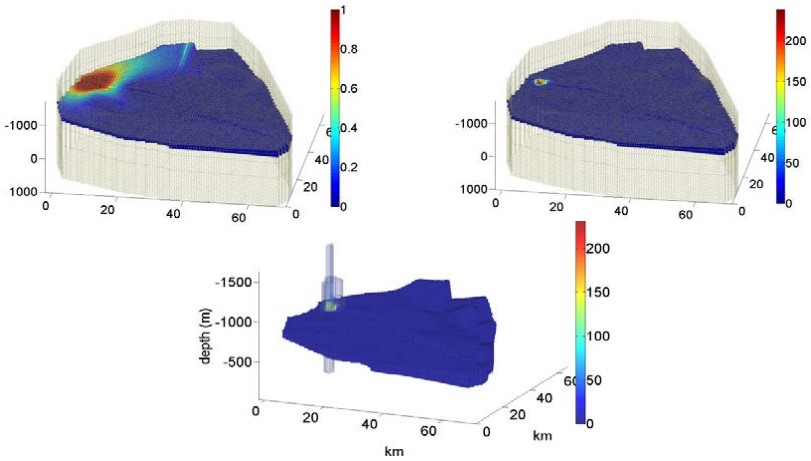


Fig. 4. Coupled thermal and two-phase flow simulation of hot CO_2 intrusion into the Bravo Dome field. *Upper left:* CO_2 saturation after 1000 years of injection. *Upper right:* Temperature increase (°K) in aquifer after 1000 years of injection. *Lower:* Heat transfer into overburden (°K).

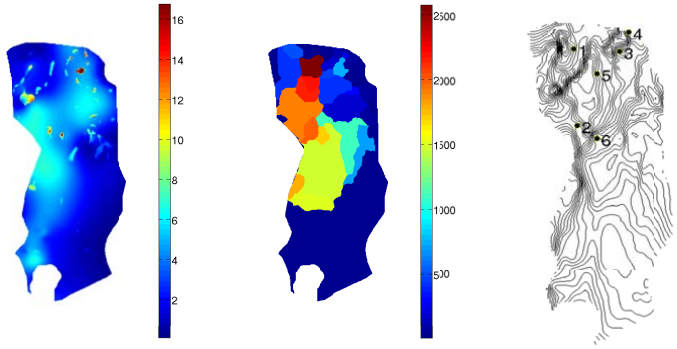


Fig. 5. Storage capacity of the Statfjord formation. *Left:* Map of combined total storage capacity from residual, structural, and solubility trapping mechanisms [megatonnes/m^2]. *Middle:* Map of structural capacity reachable by gravity-driven migration from any given point [megatonnes]. *Right:* Injection locations chosen to maximise amount of structural trapping reached.

4. Outline of an integrated toolchain

By "integrated toolchain", we here refer to the idea that individual modelling methods, tools, data-objects and other MRST functionality can be combined in a flexible way to investigate larger questions related to CO_2 storage. The details of the resulting workflow will necessarily depend on the questions asked, the underlying assumptions made, and the particularities of any specific site. We here exemplify the idea though an approach that integrates the use of EOS-based fluid properties, spill-point analysis, VE simulations, and nonlinear optimization to investigate storage capacity and optimal utilization of the Statfjord aquifer in a large-scale, multi-site injection scenario.

The left plot of Figure 5 displays the local, combined trapping capacity of the Statfjord aquifer, taking structural, residual and dissolution into account, using the methodology of [29,30] and considering the impact of local reservoir conditions on the density of injected CO_2. Adding up the numbers, a total of 95 Gt of CO_2 could be retained in the aquifer by the mechanism of structural (6.95 Gt), residual (69.6 Gt), and dissolution (18.3 Gt) trapping. However, these values are not realistically achievable as they assume 100% sweep efficiency. Structural trapping is the only mechanism that is independent of time and sweep, and can as such be considered the most easily exploitable component of the total capacity. Even so, not all traps can be reached since there will be constraints on the number of injection sites for any real scenario.

One strategy for placing wells could be to maximize utilization of structural traps. The middle plot of Figure 5 displays the maximum mass that can accumulate in structural traps as a result of gravity-driven migration from any specific injection point in the aquifer. Appling the "greedy" algorithm described in [11] to select optimal placement of six wells causes all wells to be placed in the northern half of the aquifer (see Figure 5), where structural traps are larger and more abundant than in the south. Since the algorithm does not take into account other trapping mechanisms, the southern half of the aquifer ends up underutilized, and the method should primarily be used for guidance. Nevertheless, for the purpose of the rest of this section, we will use this well configuration as a starting point to determine injection rates that will optimize use of both structural *and* residual trapping. For any choice of rates, VE simulations will provide a detailed inventory of the state of injected CO_2, as seen in Figure 3. We can thus define an *objective function* that takes a choice of well rates as input and computes a single value representing the "desirability" of the resulting outcome. As our implementation of the VE equations is based on automatic differentiation [27], the Jacobian of the system equations is always available. We can therefore use a continuous, adjoint-based, nonlinear optimization algorithm already available in MRST to determine well rates that represent a local maximum of this function. To this end, VE simulations are essential since any general nonlinear optimization procedure will usually require many evaluations of the objective function, which again necessitates a complete simulation of the migration scenario

We assume that the Statfjord aquifer has open boundaries with constant pressure and aim to maximize the total amount of CO2 injected over 50 years, while minimizing the amount of CO2 that leaves the domain over the next 2900 years. Our objective function thus equals the amount of injected CO2, minus the amount of CO2 that eventually exits the domain multiplied by a penalization factor of ten.

As a starting point for the nonlinear optimization algorithm, we choose injection rates for each well that correspond to the amount of structural capacity reachable from that well (see above). These rates are indicated as blue bars in the upper-right plot of Figure 6, while red bars show the new rates obtained by the optimization procedure. These rates are sometimes quite different from the initial suggestion. In particular, the injection rate of Well 1 has been adjusted drastically downwards. The reason why can be inferred by examining the left plot in Figure 6. During injection, the flow is mainly driven by pressure, and not gravity as assumed by the spill-point analysis. Rather than directly migrating upslope, the CO_2 from Well 1 spreads out into an oblong shape, whose front already touches the northern boundary at the end of the injection period. It is clear that any higher injection rate would lead to immediate migration across the boundaries. The difference between initial and optimal injection rates for the other wells should be understood as a consequence of the effects of residual saturation and pressure-driven flow, which are not captured by the spill-point analysis. The CO_2 inventory from the simulation with optimized rates is presented in the lower right plot of Figure 6. We see that a total of 2.9 Gt is injected, and only a tiny fraction of this amount ends up leaving the domain.

We emphasize that our objective function was chosen subject to the assumption of an open aquifer. If we instead had considered the aquifer to be closed, a more important objective might be to prevent excessive pressure buildup. We could then have specified an objective function that penalizes large pressures, or pressures above some given threshold. Such an objective function is easy to define since pressure is explicitly solved for in the simulation. Moreover, if we had introduced production wells to limit pressure buildup, their production rates could easily be included as additional unknowns in the optimization problem. Linear constraints on the input variables can also be specified so that, for instance, the amount of water produced should balance the amount of CO_2 injected. Finally, we remark that the optimization algorithm used can be easily extended to include time-dependent well rates or other effects such as dissolution. This, however, comes at a higher computational cost.

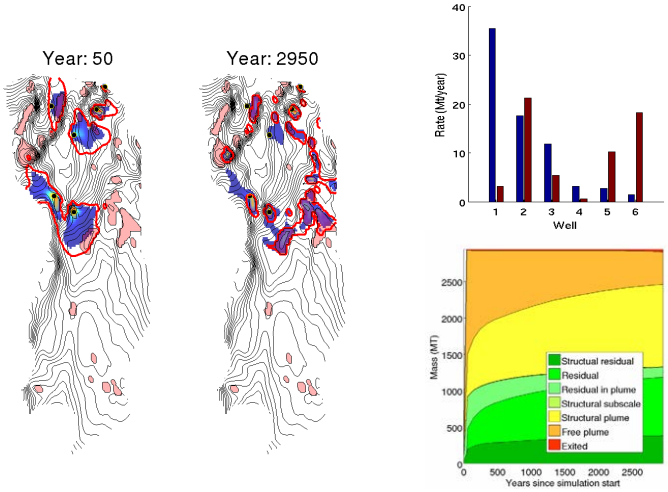


Fig.6. Optimization of injection rates for a 6-well storage scenario in the Statfjord Formation. Left: Presence of CO_2 in aquifer after end of injection, and after end of simulated migration period. Red contour indicates edge of free-flowing CO_2 plume. Regions containing significant amounts of CO_2 (whether or not mobile) shown in blue. Upper right: Initial (blue) and optimized (red) injection rates. Bottom right: Trapping state of injected CO_2 as a function of time.

5. Conclusion

Modelling and understanding CO_2 storage is a highly complex, nonlinear, multiscale, multiphase problem and any computational forecast will necessarily rely on a large number of assumptions, given the limited amount of input data that realistically can be obtained for a given site. In this paper, we suggest that the problem is best approached using a number of methods with different strengths, to be applied independently or combined as components in an integrated workflow consisting of tools that adapt to the specific scenario and questions asked. Figure 7 shows one example of such a workflow. We further emphasize the importance of simplified methods with low computational cost, discuss two such methods, namely spill-point analysis and VE simulation, and demonstrated by example how these can act as components in a more comprehensive analysis. However, we do not intend to suggest that they form a complete toolchain by themselves, nor that other simplified methods are not important in the context of CO_2 storage. A well-equipped toolchain would also include sophisticated 3D modelling tools, as well as a large range of other methods, e.g., statistical tools for assessing parameter uncertainty and model sensitivity, single-phase solvers to assess pressure build-up, flow diagnostic tools that identify volumetric connections and timelines for multiphase displacements, and specialized tools for history-matching and integration of new data. While not a part of MRST-co2lab, several of these tools are available in other modules of MRST.

Acknowledgements

The research has been funded by the Research Council of Norway through grants no. 215641 and 243729 from the CLIMIT programme. The authors thank our collaborators Prof. Marc Hesse and Dr. Kiran Sathaye (UT Austin) for helpful discussions and for providing data of the Bravo Dome.

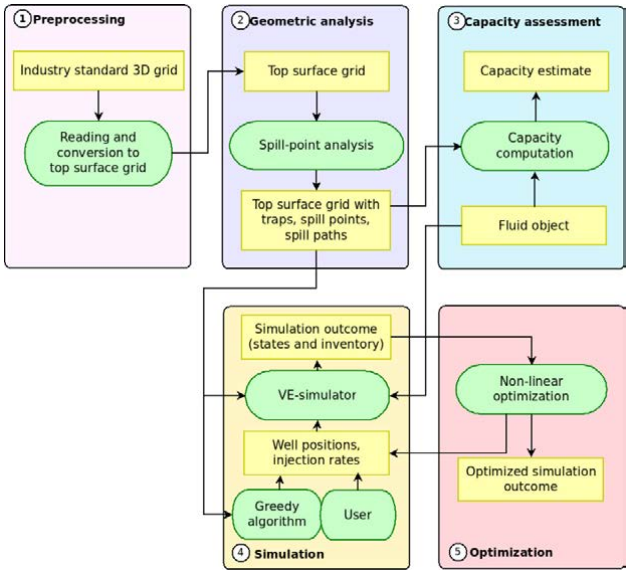


Fig.7. Flow chart for the type of workflow discussed herein divided into stages of analysis, starting with the preprocessing of 3D aquifer models from industry-standard input and ending with the proposal of optimal injection points and rates. Yellow boxes are data objects (specified input or computed output), while green ovals represent data-processing stages (software and user).

References

[1] O. Eiken, P. Ringrose, C. Hermanrud, B. Nazarian, T. A. Torp and L.. Lessons learned from 14 years of CCS operations: Sleipner, In Salah and Snøhvit. Energy Procedia, 4:5541-5548, 2011. DOI: 10.1016/j.egypro.2011.02.541

[2] The MATLAB Reservoir Simulation Toolbox: Numerical CO2 Laboratory. URL: http://www.sintef.no/co2lab (May 2015)

[3] The MATLAB Reservoir Simulation Toolbox, version 2014a, http://www.sintef.no/MRST/ (May 2015)

[4] ECLIPSE Technical Description, Schlumberger (2010).

[5] K. Pruess. The TOUGH codes - a family of simulation tools for multiphase flow and transport processes in permeable media. Vadose Zone Journal, 3:738-746, 2004

[6] B. Flemisch, J. Fritz, R. Helmig, J. Niessner and B. Wohlmuth. DuMux: a multi-scale multi-physics toolbox for flow and transport processes in porous media. In: ECCOMAS Thematic Conference on Multiscale Computational Methods for Solids and Fluids, 2007.

[7] J.D. Jansen. Adjoint-based optimization of multi-phase flow through porous media – a review. Computers & Fluids 46 (1, SI):40-51, 2011. DOI: 10.1016/j.compfluid.2010.09.039

[8] E.K. Halland, W.T. Johansen and F. Riis (Eds.). CO2 Storage Atlas: Norwegian North Sea, Norwegian Petroleum Directorate, P.O. Box 600, NO-4003 Stavanger, Norway, 2011. URL: https://www.npd.no/no/Publikasjoner/Rapporter/CO2-lagringsatlas/

[9] K.J. Sathaye, M.A. Hesse, M. Cassidy and D.F. Stockli. Constraints on the magnitude and rate of CO2 dissolution at Bravo Dome natural gas field. PNAS, 111(43) 15332-15337, 2014. DOI: 10.1073/pnas.1406076111

[10] V. Singh, A. Cavanagh, H. Hansen, B. Nazarian, M. Iding and P. Ringrose. Reservoir modeling of CO2 plume behavior calibrated against monitoring data from Sleipner, Norway. In: SPE Annual Technical Conference and Exhibition, 19-22 September 2010, Florence, Italy, 2010, SPE 134891-MS. DOI: 10.2118/134891-MS

[11] H. M. Nilsen, K.-A. Lie, O. Møyner, and O. Andersen. Spill-point analysis and structural trapping capacity in saline aquifers using MRST-co2lab. Computers & Geoscience, 75:33-43, 2015. DOI: 10.1016/j.cageo.2014.11.002

[12] J. Nordbotten and M.A. Celia. Geological Storage of CO2: Modeling Approaches for Large-Scale Simulation. Wiley, Hoboken (2012)

[13] K.W. Bandilla, M.A. Celia and E.Leister. Impact of Model Complexity on CO2 plume modeling at Sleipner. Energy Procedia, 63:3405-3415, 2014, DOI: 10.1016/j.egypro.2014.11.369.

[14] I. Ligaarden and H. Nilsen. Numerical aspects of using vertical equilibrium models for simulating CO2 sequestration. In: 12th European Conference on the Mathematics of Oil Recovery, 2010. DOI: 10.3997/2214-4609.20145011

[15] H. M. Nilsen et al.. Field-case simulation of CO2-plume migration using vertical-equilibrium models. Energi Procedia, 4:3801-3808, 2011. DOI: 10.1016/j.egypro.2011.02.315

[16] H. Class et al. A benchmark study on problems related to CO2 storage in geologic formations. Computational Geosciences., 13(4):409 434, 2009. DOI: 10.1007/s10596-009-9146-x

[17] J.M. Nordbotten and H.K. Dahle. Impact of the capillary fringe in vertically integrated models for CO2 storage. Water Resources Research, 47(2), 2011. DOI: 10.1029/2009WR008958

[18] H.M. Nilsen, K.-A. Lie and O. Andersen. Fully implicit simulation of vertical-equilibrium models with hysteresis and capillary fringe, 2015

[19] H.M. Nilsen, K.-A. Lie and O. Andersen. Robust simulation of sharp-interface models for fast estimation of CO2 trapping capacity, 2015..

[20] S.E. Gasda, J.M. Nordbottem and M.A. Celia. Vertically averaged approaches for CO2 migration with solubility trapping. Water Resources Research, 47(5), 2011. DOI: 10.1029/2010WR009075

[21] S.E. Gasda, H.M. Nilsen and H.K. Dahle, Impact of structural heterogeneity on upscaled models for large-scale CO2 migration and trapping in saline aquifers, Advances in Water Resources, 62:520-532, 2013. DOI: 10.1016/j.advwatres.2013.05.003

[22] S.E. Gasda, H.M. Nilsen, H.K. Dahle and W.G. Gray. Effective models for CO2 migration in geological systems with varying topography. Water Resources Research, 48(10), 2012. DOI: 10.1029/2012WR012264

[23] H. M. Nilsen, K.-A. Lie, and O. Andersen. Analysis of CO2 trapping capacities and long-term migration for geological formations in the Norwegian North Sea using MRST-co2lab. Computers & Geoscience, 79:15-26, 2015. DOI: 10.1016/j.cageo.2015.03.001

[24] O. Andersen, S.E. Gasda and H.M. Nilsen. Vertically averaged equations with variable density for flow in porous media. Transport in Porous Media, Vol. 107(1):95-127, 2015. DOI: 10.1007/s11242-014-0427-z

[25] T. Bjørnarå, J.M. Nordbotten, S.S. Mathias and J. Park. Vertically integrated models for coupled flow and deflection in porous media.

[26] S.E. Gasda, W.G.G. Gray and H.K. Dahle. Vertically integrated models with coupled thermal processes. In: ECMOR XIV – 14th European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8-11 September 2014, EAGE, DOI: 10.3997/2214-4609.20141851

[27] S. Krogstad, K.-A. Lie, O. Møyner, H. M. Nilsen, X. Raynaud, and B. Skaflestad. MRST-AD - an open-source framework for rapid prototyping and evaluation of reservoir simulation problems. In: 2015 Reservoir Simulation Symposium, Houston, Texas, USA. DOI: 10.2118/173317-MS

[28] O. Møyner, S. Krogstad, and K.-A. Lie. The application of flow diagnostics for reservoir management. SPE Journal, 20(2):306-323, 2015. DOI: 10.2118/171557-PA

[29] O. Andersen, H.M. Nilsen and K.A. Lie. Reexamining CO2 storage capacity and utilization of the Utsira Formation. In: ECMOR XIV – 14th European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8-11 September 2014, EAGE. DOI: 10.3997/2214-4609.20141809

[30] K.-A. Lie, H. M. Nilsen, O. Andersen, and O. Møyner. A simulation workflow for large-scale CO2 storage in the Norwegian North Sea. Computational Geosciences, 2015. DOI: 10.1007/s10596-015-9487-6