INAUGURAL-DISSERTATION

zur

Erlangung der Doktorwürde

der

Gesamtfakultät für Mathematik, Ingenieur- und Naturwissenschaften

der

Ruprecht-Karls-Universität

Heidelberg

vorgelegt von

Tóth, Michal, Mgr. aus Bratislava, Slowakei

Tag der mündlichen Prüfung:

Numerical Modeling of Chemical Degradation of Concrete

Betreuer: Prof. Dr. Peter Bastian Prof. Dr. Hans Knüpfer

Abstract

Concrete is the most common building material. New admixtures are continuously developed to improve desired traits. However, testing is expensive and many experiments take decades. Numerical modeling is a crucial tool for estimating the qualities of the concrete from accelerated experiments.

Our models simulate a two-phase flow of water and air in a porous medium (concrete), the transport of chemicals present in one or both phases, and chemical reactions that can affect porosity. The first model represents the laboratory experiment on accelerated carbonation of the concrete. The second model encompasses three scenarios from the SiTraM benchmark [Hoo+24].

The model equations form a coupled system of partial differential equations, ordinary differential equations, and algebraic equations. The system is discretized using the finite volume method. Special attention is paid to parts where the system is nonlinear or degenerate. The program is implemented in the simulation framework DUNE.

The results of this work are of numerical character. In the first model, we add a projection step into the nonlinear solver (Newton or RASPEN) to address the convergence issues caused by the strong reaction terms. In the second model, we use complementarity constraints to deal with the phase disappearance.

Zusammenfassung

Beton ist das am häufigsten verwendete Baumaterial. Ständig werden neue Zusatzstoffe entwickelt, um die gewünschten Eigenschaften zu verbessern. Tests sind teuer und Langzeitversuche können so lange dauern, dass es keine Option ist, zu warten, bis sie abgeschlossen sind. Die numerische Modellierung ist ein wichtiges Instrument zur Einschätzung der Betonqualitäten anhand kürzerer Versuche.

Unsere Modelle simulieren den Zweiphasenfluss von Wasser und Luft im porösen Medium (Beton), den Transport von Chemikalien in einer oder beiden Phasen und chemische Reaktionen. Chemische Reaktionen können die Porosität beeinflussen. Das erste Modell stellt das Laborexperiment zur beschleunigten Karbonatisierung des Betons dar. Das zweite Modell umfasst drei Szenarien aus dem SiTraM-Benchmark [Hoo+24].

Die Modellgleichungen bilden ein gekoppeltes System aus partiellen Differentialgleichungen, gewöhnlichen Differentialgleichungen und algebraischen Gleichungen. Das System wird mit der Finite-Volumen-Methode diskretisiert. Besonderes Augenmerk wird auf Teile gelegt, in denen das System nichtlinear oder entartet ist. Das Programm wird im Simulationsframework DUNE implementiert.

Die Ergebnisse dieser Arbeit sind numerischer Natur. Im ersten Modell fügen wir dem nichtlinearen Löser (Newton oder RASPEN) einen Projektionsschritt hinzu, um die Konvergenzprobleme zu lösen, die durch die starken Reaktionsterme verursacht werden. Im zweiten Modell verwenden wir Komplementaritätsbedingungen, um mit dem Verschwinden einzelner Phasen umzugehen.

Abstrakt (Slovensky)

Betón je najpoužívanejší typ stavebného materiálu a nové prímesy sú neustále vyvíjané aby sa zlepšili požadované vlastnosti. Testovanie je však drahé a dlhodobé experimenty môžu trvať dlhšie ako je praktické čakať. Numerické modelovanie je preto nepostrádateľný nástroj pri odhadovaní vlastností betónu z kratších experimentov.

V tejto práci používame modely, ktoré simulujú dvojfázový tok vody a vzduchu v poréznom médiu (betóne), transport chemikálií rozpustených vo vode alebo vo vzduchu a ich chemické reakcie. Chemické reakcie môžu zmeniť porozitu. Naším prvým modelom je laboratórny experiment akcelerovanej karbonizácie betónu. Druhý model zahŕňa tri scenáre zo SiTraM benchmarku [Hoo+24].

Oba modely pozostávajú zo systému parciálnych diferenciálnych rovníc, obyčajných diferenciálnych rovníc a algebraických rovníc. Systém je diskretizovaný metódou konečných objemov a miestam kde je nelineárny alebo degenerovaný venujeme mimoriadnu pozornosť. Program je implementovaný v simulačnom prostredí DUNE.

Výsledky tejto práce sú numerického charakteru. V prvom modeli pridávame projekciu do metódy na riešenie nelineárnych rovníc (Newtonovej alebo RASPEN) aby sme vyriešili problémy s konvergenciou spôsobené silnými členmi z chemicakých reakcií. V druhom modeli používame podmienky komplementarity aby sme zvládli zmiznutie fázy vody či vzduchu.

Acknowledgement

I would like to extend my sincere thanks to those who helped me advance academically. Above all, my supervisor Peter Bastian for his endless support and patience. My mentor Steffen Müthing, who taught me how to use DUNE. Ole Klein, who took over the mantle of my mentor after Steffen left and aided me to improve in programming. Etienne Ahusborde for his helpful advice regarding the SiTraM benchmark, in particular degeneracies caused by the phase disappearance. Chaiyod Kamthorncharoen, who implemented RASPEN solver into DUNE and explained me how it works. And Lukas Holbach, who checked and improved the German version of the abstract.

I wish to thank secretary Felicitas Hirsch for extensive help with administrative tasks. Many thanks to my fellow doctoral students for creating pleasant working atmosphere: Santiago Ospina de Los Ríos, René Hess, Freya Jensen, Dominik Kempf, Panasun Manorost, and Linus Seelinger.

I am very grateful to my family for always being there for me – my mum Gabriela, dad Ondrej, and siblings Tatiana and Ondrej.

I would like to recognize the assistance from graduate school HGS MathComp, which financed the first year of my doctoral studies and German language courses, and backed me in organizing board game sessions. I am also thankful to Heidelberg chapter of SIAM, which organized field trips to many interesting companies.

Lastly, I would like to acknowledge the typing assistant Grammarly. Its feedback helped me find many forgotten commas and articles.

This work uses a Clean Thesis LaTeX template developed by Ricardo Langner.

Contents

1	Intr	oductio	n	1				
	1.1	Histor	y of Concrete	1				
	1.2	Resear	rch on Concrete	2				
	1.3	Nume	rical Modeling	5				
	1.4	Outlin	e	7				
2	From	n Princ	piples to Equations	9				
	2.1	Porous	s Medium	9				
	2.2	Two-P	hase Flow in Porous Media	11				
		2.2.1	Hysteresis	14				
		2.2.2	Choice of Primary Variables	16				
	2.3	Transp	port of Chemical Agents	17				
		2.3.1	Advection	17				
		2.3.2	Diffusion	18				
		2.3.3	Dispersion	18				
	2.4	Concre	ete Composition	20				
		2.4.1	Cement Chemistry	21				
		2.4.2	Degradation of Concrete	24				
		2.4.3	Simulating Chemical Reactions	26				
3	Nun	nerical	Tools	31				
	3.1	Finite	Volume Discretization	31				
		3.1.1	Two-point Flux	34				
		3.1.2	Mechanical Dispersion	37				
		3.1.3	Summary	42				
		3.1.4	Boundary Conditions	44				
		3.1.5	Initial Conditions	45				
	3.2	2 Operator Splitting						
	3.3	Parallelization						
	3.4	Newto	on and RASPEN Solver	49				
		3.4.1	Projected Line Search	50				

		3.4.2 RASPEN Solver	52				
	3.5	Complementarity Constraints	53				
	3.6	Implementation in DUNE	55				
4	Firs	t Model: Leaching and Accelerated Carbonation of Concrete	59				
	4.1	Governing Equations	59				
		4.1.1 Two-Phase Flow in Porous Medium	60				
		4.1.2 Transport of Chemical Agents	62				
		4.1.3 Chemical Reactions	63				
	4.2	Model Setting	66				
	4.3	Discretization	69				
	4.4	Numerical Setup	70				
		4.4.1 Operator-splitting	70				
		4.4.2 Time Stepping	71				
		4.4.3 Solver Parameters	71				
	4.5	Results	72				
		4.5.1 Graphs of Primary Variables	72				
		4.5.2 The Effect of the Projected Line Search	84				
		4.5.3 Performance of Newton and Raspen solver	88				
		4.5.4 Parallel Scaling	89				
	4.6	Remarks on Unfruitful Attempts	95				
5	Seco	cond Model: Multicomponent Multiphase Flow with Phase Disap-					
	pear	rance 97					
	5.1	Governing Equations	98				
		5.1.1 Chemical reactions	01				
		5.1.2 Summary	03				
		5.1.3 Initial and Boundary Conditions	05				
	5.2	Numerical Setup	06				
		5.2.1 Degenerate States	08				
		5.2.2 Alternative Ways of Handling Degeneracies	11				
		5.2.3 Finite Volume Discretization	13				
		5.2.4 Regularization	16				
	5.3	One-dimensional Scenarios Results	17				
		5.3.1 Numerical Convergence	20				
	5.4	Two-dimensional Scenario Results	22				
		5.4.1 Parallel Scaling	31				
	5.5	Extended Chemistry	33				
	5.6	Summary	34				

6	Conclusion	137
Bil	bliography	139

Introduction

In this work, we discuss reactive transport in porous media. More precisely, the porous medium is cement-based material like concrete or hardened cement paste. Transported substances are water, air, and chemicals dissolved in them. The key chemicals in our setting will be calcium dissolved in water and gaseous carbon dioxide, which are the major driving forces behind the two most important reactions in aging concrete: leaching and carbonation.

The scope of this work is rather broad. We cover relevant physical processes, derivation of the model, discretization of the model equations, implementation, and the description of numerical tools used. The main contribution of this work lies in the area of numerical tools.

1.1 History of Concrete

Some form of concrete has been known for millennia. The big factor is the accessibility of the materials used to create cement. All it takes is burning crushed limestone or gypsum and adding water and sand. The first usage was more akin to mortar than what is considered to be concrete today. These cement-like materials were used to bind rocks or bricks, and even as surface treatment for waterproofing. Such techniques were known in Egypt, China, Nabatea, etc. The first concrete by modern standards was used in ancient Rome. Their recipe was unfortunately lost, and the longevity and self-healing properties of Roman concrete are puzzling scientists even today. Recent advances shed some light on its composition [Sey+23] so we might be close to unraveling the ancient recipe.

Truly modern concrete traces its origin to the invention of Portland cement by Joseph Aspdin in 1824 [Asp24]. It is still used in concrete, although the formula was refined and the definition of the Portland cement got more precise. Its use got much more varied too. The invention of reinforced concrete lifted the concrete's biggest weakness, its relatively low tensile strength¹.

Nowadays, concrete is the most widely used building material. It can be found in most buildings and almost every big construction project be it a bridge, tunnel,

¹The tensile strength is about 10% of the compressive strength. Normal concrete's compressive strength ranges from 20 to 40 MPa and its tensile strength from 2 to 5 MPa. High-performance concrete compressive strength can exceed 70 MPa [Tay97].

dam, or skyscraper. It is so common that its manufacturing releases around 5% of the global carbon dioxide emissions² [Col10]. Experiments with cement binders with low carbon footprint are underway [Ama13], but Portland cement is still the most common.

Significant effort is spent on developing cement admixtures. Admixtures form only a few percent of the cement volume but can significantly alter its properties in order to specialize the concrete for specific tasks. The effects can be minor like optimizing the workability and drying time for the given temperature, or delaying the setting if the construction site is far from the concrete mixing station. The effects can also be moderate such as reducing the hardening time, improving the resistance to freeze-thaw cycles, or increasing the durability in a chemically aggressive environment (e.g. coastal areas). The biggest change to the concrete can be made by adding superplasticizers that drastically improve the workability and are a key ingredient in high-performance concrete and self-leveling concrete³.

1.2 Research on Concrete

Concrete is both a wonderful and frustrating material for research. Its strength and versatility make it a part of many projects, and its complex composition allows for tinkering with component ratios and adding new ingredients to the mix. The curing stage is another parameter that can be controlled in experiments and must be accounted for in practice. So there is much space to optimize for desired quality, and all is done with materials that are safe enough that untrained people can make their concrete for home projects.

The frustrating part is the complexity of the concrete. Its creation involves several steps, and each adds a level of uncertainty. The composition of the cement can differ by a few percent even if it comes from the same factory. Perfect mixing does not exist, there will always be some small lumps of cement that water did not reach. Pouring can cause the aggregate to interlock and leave empty spaces inside⁴. Hard-

 $^{^{2}}$ Some portion of the carbon dioxide is reabsorbed as the concrete ages, which can take longer than a century. Good-quality concrete absorbs CO₂ slowly, although absorption rates rise sharply if the concrete is recycled and used in low-quality concrete or as a substitution for gravel in road sub-base.

³Workability can be improved up to the point when gravity is strong enough to level the concrete. High-performance concrete uses superplasticizers to reduce the water content while keeping the workability comparable to the common concrete. Less water results in lower porosity and higher strength of concrete.

⁴So-called honeycombs mostly appear on the surface of the concrete where the sludge touches the mold. Bigger aggregate pieces are compacted nicely, but the space between them is not filled with cement and sand. Honeycombs are common when workability is bad, mostly due to low water content. Vibrating the concrete during the setting prevents the creation of honeycombs to some degree.

ened concrete samples have different pore networks, some might form preferential channels that skew the permeability measurements. Drying causes microcracks and some samples can experience a bigger strength reduction than others. Many factors make results inconsistent for research purposes and the small size of experimental samples makes the differences more pronounced. Experimentalists always create several samples to catch outliers.

Another complication is the inaccuracy of noninvasive measurement methods. Cutting the sample and scanning pieces is the only reliable method to see what is really inside, but it renders the sample unusable. This is a problem, especially in long-term experiments. The cost increases quickly when each sample can be scanned only once, the experiment aims to collect data from several decades, and one has to account for the sample variability [JWG16].

Testing the state of older structures differs from collecting experimental data. Instead of collecting one particular data type, it has to assess the overall condition of the concrete. It can also better combine the results of invasive and noninvasive methods. Noninvasive methods are good at screening some types of faults and destructive cover others.

Noninvasive tests are more common since they are cheaper. Sound waves can be used to spot cracks, voids, and delamination. In short, waves scatter at interfaces so healthy concrete conducts better. Sound-based tests are (from simplest) the rebound hammer test, using radar waves, and ultrasonic pulse velocity measuring. Another noninvasive method is the pullout test. A rod with a wider disc at its end is placed into the concrete in a depth equal to the cylinder diameter (often 25 mm). The rod is then pulled out against a counterpressure and the breaking point is used to estimate the compressive strength. Some structures have the rods preinstalled, otherwise a hole can be drilled for a new disc. Since the holes are so shallow, some authors categorize this test as nondestructive, and others as destructive [MK14].

Destructive methods are used when the information can not be obtained by nondestructive testing. They involve drilling core samples which are then tested in a laboratory. Samples can be used to measure the depth of carbonation and strength. A small advantage is that measurements are direct and do not require calibration. However, drilling inflicts structural damage that needs to be repaired.

Laboratory experiments are mostly destructive. The most common test involves applying an increasing amount of force until the sample is destroyed. How the force is applied, depends on the type of strength measured: compressive, tensile, or flexural (bending). We will now look more into tests that are aligned with the scope of our work, which is the laboratory setting for measuring chemical degradation. Mercury intrusion porosimetry is used to estimate the pore size distribution. Mercury is nonwetting-fluid with a high contact angle. It fills the largest pores first and by increasing the pressure we can fill progressively smaller pores. Unfortunately, it tends to overestimate the volume of small pores due to ink-bottle effect [Dia00] and procedure alters the pore network⁵ [SW85].

Even something as simple as measuring the water retention curve can damage the concrete. The problem lies on the dry end of the curve. Concrete is a hydrated product and removing the last bits of water affects chemical bonds and leads to the formation of microcracks⁶. The special relationship between water and concrete can also be observed when the permeability is measured. Water experiences lower intrinsic permeability than air or liquids like alcohol that are inert to cement [Hal07].

Other measurements involve cutting the sample to reveal the insides. X-ray diffraction tells us the percentages of the most represented molecules. Pictures from scanning electron microscope can be used to calculate the pore size distribution (useful for estimating the permeability [WZB12]) and to identify minerals. For higher resolution, small-angle neutron scattering is used.

Preparation of the samples can take from one month up to a few decades. We need to know both short and long-term behavior. To reduce the duration, experiments are often accelerated, i.e. they use more aggressive acids or higher reactant concentrations than what is common in nature [Eks01]. Unfortunately, some processes can not be accelerated completely. Accelerated leaching is representative of its long-term counterpart⁷, whereas other processes like carbonation lead to different results [Aur+15; Cas+09].

There are even projects that were never tried before and it is too late or outright impossible to collect data. For example, storing nuclear waste relies on stable conditions that last millennia. Carbon sequestration⁸ needs a long time to reach the chemical equilibrium. The need for accurate estimation and the lack of long-term data give rise to numerical modeling. Models are our best tool to bridge gaps in data and to study solutions that have not been tried yet. Numerical modeling is also a way of obtaining results faster and more cheaply. Waiting several decades is too long for the quickly developing industry.

⁵Damage is caused first by high pressures that are used to drive the mercury in. But extraction of the mercury is even worse.

⁶Samples are usually oven-dried to remove trapped water. Parts of concrete are so hydrophilic (in particular C-S-H) that they absorb humidity from air [Sch+16].

⁷See [CEF97], where they compare the leaching due to deionized water with accelerated leaching that uses ammonium nitrate.

⁸Carbon sequestration has a similar chemistry to concrete, although the amount of carbon dioxide is massive. The storage works on three different time scales. The shortest time scale is getting the carbon dioxide into the reservoir. That does not guarantee that it stays there, but some carbon dioxide gets trapped in the pores of rocks which is the intermediate stage. Long-term storage is achieved when carbon dioxide reacts with rocks and becomes a part of the solid phase.

1.3 Numerical Modeling

Models differ greatly in complexity and predictive power. The simplest models, akin to guesstimates, are used in situ for quality assurance. The so-called bucket test involves spilling a (standardized) bucket of fresh concrete on the ground and measuring how quickly the concrete spreads [Rou+16]. The test serves to judge how long the concrete stays workable. Then there are tables that for given spacing of supports prescribe the thickness of building floors. These models are simple and not very precise so they err on the side of caution.

Jobs requiring higher precision justify the higher cost of custom models. Due to the wide variety of models we will limit the overview to models that fall within the same category as ours – reactive transport in porous media. It is still a broad class of problems that, besides the chemical degradation of concrete, contains carbon sequestration, nuclear waste storage, environmental contamination due to DNAPL⁹, and even oil extraction.

There are plenty of open-source packages capable of solving such problems. They are developed by different communities working on their projects, which is reflected in the packages' specialization. We use a module PDELab from the DUNE environment [San20]. We will talk more about the PDELab in the section 3.6. Another package based on DUNE is DuMux [Fle+11], which focuses on multiscale simulations including transport in porous media. The simulation tool PorePy [Kei+21] is designed for fractured porous media and has extra tools to embed the cracks into the grid as lower-dimensional (compared to the grid) objects. The simulator DARTS [Wan+20] is more into petroleum engineering and low enthalpy geothermal operations.

The module PhreeqcRM [PW15] adds reactions to the transport simulator PHREEQC. The coupling is done via an operator splitting technique. Theoretically, the operator splitting approach enables coupling any transport simulator and reaction simulator but interfacing them can be onerous. Reaction modules are often designed to be flexible, although transport modules are not¹⁰.

We purposefully omit proprietary models. Such models are developed together with the company's product which allows them to be more focused and better calibrated than community-developed models. But they are not open source thus we can not easily verify what they do. The connection to the product is also one reason why mentioning (and endorsing) them does not seem appropriate.

⁹Dense Non-Aqueous Phase Liquid. A liquid denser than water that does not dissolve in or mix with water. A contaminated aquifer close to a mining site is a typical problem.

¹⁰Reaction modules solve a system of ordinary differential equations that require only local data (accessible within one mesh cell). Transport simulators developed with reactions in mind already have them.

One big problem in the field is the comparability of models. Specialization makes models excel in solving problems on which they were designed and calibrated, but the calibration hides many errors. Numerical modeling is about controlling errors and there are several layers of them. Model error is the discrepancy between the reality and the mathematical formulation. Discretization error is the deviation of the numerical approximation from the true solution to the formulation¹¹. The measurement error is the uncertainty in the input data.

To compare the performance of models, we need to measure their modeling errors. The main attempt at this is the establishment of benchmarks. The benchmark is a problem or a collection thereof with a solution accepted by the community. Some benchmarks have analytical solutions, but those are exceedingly rare even for greatly simplified problems [LD09]. When the analytical solution is unknown, the benchmark's power lies in the consensus – how many groups attempted to solve the problem (and published it) and how consistent their results are. Typically, a benchmark focuses on one challenge and has a set of problems with increasing difficulty. Specializing on one challenge also means that the remaining aspects of the benchmark problem tend to be simplified. This simplification helps with interpreting results and comparing different approaches. In addition, it is easier for researchers outside the field to participate, and the benchmark can even serve as a stepping stone for people entering the field.

For example, MoMas benchmark [CKK10] comes from the nuclear waste management research area. The main challenge is chemical reactions, thus the domain is quite small with a simple shape, and the flow field is constant throughout the simulation. The benchmark features three scenarios with progressively more difficult chemical reactions. The objective is to get the best performance, which is derived from the computation time¹².

A whole collection of benchmarks is published by the SESBench initiative (Subsurface Environmental Simulation Benchmarks) [SYU15]. Their focus is subsurface environmental problems ranging from microbially mediated reactions, through isotopic fractionation, multi-component diffusion, and metal mobility in areas affected by mining, to problems associated with nuclear waste storage like cement-clay interface, effects of mineral dissolution-precipitation on porosity-tortuosity relationship, and flow in fractures. Benchmarks from this collection are more complex than MoMaS, and the main objective is accuracy, not performance.

¹¹Figuratively speaking. For many mathematical formulations of real-world problems, we do not even have the proof of existence of the solution.

 $^{^{12}}$ In order to reduce the impact of different hardware, the benchmark defines a *CPU unit* as the time it takes to multiply two dense $1\,000 \times 1\,000$ matrices. Results are compared in these hardware-specific units.

We participated in SiTraM's (advances in SImulation of reactive flow and TRAnsport in porous Media) initiative on establishing a new benchmark [Hoo+24]. This benchmark's focus is the multicomponent multiphase reactive flow in porous media, and the main challenge is dealing with the degeneracy caused by phase disappearance. We devote the chapter 5 to this benchmark.

1.4 Outline

The scope of our work is the numerical study of the effects of leaching and carbonation. Our model includes two-phase flow in a porous medium (concrete or cement paste), transport of chemical agents in either or both phases, and chemical reactions that can affect the porosity of the medium. The main challenges of our models are the strong nonlinearity of flow and reactions, and the degeneracy caused by the phase disappearance.

The next chapter describes the model derivation. It encompasses equations and physical laws used to describe our models, as well as a short discussion about neglected effects.

In the third chapter, we look at numerical tools. We start with discretizing the system of equations, continue with the description of methods used to solve the system, and finish with the implementation within the DUNE framework [Bas+21].

Chapter 4 is about the numerical model of leaching and accelerated carbonation of concrete in a laboratory setting. The sample is small, its geometry simple, and boundary conditions constant. Accelerated conditions mean that the carbon dioxide concentration is extremely high: 50% instead of atmospheric 0.04%. The system is reaction-dominant, the main challenge is to resolve efficiently the carbonation front, the thin moving zone where the carbon dioxide reacts with calcium.

Chapter 5 discusses problems that are part of the SiTraM benchmark [Hoo+24]. The benchmark is focused on dealing with the degeneracy caused by the phase disappearance. Similarly to the previous model, the geometry and boundary conditions are simple, but now they lead to the situation when one part of the sample is liquid-saturated, another is gas-saturated, and the rest is unsaturated (both phases are present). The main challenge is to manage the phase disappearance degeneracy.

The last chapter is the conclusion. It recollects our contributions and contemplates their prospects for future research.

The results of this work are of a numerical nature and our contributions likewise. In the chapter 4 we compare the performance of two nonlinear solvers. Our main contribution is the improvement of the convergence achieved by the addition of the projection step into their line search methods.

7

In the chapter 5 we take a more detailed look at our implementation than the publication [Ahu+24] allowed us. Our main contribution lies in the way we are dealing with the phase disappearance degeneracy. We use complementarity constraints to handle the branching caused by the system switching between saturated and unsaturated states. No other team working on the benchmark used this strategy¹³. The models were implemented in the numerical environment DUNE.

¹³Although otherwise approaches were similar. All teams used the cell-centered finite volume scheme, and only one used operator splitting instead of the monolithic approach.

From Principles to Equations

In this chapter, we accompany the reader through the derivation of the model equations. We introduce the notion of the porous medium and the movement of fluids inside, classify considered types of transport of chemical substances present in the pore solution, and conclude with chemical reactions.

2.1 Porous Medium

The porous medium is a solid filled with interconnected pores. These empty spaces allow fluids to flow inside and through the porous medium. Porous media problems range from the microscopic to the megascopic scale and each scale poses a different challenge. Our problems belong to the macroscopic scale.

The microscopic scale is the smallest of scales and the problem size is comparable to the pore size. The fluid flows through voids between grains of the medium that form a very complex boundary. Solving such problems requires detailed knowledge of the pore geometry and scales terribly to bigger domains. Not just because of the sheer size of real-world problems compared to the pore size, but also because the information about the pore geometry is unavailable. Among others, porescale models are useful for deriving equations for macroscopic models through the application of homogenization techniques. For example [BDR12] and [Blu01] model fluid flow at the microscopic scale using Stokes equation on a periodic domain. Using homogenization they show that Darcy flow is a good approximation of the flow at the macroscopic scale. Pore-scale models provide a great insight into how fundamental attributes like pore shape, connectivity, or capillary forces translate into the medium's averaged properties like porosity, permeability, tortuosity, or wettability.

Macroscopic models are one scale above the microscopic models. They are too big to resolve pores individually and only use averaged properties of the porous medium (porosity, saturation, permeability,...). Macroscopic models necessitate that the medium's properties can be averaged, otherwise, the problem can not be parameterized. This is done by assuming the existence of the representative elementary volume (REV) [BC10]. REV is the smallest volume whose properties are representative of the properties of the whole medium. What are the "properties" depends on the model. It can be simply the porosity or permeability but it can be more general too, for example, the rule describing the dependence of the capillary force on the saturation.

Simply said, REV must be large enough that the effects of individual pores average out. The variability of properties of smaller elements is illustrated in Figure 2.1. It shows the maximum and minimum porosity from among 30 randomly positioned cube-shaped elements. The element size, the length of the cube's side, ranges from 0.1 to 2. Porous medium's pores and grains are cubes with sides of length 1 placed in a 3D chessboard pattern. A small element can be placed completely inside a pore or a void so its porosity can be anything from 0 to 1. Larger elements contain several pores and grains in between, so they do not



Fig. 2.1: Variability of REV properties decreases with increasing element's size.

deviate far from average. Real porous media have less regular pore networks with varying pore sizes, so the REV size relative to the pore size will be bigger than our example, but it is still small compared to the sizes used in practice. Hardened cement paste has small pores, the majority of the pore volume consists of pores with diameters from 10 nm to 1 μ m [Dia00]. Concrete's aggregate is not porous in our setting.

Models bigger than macroscopic are called mesoscopic or megascopic. The nomenclature is not very strict and sometimes all three are referred to as macroscopic. One could say that meso- and megascopic models are those that have more than one REV. It happens in problems that contain quantitatively different parts, for example a fractured rock. The rock and the fracture have each flow dynamics with its own REV. The biggest challenge of these problems is interfaces. Getting right the flow through the rock or the crack can be done in isolation, but we need to know both to study how they interact with each other.

2.2 Two-Phase Flow in Porous Media

In our setting, the space inside pores is occupied by two fluids: water and air. Water is a wetting phase, i.e. the contact angle between water and the porous medium is lower than the contact angle between air and the medium. Water covers surfaces of pores and fills small pores first, whereas air is shoved into bigger pores (with surfaces covered by water). This is, with some caveats, the result of fluids minimizing their surface energy.

The equations for two-phase flow in porous media, derived from mass balance laws for water and air, are

$$\partial_t \left(S_w \phi \rho_w \right) - \nabla \cdot \left(\vec{q}_w \rho_w \right) = f_w \tag{2.1}$$

$$\partial_t \left(S_a \phi \rho_a \right) - \nabla \cdot \left(\vec{q}_a \rho_a \right) = f_a. \tag{2.2}$$

 ϕ is the porosity of the medium. S_w , ρ_w , and \vec{q}_w are water saturation, density, and volumetric flux. f_w is the water source or the sink term. S_a , ρ_a , \vec{q}_a , and f_a mark the same for the air phase. The first term of the equation 2.1 and 2.2 represents the change of the mass of water or air within the given volume (which has an infinitesimal size in the strong formulation). The second term represents how much water or air entered or left the volume through its boundary. The flux direction is determined by the pressure gradient and its magnitude also depends on other factors (explained below).

Some pores are dead ends and water or air trapped inside can not leave via pressure-driven flow – capillary forces would not allow it. Water in a small pore will not enter a bigger pore full of air just to continue flowing further. Similarly, an air bubble in a big pore partially filled with water (that encapsulates the air bubble) will not be pushed into smaller pores easily. There are secondary processes that can free trapped water and air but they are far slower and will not be considered in our models. Trapped water can form films on the surfaces of pores enabling it to spread out even when water content is so low that the phase is disconnected. Trapped air can dissolve into water and water can evaporate. It is common in laboratory experiments with cement to accelerate drying, most commonly by putting the sample into an autoclave [Eks01]. Unfortunately, oven-drying (up to $105 \,^{\circ}$ C) damages the sample's microstructure. Other methods like freeze-drying are gentler but it is practically impossible to remove all moisture from the cement without damaging it [KT06].

Trapped water and air do not contribute to the flow, thus formulas often exclude them. The fraction of the pore volume occupied by one phase is its saturation S. The saturation of trapped water or air is called residual saturation and will be denoted by S^r . The effective saturation, for water defined as

$$S_w^e = \frac{S_w - S_w^r}{1 - S_w^r - S_a^r},$$
(2.3)

is water saturation rescaled from the range $(S_w^r, 1 - S_a^r)$ to (0, 1). This filters out the problem-dependent residual saturation and makes it more convenient to use general models. Effective air saturation is derived accordingly. The space in pores is entirely filled with the two phases, therefore $S_w + S_w^r + S_a + S_a^r = 1$.

The flow inside the porous medium is laminar. Fluids are always close to the pore surface which slows them down. The overall direction of the flow is dictated by the pressure gradient. We will use Darcy's law to express the flux:

$$\vec{q} = -\frac{k}{\mu} \left(\nabla p - \rho \vec{g} \right) \tag{2.4}$$

 \vec{q} is the volumetric flux, k the permeability of the porous medium, μ , p, and ρ are the viscosity, the pressure, and the density of the fluid, and \vec{g} is the vector indicating the strength and the direction of the gravitational force. This formula is the same for the water and the air phase, but the permeability can be calculated differently. Lower indices w and a will be used to distinguish the water and the air phase, although they are omitted in many places where the formula applies to either phase.

The permeability k depends on the porosity and the saturation¹. We will split it into two parts; intrinsic permeability K which depends on the porosity ϕ , and relative permeability k_r which depends on the saturation.

$$k = K(\phi) k_r(S) \tag{2.5}$$

The intrinsic permeability represents the ability of the porous medium to allow a fluid to flow through it at its full saturation. The relative permeability then tells us how much the permeability is reduced when the medium is not fully saturated. In most applications, the intrinsic permeability is constant or at least constant in time. We deal with chemical reactions changing the porosity, and intrinsic permeability will reflect this change. The relation between porosity and intrinsic permeability is described through the Kozeny-Carman equation.

$$K(\phi) = K_0 \frac{\phi^3}{\phi_0^3} \frac{(1-\phi_0)^2}{(1-\phi)^2}$$
(2.6)

¹And other factors that are constant throughout the simulation, e.g. tortuosity.

Initial intrinsic permeability K_0 and initial porosity ϕ_0 are constants. They could differ spatially, but it is not common for cementitious materials². However, we have to distinguish the intrinsic permeability of water and air.

Concrete reacts with water and bounds it, which effectively lowers the intrinsic permeability for water by about one to two orders of magnitude [Hal07; Zho+17]. This is specific to water, liquids that do not react with concrete ([Zho+17] mentions organic liquids like ethanol or isopropanol) have the same intrinsic permeability as air.



The relative permeability has different formulas for the wetting (water) and the non-wetting (air) phases. This holds for all porous media and can be ascribed to phases occupying separate spaces in pores. The wetting phase fills the smallest pores and walls of bigger pores. As

Fig. 2.2: Van Genuchten curves for relative permeability with n = 1.65.

the saturation decreases from the water-saturated state water leaves the centers of the biggest pores. Its flow paths become more tortuous and longer. On the other hand, the non-wetting phase leaves the smallest pores first and its flow is confined to the biggest pores. Airflow paths are thus concentrated to a few largest channels.

There are various models for relative permeability. In the chapter 5 we use simple Corey's formula. The water and air relative permeability have the same formula

$$k_{rw} = (S_w^e)^n \quad \text{and} \quad k_{ra} = (S_a^e)^n \tag{2.7}$$

with user-defined parameter n. The parameter reflects the type of the porous medium, chapter 5 uses n = 2.

Chapter 4 uses a more intricate van Genuchten formula [Gen80]. The relative permeability of water is

$$k_{rw} = S_w^{e^{\frac{1}{2}}} \left(1 - \left(1 - S_w^{e^{\frac{1}{m}}} \right)^m \right)^2$$
(2.8)

and the relative permeability of air is

$$k_{ra} = S_a^{e^{\frac{1}{3}}} \left(1 - (1 - S_a^{e})^{\frac{1}{m}} \right)^{2m}.$$
 (2.9)

²Concrete is fairly isotropic with more porous areas just under the reinforcement [Ang+17], and at the surface [Kre84]. Neither area is big enough to be relevant in our models.

Their graphs are shown in Figure 2.2. These formulas are often combined with the van Genuchten-Mualem formula that defines the relation between the saturation and capillary pressure [BC10]. Labeling capillary pressure as p_c , the formula is

$$p_c = p_a - p_w = \frac{1}{\alpha} \left((S_w^e)^{-\frac{1}{m}} - 1 \right)^{\frac{1}{n}}.$$
(2.10)

The constant α represents the inverse air entry pressure³ and is linked to the biggest pores of the network. Constants n and m are related to the distribution of pore sizes. The constant m is commonly tied to n, we use $m = 1 - \frac{1}{n}$.

The relation between capillary pressure and saturation, for which we use the formula 2.10, is called the water retention curve. The capillary pressure is highest when water (the wetting phase) saturation is low. As the saturation increases, areas with the highest binding potential⁴ get filled and capillary pressure decreases.

However, not all areas are accessible to water and air at all saturation levels. A small pore can remain filled with water when surrounding bigger pores get drained, an air bubble can get stuck in the middle of the bigger pore when connecting channels get completely filled with water, etc. Given the right circumstances, such pores can be emptied or filled, but their state does not depend only on the current saturation but also on the previous state – the system has hysteresis.

2.2.1 Hysteresis

Permeability and water retention formulas we use are well-established and commonly used, e.g., in [BC10; BS04; Aur+15]. Nevertheless, they are not capable of capturing the hysteresis. The state of the porous medium depends not only on the current saturation, but also on the past. The main three causes of hysteresis are the ink bottle effect, the raindrop effect, and the snap-off effect [Sch17].

The ink bottle effect, or the bottleneck effect, is a prime example of hysteresis. It occurs in a channel with narrow and wide pores. The drainage (of the wetting phase) is limited by the width of the narrowest pore, whereas the imbibition⁵ (filling) is limited by the width of the widest pore. Under the same pressure conditions, the drainage and the imbibition stop at different places.

The raindrop effect is the dependence of the contact angle on the fluid velocity. The faster fluid flows, the larger (less wettable) the leading contact angle becomes. Different flow speeds then result in different distributions of fluids in pores. The vast

³Some sources try to find the meaning behind those coefficients, and others (including [BC10]) simply describe them as curve fitting coefficients.

⁴Typically small pores where water can cover more surface for given volume.

⁵Drainage and imbibition curves are also called drying and wetting curves.

majority of models, including those in this work, use the steady state approximation where contact angles are fixed - independent of the fluid velocity.

The snap-off effect characterizes the situation when some volume of fluid gets disconnected from the rest of the phase. The snap-off effect is related to the residual saturation because the disconnected volume can not move until it reconnects. The wetting phase can get trapped in narrow openings, and nonwetting in the middle of big pores with smallish exists.

As if things were not complicated enough, hardened cement contains hydrates - minerals that bind water. The humidity changes can open and close pores, which adds another hysteresis effect that is not common in other porous media [Sch+16].

Models incorporating hysteresis are rare in practical applications. There are problems with convergence speed and upscaling [PFB05]. The majority of models require as an input the main drainage and imbibition curve, so experiments mostly limit themselves to measuring those curves [PNM18; Bru+12]. Figure 2.4 illustrates such curves for water in soil-like material. Note that simple draining can not remove all water and filling will not push all air out. Some residual water and air will remain Fig. 2.4: Imbibition and drainage retention trapped.

For given saturation, capillary pres-



Fig. 2.3: Illustration of the ink bottle and snap-off effect. The same boundary pressures lead to different equilibrium solutions if we start with watersaturated or air-saturated pores. Draining can disconnect the water phase.



curve.

sure is higher when water gets drained than when it gets filled. If the saturation does not change from one extreme to the other, the capillary pressure can be anywhere between the drainage and the imbibition curve [KNR07]. Despite the importance of hysteresis, due to its complexity, it is common to neglect hysteresis and

use the water retention curve which depends only on the current saturation [BC10; BS04]. We do likewise.

2.2.2 Choice of Primary Variables

Two-phase flow in a porous media system has two partial differential equations from the mass balance laws. The natural choice is to set two variables as primary and calculate the remaining variables of the system from them. The common choice of primary variables is pressure and saturation of either phase (possibly mixed). The capillary pressure formula and the fact that water and air saturation add to one make it possible to calculate the remaining saturation and pressure variables. Model's formulas (e.g. van Genuchten formula 2.8) are then used to evaluate remaining terms. Using the capillary pressure formula in reverse fashion allows calculating the saturation from the capillary pressure, therefore choosing two pressures (from among water-, air-, and capillary pressure) as primary variables works too. The porosity must be given or, as in our case, have its own equation.

The optimal choice of primary variables depends on the model setting. Water pressure is ill-defined when the water phase disappears and likewise air pressure is ill-defined when the air phase disappears. Water and air saturation are well defined at all times, but one determines the other so only one can be the primary variable. In a setting with high water saturation, it is better to use water pressure as the primary variable. We do this in the chapter 4 where the water saturation is over 0.8 in the whole domain. Water pressure remains well defined even when the air phase disappears, although that does not happen in this setting.

If the water saturation is low and the phase can disappear, it is obviously better to use air pressure as the primary variable. However, we often have to pay attention to the model's formulas, which might need to be regularized or exchanged for some that better fit the data ranges of the setting. For example, the capillary pressure in our water retention curve is ill-suited for water phase disappearance, because 2.10 approaches infinity as water saturation goes to zero.

Being able to handle the disappearance of either phase is difficult. It requires switching variables or generalizing the variable to give it (at least algebraic) meaning in the regime where it is ill-defined. In Chapter 5, a global pressure approach is used, which is in fact a simplification of the model. The wetting and the nonwetting phases use the same pressure, which makes it well-defined in all situations. The global pressure approach sets capillary pressure to zero, so it is applicable only in a situation where the capillary pressure is less important.

2.3 Transport of Chemical Agents

Chemical reactions are an integral part of this work. The reaction rate is tied to the concentration of reactants. We need to know the sources and sinks of chemicals (i.e. boundary conditions), and their transportation patterns. We distinguish three types of transportation of chemical agents: advection, diffusion, and dispersion.

2.3.1 Advection

Advection is the transport caused by the movement of the carrier fluid, dissolved substance has zero velocity relative to the fluid. The simplest form of an advection equation is

$$\partial_t c + \nabla \cdot (\vec{v}c) = 0 \tag{2.11}$$

where c is the unknown (concentration, mass, energy, or other), and \vec{v} is a given velocity field of the fluid.

We are interested in the transport of chemical agents dissolved in water or air inside the unsaturated porous medium. That complicates things a little, but the basic idea stands. The change in mass of the substance within one area is proportional to the amount of substance flowing through its boundary. For a substance dissolved in water the strong form of the equation is

$$\partial_t (\phi S_w \rho_c c) + \nabla \cdot (\vec{q}_w \rho_c c) = 0, \qquad (2.12)$$

where ϕ , S_w , and \vec{q}_w were introduced in section 2.2 as porosity, water saturation, and water volumetric flux. ρ_c and c are the density of the dissolved substance and its concentration.

Using flux is more natural than velocity. Actually, in the simple case 2.11 the flux and the velocity coincide. In a more general case, the relation is given by

$$\vec{q}_w = \phi(S_w - S_w^r) \vec{v}_w.$$
 (2.13)

If we used water velocity instead of flux, the equation 2.12 would be

$$\partial_t (\phi S_w \rho_c c) + \nabla \cdot (\vec{v}_w \phi (S_w - S_w^r) \rho_c c) = 0.$$
(2.14)

The advection equation for a substance dissolved in air is derived in the same way.

$$\partial_t (\phi S_a \rho_c c) + \nabla \cdot (\vec{q}_a \rho_c c) = 0 \tag{2.15}$$

However, there is one hidden difference: air is compressible. The density is not constant, it depends on the partial pressure of the substance c.

2.3.2 Diffusion

Diffusion, more precisely molecular diffusion, is mixing caused by concentration differences. It works on the principle of Brownian motion. The simplest diffusion equation uses Fick's law and has the form

$$\partial_t u + \nabla \cdot (-D\nabla u) = 0. \tag{2.16}$$

The unknown u represents the transported quantity (heat, mass,...), and D is the diffusion tensor. Inside the porous medium the diffusion within the water phase of the substance with concentration c would have the form

$$\partial_t (\phi S_w \rho_c c) + \nabla \cdot (-\rho_c \phi S_w D \nabla c) = 0.$$
(2.17)

The diffusion coefficient must take into account the type of substance (smaller molecules diffuse faster), the amount of fluid, and the properties of the porous medium. Simpler models might use constant diffusion tensor D, but D is often a nonlinear function of porosity and saturation in order to reflect the changes to available pathways.

Concrete is isotropic⁶ and the diffusion can be represented by the scalar function. This is not universal to all porous media. Many natural porous formations were created by sedimentation. They have a layered structure, and the diffusion speed is different along the layers and across them. Over time, the layers can be rotated and compressed in an arbitrary direction resulting in an anisotropic diffusion tensor.

The diffusion within the air phase follows the same formula as 2.17. If the diffusion tensor D is nonlinear, it can have a different form for the water and the air phase – as is the case for the relative permeability. We will specify diffusion tensors later in the model description, because they are less established and more problem dependent.

2.3.3 Dispersion

In the literature relevant to our topic two definitions of dispersion are prevalent. Sometimes dispersion is considered to be the sum of diffusion and mechanical mixing, and sometimes diffusion is not included. It is not uncommon to use the term

⁶Aside from more porous bits just below the reinforcement and on the surface [Ang+17; Kre84]. Both are very localized and can be neglected.

dispersivity, or even diffusion for the combined effect. We will use the term dispersion for the effect of mechanical mixing only, molecular diffusion is not included.

Dispersion is the mixing caused by variations in fluid velocities in the porous medium. Flowing from the point A to B, the fluid can take various paths. Paths have different lengths and are branching and connecting. The branching and connecting mixes the fluid and substances dissolved in it.

In a fully saturated porous medium, the dispersion depends on the geometry of the medium and the speed of the carrier fluid. Importantly, it is completely independent of the properties of the transported substance. At first glance, the dispersion equation is similar to the diffusion equation.

$$\partial_t (\phi S_w \rho_c c) + \nabla \cdot (-\rho_c D \nabla c) = 0$$
(2.18)

When the medium is fully saturated, the dispersion D has the form of the Scheidegger tensor. The medium is isotropic, but the direction of the flow plays a role [BC10]. The dispersion can be expressed in the vector form as

$$D = \left(\alpha_T \|\vec{v}\| I_n + \frac{(\alpha_L - \alpha_T)}{\|\vec{v}\|} \vec{v} \vec{v}_w^T\right)$$
(2.19)

or element-wise with separate formulas for diagonal entries and off-diagonal entries.

$$D_{ii} = \left(a_L v_i^2 + a_T \sum_{j \neq i} v_j^2\right) \frac{1}{|\vec{v}|}$$
(2.20)

$$D_{ij} = (a_L - a_T) v_i v_j \frac{1}{|\vec{v}|}, \qquad i \neq j$$
 (2.21)

Constants a_L , and a_T are longitudinal and transversal dispersion coefficients⁷. The longitudinal dispersion represents the strength of mixing in the flow direction and the transversal dispersion represents the strength of the mixing in the perpendicular direction(s). Longitudinal mixing is stronger then transversal, we use $\alpha_L = 8\alpha_T$. \vec{v} is the velocity vector of the carrier fluid and v_i are its entries.

Although the dispersion is the same for the wetting and the nonwetting phase when the medium is saturated, things change when the medium becomes unsaturated [CB02]. Dispersion inside the wetting phase increases as the saturation decreases. When its saturation is decreasing, the wetting phase leaves the biggest pores first. Main flow channels can not be used to their full capacity anymore and larger portion of the wetting phase flows through smaller pores. This leads to more

⁷If our medium was not isotropic, the strength of mixing could differ in various directions. Such a medium would have more dispersion coefficients [BC10].

tortuous paths, bigger differences in lengths of pathways, and thus more mixing [Mar+17].

How much the dispersion increases when the saturation is decreasing depends on the type of porous medium. Works we found kept the Scheidegger tensor and multiplied it by some factor. The relative strength of longitudinal and transversal dispersion is constant. The multiplicative factor is a negative power of saturation. Experiments in articles [Lat13; NMJ02] give estimates for the exponent on the saturation ranging from -2.16 for sand to -2.89 for glass beads. Weighting the dispersion by this factor makes the dispersion stronger at lower saturation assuming the velocity is kept constant. But at lower saturation, the velocity gets lower too.

Dispersion inside the non-wetting phase is even less researched than the dispersion in the wetting phase. Non-wetting phase leaves the smallest pores first, and the flow is concentrated into the biggest pores. This strengthens preferential pathways causing huge inconsistencies in experiment results. The overall effect is that the dispersion in the non-wetting phase stays constant, or decreases [CB02]. However, experimental results are often conflicting and the lower the saturation gets the less consistent results become. We found data covering air saturation at 70 - 100%, which is far from our models. We decided not to include dispersion in the transport of chemical agents in the air.

2.4 Concrete Composition

The concrete is a mixture of aggregate, cement, water, and optionally admixture. The aggregate is usually a mix of gravel and sand, ideally with particle size distribution that leaves minimum space in between. It forms the bulk of the concrete, around 70 - 80%. It is chemically inert, although the cement-binding qualities of its surface are of some importance during the pouring and hardening of the concrete.

The cement binds the aggregate together. Adding water starts the cement hydration process during which minerals precipitate and adhere to aggregate surfaces connecting them⁸. Besides enabling cement hardening, water has a secondary role. It improves the workability of the mixture before it hardens into concrete. More water translates into a better slurry flow but also weaker concrete with higher porosity.

⁸There are non-hydraulic cements that also need (aerial) carbon dioxide to harden. But those are less common, and the scope of this work is limited to the Portland cement anyway.

The amount of water is thus kept to the necessary minimum⁹. Water to cement ratio (by weight) usually ranges from 0.4 to 0.6 [Tay97].

After the concrete is poured it enters the hydration stage, which involves chemical reactions of cement constituents and hardening of the concrete. Overall, it is an exothermic process during which minerals forming the concrete crystallize. Two properties are of the greatest interest: early and late strength. As the name suggests, early strength is the (compressive) strength of the concrete shortly after pouring – mostly 24 hours but depending on the application this can range from a couple of hours to a week. Cement continues to harden, and its late strength is measured when it is 28 days old [Tay97].

This work studies the degradation of the concrete which comes after the hydration. The key feature distinguishing degradation and hydration processes is that the degradation is always caused by outside factors. Carbonation is caused by carbon dioxide entering the concrete. Leaching is caused by fresh water seeping into the concrete and leaving enriched in calcium. Freeze-thaw cycles are caused by changes in outside temperature, and so on.

2.4.1 Cement Chemistry

We will limit ourselves to the most common type of cement, Portland cement. Its composition is defined by European Standard EN 197-1 as

"Portland cement clinker is a hydraulic material which shall consist of at least two-thirds by mass of calcium silicates ($3CaO \cdot SiO_2$ and $2CaO \cdot SiO_2$), the remainder consisting of aluminum and iron-containing clinker phases and other compounds. The ratio by mass (CaO)/(SiO_2) shall be not less than 2,0. The content of magnesium oxide (MgO) shall not exceed 5,0% by mass".

This definition is quite broad, and Portland cement types are further divided into categories based on admixtures. Table 2.1 shows cement compositions from different experiments. All of them are ordinary Portland cement without additives (having the mark CEM I), although the first one belongs to a different subcategory since its late strength is lower.

As Table 2.1 suggests, in cement chemistry, it is common to denote the content in oxides and not in elements. For convenience, cement chemistry uses its own notation which is summarized in Table 2.2 [Tay97].

⁹Which entirely depends on the desired workability, the amount of water required for the complete cement hydration is lower. The risk is the formation of so-called honeycombs, places where you can see interlocked aggregate particles embalmed with a cement film but with voids in between. It happens when the ingredients are not mixed well or when the concrete layer is thin. Vibrating the concrete as it sets helps to prevent the formation of honeycombs.

Tab. 2.1: Composition of ordinary Portland cement without additives, CEM I. Nomenclature further includes its 28-day strength, usually 32.5, 42.5, or 52.5 MPa. Note that the composition of the same type of cement can vary. The column w/c has water-to-cement ratios used to form the concrete.

Туре	CaO	SiO ₂	Al_2O_3	Fe_2O_3	SO ₃	MgO	w/c	Source
CEM I 42.5	65.61	24.84	2.91	2.34	2.2	0.75	0.40	[JWG16]
CEM I 52.5	63.40	20.00	5.34	3.78	3.3	0.86	0.40	[JWG16]
CEM I 52.5	68.02	21.04	3.51	2.84	1.8	2.03	0.65	[CC06]
CEM I 52.5	64.02	19.81	5.19	2.38	3.5	-	0.43	[BS04]

Portland cement is named after calcium hydroxide CH (cement chemistry notation for Ca(OH)₂), which looks like the naturally occurring mineral Portlandite. CH forms big hexagonal plates that are usually situated in bigger pores.

The major part of the cement paste is composed of C-S-H phase. C-S-H is a collection of calcium silicate hydrates. In fresh cement paste its averaged formula is $1.7 \text{ C} \cdot \text{S} \cdot x\text{H}$ with $x \in [1, 4]$ and as the cement ages the ratio of calcium to silica decreases [Eks01]. C-S-H does not have any particular crystal structure, due to its lack of structure it is often called C-S-H gel. At the microscale it is formed by interlocking needles, the microstructure of the C-S-H resembles natural minerals Tobermorite and Jennite. C-S-H forms later in the curing process from bigger chunks of cement which were not properly mixed with water. It absorbs water from its vicinity and crystallizes, and often contains pieces of unhydrated cement in its center [Tay97]. C-S-H contributes to the strength of the concrete more than any other phase, but loses its strength when it degrades and the ratio of calcium to silica decreases.

notation.				
С	CaO			
S	SiO_2			
А	Al_2O_3			
F	Fe_2O_3			
Ī	SO_3			
Μ	MgO			
Т	TiO_2			
Κ	K_2O			
Ν	Na_2O			
Η	H_2O			
Ē	CO_2			
Р	P_2O_5			

Other minerals in the cement phase contain a significant amount

of other oxides from Table 2.2. Hydrates containing alumina and ferrite fall into the same category because those two oxides can easily substitute each other. Overgeneral formula of them is $C_3(A,F) \cdot xCaX \cdot yH$, where X is some anion that balances Ca^{+2} , for example SO_4^{-2} , $2OH^-$, or $2CO_3^-$. The term (A,F) stands for either Al₂O₃, or Fe₂O₃. Structure-wise there is little difference.

The two most common minerals in this category are AFt and AFm. AFt is an abbreviation for "alumina, ferric oxide, tri-sulfate". Its formula is $C_3(A,F) \cdot 3C\bar{S} \cdot 32H$. The closest natural mineral and the most common AFt component is ettringite, $C_3A \cdot 3C\bar{S} \cdot 32H$. M in AFm stands for monosulfate. AFm has general formula $C_3(A,F) \cdot C\bar{S} \cdot 12H$.

Older concrete contains calcium carbonate $C\bar{C}$ (CaCO₃). It comes from the carbonation reaction, where calcium in the pore solution reacts with carbon dioxide. The carbon dioxide is sourced from outside, so calcium carbonate appears when the concrete degrades. The deposition of calcium carbonate plugs pores, which strengthens the concrete, and reduces permeability and porosity. Moreover, it happens mostly in the main flow channels, which further reinforces the self-healing effect [SCK13].

Other minerals are less common and constitute a few percent of the total volume. They still have a significant impact on the cement quality and contain the oxides that made it into the cement chemistry notation but we did not mention yet¹⁰. Some minerals form only during the life of the concrete and only in certain environments, the most noteworthy example is Friedel's salt. This mineral contains chloride which is accumulated during the lifetime of the concrete. It is used to study chloride attack in environments rich in chloride like coastal areas, or bridges that are salted in winter [FB12].

Mineral Polymorphism

To make things more complicated, one molecular composition can lead to multiple crystal systems. Aside from the chemical conditions, the stability of the structure depends on temperature and pressure. Therefore, the type of the present polymorph is specific to the problem. For example, calcium car-

Polymorph	Crystal	cm ³ /mol
Calcite	Trigonal	36.93
Aragonite	Ortorhombic	37.32
Vaterite	Hexagonal	37.71

Tab. 2.3: Calcium carbonate polymorphs.

bonate, CaCO₃, has three polymorphs: calcite, aragonite, and vaterite.

Calcite is the most common. It is the only stable polymorph at ambient pressure and temperature and is greatly favored when calcium carbonate crystallizes. Aragonite is stable at high pressure, and is metastable at atmospheric pressure. It is only slightly less favored than calcite, and for all practical purposes is stable. The third polymorph, vaterite, is rare. Vaterite is stable at high temperatures, and at lower temperatures readily dissolves into calcite or aragonite [Laf+15; Chi+97; Hum+02]. These three polymorphs can be distinguished by their crystal systems, see Table 2.3. Our models do not consider crystal systems but use the molar volume. Usually, we deal with only one polymorph and in the case of multiple polymorphs, we use their averaged molar volume.

¹⁰We attempted to arrange them in decreasing order of importance and arrived at M, K, T, N, P.

Ion Substitutions

Ion substitution is, with some grain of salt, the opposite of polymorphism. It is about how various molecular compositions can create the same mineral. Typically, one atom is replaced by another with identical ionic charge without significantly affecting the molecular structure. The most common ion substitution pair is aluminum and iron. Both atoms have a charge of +3 and are of similar size. As mentioned in subsection 2.4.1, this substitution is common in AFt and AFm phases [Tay97].

Multiple substitutions are rarer, but not unheard of. The only condition is that replacing pair, triple, etc. have the same total ionic charge, and that the resulting mineral resembles the original. Minerals inside the concrete form from locally available atoms. Higher content of some atoms will lead to bigger structures resembling crystals that contain that atom. Less represented atoms get somehow incorporated too. Possibilities range from working as substitutes in other minerals to distorting the structure so much that it can no longer be considered a substitution.

2.4.2 Degradation of Concrete

In this work, we contemplate the two most prevalent degradation processes in the concrete: leaching and carbonation. Other, like the corrosion of the reinforcement, chloride intrusion, and damage caused by freeze-thaw cycles, are common too but not all structures are subject to them.

The leaching process has two parts. First, minerals forming the concrete are dissolved, and then their ions get carried away. The water in the concrete is highly alkaline as it is in chemical equilibrium with minerals forming the concrete. It is called a pore solution because it contains many ions, mostly calcium. Leaching happens when the alkalinity of the pore solution decreases. Minerals adjust to the new equilibrium – some dissolve, and others transform. Many minerals are hydrated products (i.e. their chemical composition contains a significant amount of water) so the dissolution releases ions and water. Release of ions slows down the alkalinity decrease [Eks01].

The alkalinity of the pore solution in fresh concrete is around 13 pH. The first to dissolve when pH is reduced is Portlandite. The next is the C-S-H phase which is progressively losing calcium and the ratio of calcium to silica decreases. This happens over a much larger interval of calcium concentrations and is accompanied by changes in the structure. The C-S-H phase undergoes polymerization, it loses its crystalline microstructure and becomes amorphous. In doing so its strength decreases.
Structures most susceptible to the leaching are dams and tunnels. Water flows through the concrete in one direction, so the alkaline pore solution is flushed out at one side and replenished by fresh water at the opposite side. Another cause of leaching can be a chemical reaction that reduces the alkalinity. Most commonly this reaction is carbonation.

The carbonation is the reaction of aqueous carbon dioxide with calcium hydroxide forming the calcite.

$$Ca(OH)_{2(s)} + H_2CO_{3(aq)} \rightarrow CaCO_{3(s)} + 2H_2O_{(l)}$$

The reaction takes place in water and calcite is deposited on pore walls. Carbon dioxide moves through the air much faster and in higher concentrations than in water, plus the high surface area between the water and the air phase allows carbon dioxide to rapidly transition from one phase to another. As a result, the carbonation reaction is fastest when both phases are present. Optimal relative humidity ranges from 50% to 70%, but for more porous concretes the optimum can be even higher [ŠL16].

The carbonation reaction is quick and can use all calcium in the pore solution. The carbonation depth is usually determined by the formaldehyde test. The carbonated region has a pH of around 9. The area where the carbonation takes place is fittingly called the carbonation front. It is a thin zone separating the region with calcium in pore solution and no carbon dioxide from the region with carbon dioxide and no calcium.

The effect of carbonation on concrete is mostly negative. It consumes calcium in the pore solution, reduces the pH, and, consequently, causes leaching. One positive aspect is the formation of the calcite. It adds a little to the strength of the concrete¹¹, and lowers porosity. The decrease in porosity will never balance the losses caused by leaching. However, the carbonation is faster in channels with higher mobility of reactants, so it seals areas where the concrete is the most vulnerable, like cracks. The carbonation is the main actor in the self-healing feature of the concrete [SCK13].

The most damaging part of the carbonation is the corrosion of the steel reinforcement. Initially, the reinforcement is passivated by the high alkalinity of the pore solution, but it can start to corrode when the pH decreases to around 10.5 [ŠL16]. The corrosion products have a bigger volume than the steel so the pressure builds up to the point when cracks form. The degradation is then accelerated because the carbon dioxide can get faster inside the concrete through the cracks.

¹¹Calcite is stronger than portlandite (CH), so carbonation is beneficial to young concrete. But when C-S-H starts dissolving the overall strength will only decrease.

If the corroding agent is reactive enough, the corrosion can happen even when the steel is passivated (to air). Usually, the corroding agent is chloride, and the corrosion has a different pattern to carbonation. While carbonation forms a front, chloride induces a pitting corrosion, which is localized. Chloride penetrates along preferential paths and corrodes the steel it reaches¹² [FB12].

The last chemical degradation we mention is the sulfate attack. The source of sulfates is typically contaminated soil. Sulfates cause pitting corrosion like chlorides but do not penetrate as deep. The sulfate attack poses a different danger. Sulfates react and form minerals of high volume (e.g. ettringite). Similar to corrosion, this builds up the pressure and can lead to the formation of cracks.

The final two degradation mechanisms are related to temperature. The first is the freeze-thaw cycle. When water in pores freezes it expands. The pressure buildup is weaker than in previous examples and the damage caused by the freeze-thaw cycles does not go very deep. Moreover, the damaged area is more resistant to the future cycles. Small pores are more vulnerable and once they are broken (and fused into bigger pores) higher pressure is necessary for further degradation¹³.

Heat-induced damage is studied too. Especially the ability of concrete to withstand fire [BH12]. The uneven thermal expansion builds up pressure and can cause cracking.

2.4.3 Simulating Chemical Reactions

Based on the speed of the reaction we distinguish two types of reactions: kinetic and equilibrium. Following [Krä08], let's describe the reactions on a simple example with three species A, B, and C and a two-way reaction

$$A + 2B \leftrightarrows C. \tag{2.22}$$

The speed of the kinetic reaction is derived from the chemical activity of the species. The forward reaction rate is

$$R_f = k_f a_A a_B^2 \tag{2.23}$$

and the backward reaction rate is

$$R_b = k_b a_C. \tag{2.24}$$

¹²So far, the best defense against chloride attack proved to be minerals that absorb chlorides.

¹³Modern technique to limit the effect of freeze-thaw cycles is to make a concrete with entrained air (not entrapped). The cement contains micro air bubbles that are not connected to the pore network. Empty spaces allow the concrete to be pushed aside by the freezing water without getting damaged.

 k_f and k_b are forward and backward reaction rate constants. a_* are chemical activities of the corresponding species. The chemical activity of a soluble species is usually proportional to its concentration, it is called the law of mass action.

The kinetic reaction is in chemical equilibrium when the forward and backward reaction rates are equal. If we use equilibrium reactions instead of kinetic the system is always in the chemical equilibrium. However, that does not mean that forward and backward reaction rates are equal. Quite the opposite. Overall, the reaction rate is as big as to maintain the equilibrium. There is no formula for the reaction rate, it is given implicitly. The reaction rate is a new unknown and the system gets a new equation

$$\frac{c_A c_B^2}{c_C} = k_e \tag{2.25}$$

called equilibrium law of mass action. The equilibrium constant k_e determines the ratio between the species of the forward and backward reaction.

Equilibrium reaction is favored when the reaction is fast, it helps to reduce the stiffness of the system. Increasing the kinetic reaction rate to infinity quickens the reaction and as a result concentrations of reactants are closer to the equilibrium. Equilibrium reaction enforces the chemical equilibrium but it is not infinitely fast, only as fast as to maintain the equilibrium.

The general system of chemical reactions is typically denoted with a stoichiometric matrix. Element s_{ij} of the stoichiometric matrix S is equal to the number of molecules of species i reacting in the reaction number j. (The stoichiometric matrix of the simple example is $S = (1, 2, -1)^T$.) The system of size I with J reactions can be written as

$$\partial_t(\phi S_w c_i) + L_i(c_i, \nabla c_i) - \phi S_w \sum_{j=1}^J s_{ij} R_j(c) = 0, \quad i = 1, \dots, I$$
 (2.26)

or in the vector form as

$$\partial_t(\phi S_w c) + L(c, \nabla c) - \phi S_w SR(c) = 0.$$
(2.27)

 ϕ and S_w are the porosity and the saturation. L_i is the mobility operator (e.g. advection plus diffusion). Reaction rate coefficients R_j can be calculated directly if the reaction j is kinetic,

$$R_j = k_{fj} \prod_{s_{ij} < 0} c_i^{|s_{ij}|} - k_{bj} \prod_{s_{ij} > 0} c_i^{s_{ij}}.$$
(2.28)

If the reaction j is an equilibrium reaction, then R_j is unknown, and the system has an additional equation of the form

$$\prod_{i=1}^{I} c_i^{s_{ij}} = k_{ej}.$$
(2.29)

Often the equilibrium equation 2.29 is written in the logarithm form

$$\sum_{i=1}^{I} s_{ij} \ln(c_i) - \ln(k_{ej}) = 0$$
(2.30)

if the positivity of concentrations can be guaranteed¹⁴.

Reactions of Minerals

Unlike soluble species, the chemical activity of a mineral is not proportional to its concentration. It is proportional to the surface area of the mineral in contact with the pore solution. The chemical activity of a mineral is usually deemed constant when its concentration is positive and is zero when the mineral is depleted.

Let's return to the simple example $A + 2B \rightleftharpoons C$ and let C be the mineral. The kinetic reaction rate is

$$R_{kin} = \begin{cases} k_f c_A c_B^2 - k_b, & \text{if } c_C > 0 \lor k_f c_A c_B^2 - k_b \ge 0\\ 0, & \text{if } c_C = 0 \land k_f c_A c_B^2 - k_b < 0. \end{cases}$$
(2.31)

As long as the mineral concentration is positive or the mineral formation is favored, the reaction rate is calculated as before. Just the chemical activity of the mineral is constant. When the mineral is depleted the dissolution is of course no longer allowed.

If we use chemical equilibrium, the reaction rate is still unknown but the equilibrium condition must be altered. It has the form of the complementarity constraint.

$$c_C \ge 0 \quad \wedge \quad k_e - c_A c_B^2 \ge 0 \quad \wedge \quad c_C (k_e - c_A c_B^2) = 0 \tag{2.32}$$

We require that the mineral concentration is nonnegative and that the solute is in chemical equilibrium or undersaturated. Undersaturated conditions are allowed only when the mineral is depleted, otherwise it would dissolve and replenish species A and B.

 $^{^{14}}$ The formula 2.29 and 2.30 struggle with species of zero concentration that do not participate in the reaction. In that case $0^0=1$ and $0\ln(0)=0.$

The general formula follows. Let c_i be concentrations of soluble species for i = 1, ..., I and mineral concentrations for i = I + 1, ..., I + M. Let J be the number of chemical reactions. Chemical reactions can affect minerals, but they never affect more than one mineral. Minerals are localized and can not interact with each other. Furthermore, let's assume each mineral participates in only one chemical reaction, the porous medium is fully saturated ($S_w = 1$), and the porosity is constant and unaffected by the reactions. The system then reads:

$$\partial_t c_i + L_i(c_i, \nabla c_i) - \sum_{j=1}^J s_{ij} R_j(c) = 0, \quad i = 1, \dots, I$$
 (2.33)

$$\partial_t c_i - \sum_{j=1}^J s_{ij} R_j(c) = 0, \quad i = I+1, \dots, I+M$$
 (2.34)

Reaction rates that do not involve minerals are the same as before. The formula of the kinetic reaction rate involving a mineral is

$$R_{j} = \begin{cases} k_{fj} \prod_{\substack{i=1\\s_{ij}<0}}^{I} c_{i}^{|s_{ij}|} - k_{bj} \prod_{\substack{i=1\\s_{ij}>0}}^{I} c_{i}^{s_{ij}}, & \text{if } c_{m_{j}} > 0 \lor \left(k_{fj} \prod_{\substack{i=1\\s_{ij}<0}}^{I} c_{i}^{|s_{ij}|} - k_{bj} \prod_{\substack{i=1\\s_{ij}>0}}^{I} c_{i}^{s_{ij}} \ge 0\right) \\ 0, & \text{if } c_{m_{j}} = 0 \land \left(k_{fj} \prod_{\substack{i=1\\s_{ij}<0}}^{I} c_{i}^{|s_{ij}|} - k_{bj} \prod_{\substack{i=1\\s_{ij}>0}}^{I} c_{i}^{s_{ij}} < 0\right) \end{cases}$$

$$(2.35)$$

Note that the reaction rate does not depend on the mineral concentration directly, it appears only in the branching condition. Products of chemical concentrations include only soluble species.

Equilibrium reaction rates are unknown and each adds additional condition to the system. The condition has the form of a complementarity constraint

$$c_{m_j} \ge 0 \quad \land \quad k_{ej} - \prod_{i=1}^{I} c_i^{s_{ij}} \ge 0 \quad \land \quad c_{m_j} \left(k_{ej} - \prod_{i=1}^{I} c_i^{s_{ij}} \right) = 0.$$
 (2.36)

When the medium is not fully saturated, mineral reaction rates become nonlinear. The exact formula is problem-specific and has to capture the dependence of the reactive surface area on the saturation. If the porosity is not constant the surface area relation will get even more complex.

Numerical Tools

99 All models are wrong, but some are useful.

— George Box

In this chapter, we cover all the steps taken to get from the system of equations to numerical results. First, we describe the finite volume scheme, where special attention is paid to averaging nonlinear terms in the flow calculation. Other discussed topics are iterative operator splitting, additive Schwarz domain decomposition that is used for parallelization, nonlinear solvers with added projection step, and the implementation of complementarity constraints which does not use inequalities. The last section summarizes the implementation in the numerical toolbox DUNE [Bas+21].

3.1 Finite Volume Discretization

The finite volume method is popular for its simplicity and ease of implementation. It is a way to go when the solution does not have high regularity, which is exactly our case. We deal with highly nonlinear problems, so the calculation of the flux between control volumes can be complicated. The focus of this section is the evaluation of the flux, in particular the combination of upwinding and averaging of terms.

In this work, we use two types of the finite volume method: cell-centered and vertex-centered. They are very similar, so we first explain the cell-centered scheme and then cover the differences of the vertex-centered scheme.

The finite volume method approximates the test space by piecewise constant functions. The functions are constant on each element of the mesh \mathcal{T} , and discontinuous between mesh elements. Let's denote this space as $V_{\mathcal{T}}$, and its basis functions as φ_{τ} .

$$\forall \tau \in \mathcal{T} \qquad \varphi_{\tau}(x) = \begin{cases} 1 & \text{for } x \in \tau, \\ 0 & \text{for } x \in \Omega \setminus \tau \end{cases}$$

In this work, the mesh is simple. It is 2D, rectangular, and axiparallel, although not necessarily equidistant.

We will explain the finite volume discretization on a mass balance equation for a single species

$$\partial_t u + \nabla \cdot Q + z = 0$$
 in Ω , (3.1)

where u is the mass of the species, Q is its mass flux, and z is the source or the sink term. The type of the species is irrelevant. For water in a porous medium, there would be $u = S_w \phi \rho_w$.

Let's look at the weak formulation of the problem. Multiplying the equation 3.1 with a test function $v \in V_T$ and integrating over the cell τ yields

$$\int_{\tau} \partial_t uv \, dx + \int_{\tau} \nabla \cdot Qv \, dx + \int_{\tau} zv \, dx = 0.$$
(3.2)

Naturally, the test function v is not time-dependent. We apply Green's formula on the middle term to move one derivation onto the test function.

$$\int_{\tau} \nabla \cdot Qv \, dx = -\int_{\tau} Q \cdot \overleftarrow{\nabla v} \, dx + \int_{\partial \tau} Q \cdot \vec{\nu} v \, dS \tag{3.3}$$

The test function is piecewise constant and its gradient is zero inside cells. Values on $\partial \tau$, the boundary of the cell, are obtained via the trace operator. Substituting 3.3 into 3.2 and summing over all cells results in

$$\sum_{\tau \in \mathcal{T}} \int_{\tau} \partial_t uv \, dx + \sum_{\tau \in \mathcal{T}} \int_{\partial \tau} Q \cdot \vec{\nu} v \, dS + \sum_{\tau \in \mathcal{T}} \int_{\tau} zv \, dx = 0.$$
(3.4)

We assume that the mesh is conforming so each cell's face is either on the boundary or shared with exactly one other cell. Right now we first sum over cells and then over cell's faces. Each face is thus visited twice. Those face integrals have different contributions and do not cancel out, because the test function v is discontinuous at cell boundaries. We want to rearrange the middle sum so that the sum visits each face only once but we have to introduce some notation first. For each face, we will use its normal vector to distinguish two neighboring cells. The cell in the direction of the normal vector will be called the outside cell and marked with the lower index plus. The other cell will be called the inside cell and indexed with minus. The notation is summarized in Figure 3.1.

The notation is extended to traces of functions. Trace from the inside cell is marked with index – and trace from the outside cell with +. In equation 3.4, all traces use cell τ , which according to the notation would be the inside cell because the outer normal vector points outwards. However, when we rearrange the face integrals and integrate over each face once, the outside cell's contribution is integrated in the opposite direction which introduces the multiplicative factor -1.

By the way, for this to work we have to assign a normal vector to each face. The direction of the normal vector does not matter, it can be either of the two. The normal vector of the boundary face will always point out of the domain.



Fig. 3.1: Given the face f and its normal vector $\vec{\nu}$, we distinguish inside and outside cell and mark them with index – and + respectively.

Let's denote the set of all cell faces as \mathcal{F} , boundary faces as \mathcal{F}_b , and inner faces as \mathcal{F}_i . Naturally, $\mathcal{F}_b \cup \mathcal{F}_i = \mathcal{F}$. The rearranged weak formulation is

$$\sum_{\tau \in \mathcal{T}} \int_{\tau} \partial_t uv \, dx + \sum_{f \in \mathcal{F}_b} \int_f Q \cdot \vec{\nu} v_- \, dS + \sum_{f \in \mathcal{F}_i} \int_f Q \cdot \vec{\nu} \left(v_- - v_+ \right) dS + \sum_{\tau \in \mathcal{T}} \int_{\tau} zv \, dx = 0.$$
(3.5)

The next step is the discretization of the primary variable and dependent functions. First, we use the method of lines to discretize the system in space and that system is then discretized in time¹. In each time step the numerical solution has the form $c_h(t_n) = \sum_{\tau \in \mathcal{T}} w_{\tau,n} \varphi_{\tau}$. We use a fully implicit scheme in time so at the time t_n the derivation in time is $\partial_t u_h|_{t_n} = \frac{u_h(t_n) - u_h(t_{n-1})}{t_n - t_{n-1}}$ and other terms use solely values from the time t_n . To get the finite volume system, the test function v is substituted for φ_{τ} for each $\tau \in \mathcal{T}$.

Integrals over cells τ in the equation 3.5 are discretized via the midpoint rule. It is the second order quadrature rule, so when h is the longest edge in the mesh \mathcal{T} the approximation is of $\mathcal{O}(h^2)$. The discretization of the flux Q is more complicated, depending on the nature of the flow different numerical fluxes will be used. We distinguish the flux induced by molecular diffusion, advection, and (mechanical) dispersion.

¹Discretizing in the reverse order, first in time and then in space, is called Rothe method. The Rothe method theory is richer as it allows changing spatial discretization between time steps. The method of lines is simpler, the discretized system is "just" a big system of ordinary differential equations. In this case, the primary variable of the discretized system has the form $c(t) = \sum_{\tau \in \mathcal{T}} w_{\tau}(t)\varphi_{\tau}$ with $w_{\tau}: (0, T_{\text{max}}) \to \mathbb{R}$ being the primary variables of the ODE system.

Vertex-centered scheme

In the vertex-centered scheme, the degrees of freedom are not situated in the middle of the mesh cells, but in the vertices. The test space with piecewise constant functions has basis functions that do not align with the mesh cells. Let \mathcal{P} be the set of all vertices of the mesh \mathcal{T} . Then for each vertex $P \in \mathcal{P}$ there is a basis function $\hat{\varphi}_P : \Omega \to \mathbb{R}$ defined as

$$\hat{\varphi}_P(x) = \begin{cases} 1 & \|x - P\| \le \|x - P'\| & \forall P' \in \mathcal{P} \\ 0 & \text{elsewhere,} \end{cases}$$
(3.6)

that is, the basis function $\hat{\varphi}_P$ is 1 in points that are closer to the vertex P than to any other vertex. The area where $\hat{\varphi}_P = 1$ is called control volume. On the rectangular mesh, the control volumes are rectangles too, but unlike in the cell-centered scheme they do not align with the mesh.

To derive the system of equations we follow the same steps we took for the cellcentered scheme. If we mark $\hat{\mathcal{T}}$ the set of control volumes and $\hat{\mathcal{F}}$ the set of control volumes' faces, we can use the same steps we did from 3.1 to 3.5 to arrive to the weak formulation

$$\sum_{\tau \in \hat{\mathcal{T}}} \int_{\tau} \partial_t uv \, dx + \sum_{f \in \hat{\mathcal{F}}_b} \int_f Q \cdot \vec{\nu} v_- \, dS + \sum_{f \in \hat{\mathcal{F}}_i} \int_f Q \cdot \vec{\nu} \left(v_- - v_+ \right) dS + \sum_{\tau \in \hat{\mathcal{T}}} \int_{\tau} zv \, dx = 0.$$
(3.7)

3.1.1 Two-point Flux

As the name suggests, two-point flux is the approximation of the flux that uses only values from two points. The points are centers of the control volumes between which the flux happens. Two points are insufficient to evaluate the full gradient, but we need to know only the flux in the normal direction. Marking centers of outside and inside control volumes as x_+ and x_- , the normal derivative is

$$\nabla c \cdot \vec{\nu} = \frac{c(x_+) - c(x_-)}{\|x_+ - x_-\|}.$$
(3.8)

The approximation requires the line connecting centers of control volumes to be perpendicular to the face $(\exists \alpha \in \mathbb{R} : \alpha(x_+ - x_-) = \vec{\nu})$. We use a rectangular mesh so this condition is fulfilled trivially. Now, to get the volumetric flux we just need to approximate the permeability.

Harmonic mean

Intrinsic permeability is evaluated as the weighted harmonic mean of control volumes' permeabilities. It is the analytical solution to the flow in the 1D case, which we derive following the proof in [Ipp14].

Suppose we have a 1D problem

$$-\partial_x \left(K(x)\partial_x p \right) = 0 \tag{3.9}$$

on the domain $p \in (0, l)$ with Dirichlet conditions $p(0) = p_0$ and $p(l) = p_l$. The permeability K(x) is positive and piecewise constant.

$$K(x) = \begin{cases} K_a & \text{for } x \in (0, a) \\ K_b & \text{for } x \in (a, l) \end{cases}$$
(3.10)

 $K_a, K_b > 0$ and $a \in (0, l)$. Let b = l - a.

This problem is well-posed and has a unique solution. Let's define the volumetric flux q as

$$q = -K(x)\partial_x p. \tag{3.11}$$

The equation 3.9 states that $\partial_x q = 0$. The flux q is constant although its value is not known yet. To calculate q we first isolate the $\partial_x p$ and then integrate.

$$q = -K(x)\partial_x p \quad \iff \quad -\frac{q}{K(x)} = \partial_x p$$
 (3.12)

The right side is straightforward

$$\int_0^l \partial_x p \, dx = p_l - p_0. \tag{3.13}$$

The permeability is piecewise constant, evaluating the left side is easy too.

$$\int_{0}^{l} -\frac{q}{K(x)} dx = -q \left(\int_{0}^{a} \frac{1}{K_{a}} dx + \int_{a}^{l} \frac{1}{K_{b}} dx \right) = -q \left(\frac{a}{K_{a}} + \frac{b}{K_{b}} \right)$$
(3.14)

$$q = -\frac{1}{\frac{a}{K_a} + \frac{b}{K_b}}(p_l - p_0) = -\frac{(a+b)K_aK_b}{bK_a + aK_b} \frac{p_l - p_0}{l}$$
(3.15)

To sum up, given endpoint pressures p_0 , p_l , and permeability K, the flux is equal to minus (directional) pressure gradient times the harmonic mean of the permeability. Even when the permeability is more complicated, it is still averaged by harmonic mean, possibly in the integral form $l/(\int_0^l K(x) dx)$.

In the finite volumes scheme, the endpoints are control volume centers and the position of the face determines the weights in the harmonic mean. The derivation of this formula relies on the fact that the permeability does not depend on the primary variable. In our models harmonic mean is used for intrinsic permeability and relative permeability is upwinded.

Upwind

Contrary to harmonic mean, upwind does not follow any analytical solution. It is a worse approximation than arithmetic mean², but it is necessary for numerical stability. When the equation has hyperbolic traits, using arithmetic mean leads to an unconditionally unstable scheme [SS18]. This time we will skip the derivation of the basic principles because it is much longer³.

Upwind means that the value is averaged with greater weight on the upstream cell – the one from which the flow originates. We use the so-called *full upwind* scheme where the value on the face is taken from the upwind cell and the downwind cell's value is downright disregarded. For the linear advection equation

$$\partial_t c + \vec{q}c = 0 \tag{3.16}$$

the flow between cells is approximated as

$$\int_{f \in \mathcal{F}_i} c \, \vec{q} \cdot \vec{\nu} \, dS = \begin{cases} c(x_-) \, \vec{q} \cdot \vec{\nu} \, |f| & \text{for } \vec{q} \cdot \vec{\nu} \ge 0\\ c(x_+) \, \vec{q} \cdot \vec{\nu} \, |f| & \text{for } \vec{q} \cdot \vec{\nu} < 0. \end{cases}$$
(3.17)

Numerically, upwind ensures that the matrix is an M-matrix. The flow direction determines which cell is upstream and using upwind results in putting the negative value on the off-diagonal and the positive on the diagonal⁴. Changing the formula to the upwind form is equivalent to adding diffusion in the upwind direction.

Some methods reduce the diffusion to a necessary minimum, but they are more costly and introduce spurious oscillations [JN11]. More complex methods can diminish these oscillations [Xu18; IH08]. Unsurprisingly, such methods work best in

²Upwind is the first order whereas arithmetic mean is the second order approximation in spatial coordinates.

³The proof of the instability of the simplest case (linear advection equation with forward Euler time stepping) shows that energy of the solution $E(t) = \int_{\Omega} u(x, t)^2 dx$ grows in time. Numerical solutions tend to get oscillatory and differences between neighboring cells grow in time. Extending these results to implicit time-stepping schemes is nontrivial.

⁴Arithmetic average would put half of the positive term on the diagonal and the other half on the off-diagonal. Half of the negative term would end up on other off-diagonal and half would go to the diagonal and cancel out the positive part. We end up with one positive and one negative term, both on off-diagonals.

situations for which they are designed⁵. In models on porous media, it is not common to limit the numerical diffusion. Flows have a diffusive nature and additional upwind-induced diffusion is not crucial. Meanwhile, it is common to neglect mechanical dispersion which adds mixing predominantly in the flow direction.

We also use upwind to evaluate the relative permeability coefficient. Averaging the relative permeability by harmonic mean is incorrect because it depends on the primary variable. Stability-wise, upwinding is similar to the situation in the advection equation. The pressure gradient (with intrinsic permeability) takes the function of the flux and the relative permeability takes the place of the advected substance.

$$\vec{q} \cdot \vec{\nu} = \begin{cases} -\frac{2K_{-}K_{+}}{K_{-}+K_{+}}\frac{k_{+}}{\mu_{+}}\left(\frac{p_{+}-p_{-}}{\|x_{+}-x_{-}\|} + \vec{g} \cdot \vec{\nu}\right) & \text{for } \frac{p_{+}-p_{-}}{\|x_{+}-x_{-}\|} + \vec{g} \cdot \vec{\nu} \ge 0\\ -\frac{2K_{-}K_{+}}{K_{-}+K_{+}}\frac{k_{-}}{\mu_{-}}\left(\frac{p_{+}-p_{-}}{\|x_{+}-x_{-}\|} + \vec{g} \cdot \vec{\nu}\right) & \text{for } \frac{p_{+}-p_{-}}{\|x_{+}-x_{-}\|} + \vec{g} \cdot \vec{\nu} < 0 \end{cases}$$
(3.18)

When the viscosity of the gas phase is not considered constant it is upwinded too.

3.1.2 Mechanical Dispersion

Remembering the section 2.3.3 and choosing the scaling factor for unsaturated conditions to be S_w^{-2} , formula for the dispersion is

$$S_{w}^{-2}D_{ws}\nabla c = \frac{1}{S_{w}^{2}\|\vec{v}_{w}\|} \begin{pmatrix} \alpha_{L}v_{1}^{2} + \alpha_{T}v_{2}^{2} & (\alpha_{L} - \alpha_{T})v_{1}v_{2} \\ (\alpha_{L} - \alpha_{T})v_{1}v_{2} & \alpha_{L}v_{2}^{2} + \alpha_{T}v_{1}^{2} \end{pmatrix} \begin{pmatrix} \partial_{x}c \\ \partial_{y}c \end{pmatrix},$$
(3.19)

where v_1 , v_2 are entries of the water velocity vector \vec{v}_w . On the axiparallel mesh, only one entry of the vector 3.19 is necessary to evaluate the flow between two neighboring control volumes. Still, knowing the full gradient of c and the water velocity vector is necessary.

In our setting, it is more natural to use the volumetric flux instead of the velocity. The transformation is simply

$$\vec{q}_w = \phi(S_w - S_w^r)\vec{v}_w, \tag{3.20}$$

⁵One common test has the concentration in three shapes: a cylinder, a cone, and a sinusoid. The flow field rotates these shapes around a common point and the distortion observed after one full rotation is used to compare numerical methods. Of interest are the preservation of sharp edges, the height of the cone, and avoiding the creation of spurious oscillations [JS08]. Full upwind smears edges and reduces the cone height, but does not add oscillations.



Fig. 3.2: Vertex-centered finite volumes on a rectangular mesh. Dashed lines separate control volumes, one is hatched. Solid lines divide mesh cells. Blue cell's vertices are numbered from 0 to 3 in a way DUNE orders local degrees of freedom.

i.e. the volumetric flux is the velocity scaled by the volume of the moving fluid. S_w^r is the residual saturation and for simplicity will be assumed to be zero⁶. The dispersion is then

$$S_{w}^{-2}D_{ws}\nabla c = \frac{1}{\phi S_{w}^{3} \|\vec{q_{w}}\|} \begin{pmatrix} \alpha_{L}q_{1}^{2} + \alpha_{T}q_{2}^{2} & (\alpha_{L} - \alpha_{T})q_{1}q_{2} \\ (\alpha_{L} - \alpha_{T})q_{1}q_{2} & \alpha_{L}q_{2}^{2} + \alpha_{T}q_{1}^{2} \end{pmatrix} \begin{pmatrix} \partial_{x}c \\ \partial_{y}c \end{pmatrix}.$$
 (3.21)

Those who prefer vector notation might be more familiar with

$$S_w^{-2} D_{ws} \nabla c = \frac{1}{\phi S_w^3} \left(\alpha_T \| \vec{q}_w \| I_n + \frac{(\alpha_L - \alpha_T)}{\| \vec{q}_w \|} \vec{q}_w^T \right) \nabla c.$$
(3.22)

Mechanical dispersion is used only with the vertex-centered finite volume scheme. On each cell, the dispersion is evaluated using degrees of freedom located in the vertices. We number the vertices from 0 to 3 like Figure 3.2 shows. The volumetric flux is calculated via the same formula as in the advection term, but incoming pressure, saturation, and porosity are averaged.

$$\vec{q}_w = - \begin{pmatrix} \frac{(K_0 + K_2)(K_1 + K_3)}{K_0 + K_1 + K_2 + K_3} \frac{\{k\}_u}{\mu} \begin{pmatrix} p_1 + p_3 - p_0 - p_2 \\ 2\|x_1 - x_0\| \end{pmatrix} \\ \frac{(K_0 + K_1)(K_2 + K_3)}{K_0 + K_1 + K_2 + K_3} \frac{\{k\}_u}{\mu} \begin{pmatrix} \frac{p_2 + p_3 - p_0 - p_1}{2\|x_2 - x_0\|} + \rho_w g \end{pmatrix} \end{pmatrix}$$
(3.23)

The symbol $\{\cdot\}_u$ in the formula 3.23 represents upwinding. The relative permeability is calculated using upstream average, which is either $k\left(\frac{S_w(x_0)+S_w(x_2)}{2}\right)$ or $k\left(\frac{S_w(x_1)+S_w(x_3)}{2}\right)$ in the first row of \vec{q}_w , and either $k\left(\frac{S_w(x_0)+S_w(x_1)}{2}\right)$ or $k\left(\frac{S_w(x_2)+S_w(x_3)}{2}\right)$ in the second row. The upstream direction is calculated from the pressure differential in the same row of \vec{q}_w . If the viscosity was not constant, it would be upwinded like the relative permeability.

The gradient of \boldsymbol{c} is handled the same way as pressure.

$$\begin{pmatrix} \partial_x c \\ \partial_y c \end{pmatrix} = \begin{pmatrix} \frac{c(x_3) + c(x_1) - c(x_2) - c(x_0)}{2\|x_1 - x_0\|} \\ \frac{c(x_3) + c(x_2) - c(x_1) - c(x_0)}{2\|x_2 - x_0\|} \end{pmatrix}$$
(3.24)

The porosity and saturation in the denominator of 3.21 and 3.22 are arithmetic means of all four values in the current mesh cell's vertices.

Mechanical dispersion uses one volumetric flux and gradient of c over the whole mesh cell. The mesh cell contains parts of four control volumes, the flux and gradient are reused to calculate the dispersion-induced flow between each pair of neighboring control volumes.

⁶Both our models have zero residual saturation for the water and the air phase. The scaling factor for unsaturated conditions is another place where nonzero residual saturation could make a difference. We believe that the scaling factor should account for residual saturation as well but we did not find a paper that would address this.

A Note on Alternative Dispersion Formula

The current scheme (equations 3.23 and 3.24) puts the same weight on all DoFs and evaluates the dispersion in the middle of the cell, indicated by the black point in Figure 3.3. In a simpler test case, we compared the midpoint scheme to two others that evaluate the dispersion in different points (and use a different gradient and velocity for each face). The first scheme evaluated the dispersion in the middle of the face between control volumes, i.e. in the point where the face intersects with the cell boundary – blue in Figure 3.3. The second scheme evaluated the dispersion in the middle of the face that is inside the cell – red in Figure 3.3.

The test case was Poisson problem $\nabla \cdot D\nabla u = 0$ on a rectangular domain $\Omega = \{(x, y) \in \mathbb{R}^2 : x \in (0, 1), y \in (0, 2)\}$ with nonhomogeneous diffusion $D = \begin{pmatrix} \sqrt{3} & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix}$. The boundary condition is Dirichlet with value $x^2 - 8\sqrt{3}xy + 2y^2$, which is also the problem's exact solution. All three schemes had a quadratic convergence rate⁷, with negligible differences between errors.

We also experimented with a scheme that reuses the volumetric flux from the advection term, so computing the extra fluxes for the dis-







Fig. 3.4: Three discretization schemes have the same convergence rate.

persion was avoided. Advection uses the two-point flux approximation, and four fluxes are available in each cell, one for each pair of neighboring control volumes. This choice is natural for the water volumetric flux in the direction of the normal vector (also used by the scheme marked by blue in Figure 3.3). For the water volumetric flux in the tangential direction to the normal vector, we used the flux calculated from two DoFs that were upstream relative to the flux in the normal direction. The saturation and porosity in the denominator of equation 3.19 were upwinded according to the flux in the normal direction too. The concentration gradient was treated similarly: Each partial derivation used two DoFs, the derivation

⁷The quadratic convergence is achieved for the points in vertices and, if we linearly interpolate the function between DoFs, on the whole domain too. Piecewise constant functions that have one value in each control volume are only linearly convergent.

in the normal direction used DoFs of control volumes touching the integrated face, and the derivation in the tangential direction used DoFs of control volumes lying upstream relative to the flux in the normal direction. Considering how little impact the dispersion has in the model in the chapter 4, it was surprising to see, that this alternative formula led to convergence problems. It remains unclear what was the particular problem with this scheme.

Remark on Efficiency

The fact that mesh cells do not match control volumes is concerning. Contributions to the residual and the Jacobian are spread over several cells and we have to repeat some work. Volume integrals use only DoFs of one control volume so their workload is quadrupled⁸. Face integrals fare better, the workload is doubled. Dispersion is the only term that has a different value for parts of one control volume in different mesh cells.

However, compared to the cell-centered scheme, the vertex-centered scheme allows better reuse of intermediate evaluations. Calculating the capillary pressure is done four times per control volume instead of five times (once per volume integral and once for each face integral). The permeability can be reused for horizontal and vertical flow so the number of evaluations of intrinsic permeability stays the same, although the number of evaluations of the relative permeability is higher⁹. Averaging and upwinding are done more times too but it is a trivial effort compared to the evaluation of nonlinear terms.

Iterating over the mesh gets simpler. The vertex-centered scheme iterates only over cells whereas cell-centered has to iterate over faces too, and there are more than twice as many faces than cells. Vertex-centered scheme loads more DoFs at once¹⁰, and (discounting any clever caching in either scheme) loads each DoF four times instead of five.

To sum up, the advantage is better reuse of loaded DoFs and simpler iteration over the mesh. The disadvantage is the repetition of some work. Among expensive nonlinear terms, capillary pressure is evaluated fewer times, but relative permeability more. Even the worst-case scenario of quadrupling the workload of the assembly might have an acceptable impact on the overall computational time if most of

⁸We assume the mesh is rectangular, and estimates are asymptotic, i.e., the workload changes on the boundary are not taken into account.

⁹Upwinding does not allow for good reuse. When the flow direction is consistent across the mesh cell, the vertex-centered scheme needs three evaluations, but doing two or four is possible. The cell-centered scheme needs one evaluation per face, which is equivalent to two per mesh cell.

¹⁰One disadvantage of finite volumes is its low arithmetic intensity, so loading more DoFs favors vertex-centered scheme. Loaded DoFs are used to calculate flows in two directions. Unfortunately, this work has to be repeated.

the work was spent on solving the system, not assembling the matrix and the residual. This is exactly our case, the model in chapter 4 spends less than 30% of time on the assembly and the percentage gets lower on bigger meshes¹¹.

3.1.3 Summary

Let's return to advection-diffusion-reaction (but not dispersion) equation 3.5 and its integral over the non-boundary face. We will mark the upwind, harmonic mean, and arithmetic mean by braces with lower index u, h, and a respectively, i.e., $\{\cdot\}_u$, $\{\cdot\}_h$, and $\{\cdot\}_a$. The integral is discretized as

$$\int_{f\in\mathcal{F}_i} (\vec{q}\rho c - \rho\phi SD\nabla c) \cdot \nu \, dS = \left(\vec{q}\cdot\vec{\nu}\{\rho c\}_u - \{\rho\}_u\{\phi SD\}_h \frac{c(x_+) - c(x_-)}{\|x_+ - x_-\|}\right) |f|.$$
(3.25)

In the advection term we upwind $\{\rho c\}_u$. When the density is constant, we get exactly the situation mentioned in subsection 3.1.1 that discusses the linear advection equation. When the density is not constant, for example, when gas density depends on the pressure, we upwind it too. Together, ρc represents the mass density or molar density of the component c, so we can not mix an upwinded c with averaged ρ .

The diffusion has no gravity part¹² and the gradient is evaluated via two-point approximation. The density is upwinded because from its point of view, the other terms look like the flux – it has the structure of the advection term. The harmonic average of the porosity is obvious. It changes slowly and has a similar position as the intrinsic permeability. The hardest to determine is the averaging of the saturation and the diffusion coefficient which is often nonlinear and depends on the saturation and porosity. The exact formula is open to discussion and often problem-dependent. We use harmonic mean because it gives the exact solution when porosity and saturation remain constant in time. Moreover, the diffusion acts within the phase so we have to prevent the diffusion from happening when the phase is not present, i.e. when the saturation is zero. The harmonic mean handles that naturally, we just need to continuously extend it to zero¹³. Beware that $\{\phi SD\}_h \neq \{\phi\}_h \{S\}_h \{D\}_h$.

 $^{^{11}}$ Sequential code on the coarsest mesh we used spends 29.2% of time on assembly. Refining the mesh reduces this percentage to 23.7% and then to 18.5%. Parallelizing the code further decreases it, but the effect is smaller and less consistent (roughly an additional 1.5 – 5%).

¹²Although it looks similar to the Darcy flow formula, effect of gravity on the concentration much weaker than on the pressure. Often the component c is an ion and the diffusion is accelerated by electrical charges which makes the gravity even weaker in comparison [Tok+17; GG91].

 $^{^{13}}$ We actually set it to zero also for small positive numbers to control rounding errors. Our regularization is quite weak, the mean is zero if either input is below 10^{-100} .

If the volumetric flux \vec{q} is not given, we have to calculate it on the spot. Our models use Darcy flux, and pressure and saturation variables are represented by finite volumes too. The directional flux is evaluated as

$$\vec{q} \cdot \vec{\nu} = -\{K(\phi)\}_h \left\{\frac{k(S)}{\mu}\right\}_u \left(\frac{p(x_+) - p(x_-)}{\|x_+ - x_-\|} + \{\rho_m\}_a \, \vec{g} \cdot \vec{\nu}\right).$$
(3.26)

Intrinsic permeability is evaluated by harmonic mean. It is not constant as the proof in the subsection 3.1.1 assumed, but the porosity change is slow.

Relative permeability is upwinded. The reasoning is twofold. From the view of saturation as the primary variable the term has hyperbolic properties of the advection. k(S) is the term evaluated at the face and the rest falls to the position of the flux. Another important reason, why the relative permeability does not use harmonic mean, is resolving the situation when saturation in one cell is zero. Relative permeability of water in such a cell is zero and the harmonic mean (more precisely its continuous extension) would be zero. This would stop any inflow to an empty cell keeping it empty forever.

Viscosity is usually constant, but in Chapter 4 we use gas mixtures where the change is significant. The viscosity then depends on the gas composition and is upwinded for the same reason as density.

The directional gradient of pressure is discretized by two-point method. No surprise here. The fluid density in the gravity term is evaluated by the arithmetic mean. Upwind is not necessary so a more accurate approximation is used. Note that the term with gravity has the mass density even if we use molar density elsewhere, otherwise units would not fit.

To conclude, the advection-diffusion equation in a porous medium is discretized as

$$\sum_{\tau \in \mathcal{T}} \phi S \rho c v |\tau| + \sum_{\tau \in \mathcal{T}} q v |\tau| - \sum_{f \in \mathcal{F}_b} \int_f (\vec{q} \rho c - \rho \phi S D \nabla c) \cdot \vec{v} v_- dS$$
$$- \sum_{f \in \mathcal{F}_i} \left(\vec{q} \cdot \vec{v} \{ \rho c \}_u |f| - \{ \rho \}_u \{ \phi S D \}_h \frac{c(x_+) - c(x_-)}{\|x_+ - x_-\|} \right) (v_- - v_+) |f| = 0 \quad (3.27)$$

The system has size $|\mathcal{T}|$ and is obtained by substituting the test function with basis functions $\varphi_{\tau}, \forall \tau \in \mathcal{T}$.

3.1.4 Boundary Conditions

The discretization of the boundary integral

$$-\sum_{f\in\mathcal{F}_b}\int_f \left(\vec{q}\rho c - \rho\phi SD\nabla c\right)\cdot\vec{\nu}v_-\,dS\tag{3.28}$$

depends on the type of boundary condition. The easiest is the Neumann condition that prescribes the overall flux b_f through the boundary.

$$\forall f \in \mathcal{F}_{bN} \quad -\int_{f} \left(\vec{q}\rho c - \rho\phi SD\nabla c \right) \cdot \vec{\nu}v_{-} \, dS = b_{f}v_{-}|f| \tag{3.29}$$

Dirichlet boundary condition depends on the type of the scheme. Vertex-centered finite volume has degrees of freedom on the boundary and we simply prescribe their value. Cell-centered finite volume scheme does not have any degrees of freedom on the boundary, thus we evaluate the boundary integral in a similar way we do for internal faces. Only instead of the neighboring cell's value we use the boundary value. Misusing the notation to make the trace of the outer cell mean the Dirichlet value, we arrive at

$$\forall f \in \mathcal{F}_{bD} \quad -\int_{f} \left(\vec{q}\rho c - \rho\phi SD\nabla c \right) \cdot \vec{\nu}v_{-} \, dS = \\ - \left(\vec{q} \cdot \vec{\nu} \{\rho c\}_{u} - \{\rho\}_{u} (\phi SD)_{-} \nabla c \cdot \vec{\nu} \right) v_{-} |f|. \quad (3.30)$$

Upwinded variables attain the value of the inside cell or the boundary condition. The harmonic mean is no longer used, we integrate only over one cell and use the value from the inside. This change affects the diffusion coefficient and the calculation of the flux \vec{q} . Directional gradient calculation now uses the inside of the cell and the Dirichlet value at the boundary face.

In Chapter 5 we also use another boundary condition type called *free outflow* condition, or *do nothing* condition. This condition prescribes only the pressure, which is then used to calculate the flux. If the pressure inside the cell is lower, the flux is zero. Otherwise, it is similar to Dirichlet condition 3.30, only the diffusion is omitted.

$$\forall f \in \mathcal{F}_{bF} \quad -\int_{f} \left(\vec{q}\rho c - \rho\phi SD\nabla c \right) \cdot \vec{\nu}v_{-} \, dS = -\vec{q} \cdot \vec{\nu}(\rho c)_{-}v_{-}|f| \tag{3.31}$$

$$\vec{q} \cdot \vec{\nu} = \begin{cases} -\left\{K_{\mu}^{\underline{k}}\right\}_{u} \left(\frac{p_{+}-p_{-}}{\|x_{+}-x_{-}\|} + \rho \, \vec{g} \cdot \vec{\nu}\right) & \text{for } p_{-} > p_{+} \\ 0 & \text{for } p_{-} \le p_{+} \end{cases}$$
(3.32)

Here, x_+ and p_+ are the location of the center of the face and prescribed pressure on the boundary.

It is noteworthy that even though the free outflow condition works numerically just fine, it is much harder to handle in the analysis. This boundary condition requires prescribing only the boundary pressure, other unknowns from the boundary do not appear in the formula. Fluxes are computed during the simulation, we do not know their values beforehand. The lack of prior knowledge about the unknowns' values or fluxes gives analysis problems.

3.1.5 Initial Conditions

Our system has only the first derivation in time. Prescribing the initial value for all variables is sufficient. Some regularization concerning the compatibility with the boundary condition might appear, but that is problem-dependent and will be discussed locally.

3.2 Operator Splitting

Operator splitting is a way to split the operator of the system of ordinary differential equations (ODEs) into one or more smaller parts. Since we use a method of lines to interpret the discretized system of PDEs as a system of ODEs, the operator splitting method is applicable to our problem too. For the linear problem

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = Ax(t) = Bx(t) + Cx(t), \quad t \in [0,T], \quad x(0) = x_0$$
(3.33)

with constant operator A = B + C, we know that the exact solution is

$$x(t) = e^{tA} x(x_0)$$
 (3.34)

with the matrix exponential defined as

$$e^{tA} = \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k = I + tA + \frac{t^2}{2!} A^2 + \frac{t^3}{3!} A^3 + \dots$$
(3.35)

Going from a time step t_n to $t_{n+1} = t_n + \Delta t$ and decomposing A into the sum of B + C, we arrive at the formula

$$x(t_{n+1}) = x(t_n) + \Delta t(B+C)x(t_n) + \frac{\Delta t^2}{2}(B^2 + BC + CB + C^2)x(t_n) + \mathcal{O}(\Delta t^3).$$
 (3.36)

Using the operator splitting means that the operator A is applied in two smaller steps B and C.

$$x(t_{n+1}) = e^{\Delta tC} e^{\Delta tB} x(t_n)$$
(3.37)

This leads to the solution

$$x(t_{n+1}) = x(t_n) + \Delta t(B+C)x(t_n) + \frac{\Delta t}{2}(B^2 + 2CB + C^2)x(t_n) + \mathcal{O}(\Delta t^3)$$
(3.38)

that might not be the same if B and C do not commute and $BC - CB \neq 0$. Some schemes apply operators in different ways to achieve a higher convergence rate (e.g. the second order Strang splitting $A = \frac{B}{2} + C + \frac{B}{2}$) or split A into more parts, but when the operators do not commute the scheme always produces errors.

Dealing with highly nonlinear systems complicates the situation even more. Our operators B and C do not commute and we do not have any good error estimate. Therefore, we use an iterative operator splitting method which works on the principle of the fixed point iteration. The method is summarized in the algorithm 1.

Algorithm 1 Iterative Lie-Trotter Operator Splitting with Backward Euler Time Stepping

Given: $u_{n-1}, c_{rel}, c_{abs}, n_{it}, i = 0.$
initialize $u_n^0 = u_{n-1}$
evaluate initial defect $d_0 = -Bu_n^0 - Cu_n^0$
while $d_i/d_0 > c_{rel}$ and $d_i > c_{abs}$
if $i = n_{it}$ stop, method failed to converge
solve for $u_n^{i+1/2}$: $\frac{u_n^{i+1/2} - u_{n-1}}{\Delta t} - Bu_n^{i+1/2} - Cu_n^i = 0$
solve for u_n^{i+1} : $\frac{u_n^{i+1}-u_{n-1}}{\Delta t} - Bu_n^{i+1/2} - Cu_n^{i+1} = 0$
evaluate defect $d_{i+1} = rac{u_n^{i+1} - u_{n-1}}{\Delta t} - Bu_n^{i+1} - Cu_n^{i+1}$
increment <i>i</i>
return the solution $u_n = u_n^i$

We stop iterating when the desired reduction c_{rel} of the defect is achieved or the defect is below the threshold c_{abs} . The stopping criteria can be met after the first half iteration, but this happens rarely in our model. If the operator splitting fails to converge within the given number of iterations, we try again with a smaller time step. If the time step becomes too small the computation is terminated.

When B and C are nonlinear they always use the newest u. After adding j as the index of the nonlinear solver iteration, the first solution step of Algorithm 1 would be divided into a sequence of linear systems

solve for
$$u_n^{i+1/2,j+1}$$
: $\frac{u_n^{i+1/2,j+1} - u_{n-1}}{\Delta t} - B(u_n^{i+1/2,j})u_n^{i+1/2,j+1} - C(u_n^{i+1/2,j})u_n^i = 0.$

In Chapter 4 we use operator splitting to split the system into two smaller parts. (It could also be treated as a relaxation method.) Operators B and C have empty rows, and no row is nonzero for both B and C.

3.3 Parallelization

The parallelization is achieved via overlapping Schwarz domain decomposition method [Bas18]. In short, the domain is split into several subdomains, and subdomain problems are solved in parallel. In this chapter, we explain what the subdomain problems look like, how solving them advances the solution of the original problem, and how it is implemented. We will refer to the original problem and its solution on the whole domain as *global* problem and solution.

The subdomain problem is similar to the original problem. It solves the same system of equations and has the same parameters and initial conditions. The only difference is (as the name suggests) the domain and, consequently, boundary conditions. The union of all subdomains is the domain and, since we use an overlapping method, the subdomains are not disjoint.

Implementation-wise, the domain is first split into disjoint subdomains and then the overlap is added. Figure 3.5 depicts the splitting of the domain into subdomains. The solid lines are the boundaries between the disjoint subdomains from before the overlap was added. The dashed lines are the boundaries of the subdomain Ω_4 and dotted of the subdomain Ω_5 with the overlap included. The width of the added overlap can be specified either in meters or in layers of cells. We will use rectangular axiparallel mesh and the overlap will be given in the number of layers (rows or columns)



Fig. 3.5: A domain split into nine subdomains in a 3×3 pattern. Subdomains Ω_4 , Ω_5 , and their overlap are highlighted.

of cells. Since all subdomains are widened the overlap region is twice as big as the added overlap.

On the original domain's boundary, the subdomain problem uses the same boundary condition. Elsewhere, the subdomain's boundary is called *inner* boundary and uses the Dirichlet condition with the value equal to the global solution at the corresponding place. Schwarz domain decomposition method is iterative and the inner boundary value comes from the previous iteration. Once all subdomain problems are solved, the global solution (and thus inner boundary values) is updated and the iteration cycle continues until the convergence criterion is fulfilled¹⁴.

Each subdomain problem computes a correction to the global solution. Having multiple subdomains overlapping at one place means that that place gets several corrections, one per subdomain. Sometimes, the corrections in the overlap region are weighted to give them the total weight of one (as is the case outside the overlap). Such a method is called the *restrictive* overlapping Schwarz method.

Each subdomain has its process rank and the communication between them is handled through message passing interface MPI [Gab+04]. Two types of communication are used. Pairwise communication passes information only between neighboring subdomains and is used for overlap values. Global communication passes information among all processes. It gathers data from all processes to one, which then scatters the processed data back. It is used to communicate the convergence criteria, e.g., to accumulate the norm of the defect or to find the maximum.

Besides the problem itself, the convergence of the Schwarz domain decomposition method depends on several parallelization properties [Bas18]. The convergence speed is affected by the width of the overlap region and the size of subdomains. The wider the overlap is, the better convergence. However, wider overlap also makes the subdomain problems bigger and thus more expensive to solve. Smaller subdomain problems are easier to solve, but we can not solve more than the number of available processing cores in parallel. Moreover, more subdomains bring a bigger communication overhead.

We do not want to delve into the theory, but we can not speak about the Schwarz method without mentioning the coarse grid correction. The coarse grid correction is necessary to keep the number of iterations level as the number of subdomains increases. It serves to communicate "long frequencies", i.e. effects that reach over distances comparable to the size of a subdomain. Coarse correction solves a system on the whole domain but with fewer degrees of freedom. Our solvers do not use coarse correction. The difficulty of our problems does not come from the system size but from local nonlinearity or degeneracy.

¹⁴This is called *additive* Schwarz method. Another common variant is the *multiplicative* method that updates boundary values each time a subdomain problem is solved. Unlike the additive, the multiplicative variant is not fully parallelizable due to these dependencies.

3.4 Newton and RASPEN Solver

In Chapter 4 we compare the performance of solvers called Newton and Raspen. In this section, we briefly describe both of them and highlight key differences from the parallelization point of view.

Original Newton-Raphson method predates modern computers and is sequential. Some steps can be fortunately parallelized. We use additive Schwarz domain decomposition to parallelize the solving of linear systems.

For the root finding problem $F(x) = (0, ..., 0)^T$, Newton iteration is

$$x_{j+1} = x_j - \lambda_j [\nabla F(x_j)]^{-1} F(x_j)$$
(3.39)

with *j* being the iteration number and λ_j a suitable damping parameter¹⁵. In our setting F(x) is the residual of the problem.

The algorithm is customizable. The main line is summarized in Algorithm 2 and the description of its steps follows.

Algorithm 2 Newton-Raphson Method

Given: $x_0, c_{rel}, c_{abs}, n_{it}, j = 0$ while not **converged assemble** matrix $[\nabla F(x_j)]$ **calculate** step direction $d = [\nabla F(x_j)]^{-1}F(x_j)$ use **line search** method to update $x_{j+1} = x_j + \lambda^j d$ j = j + 1

The algorithm's input is the initial guess x_0 . Other given variables are the algorithm's parameters: c_{rel} and c_{abs} are convergence criterion limits, and n_{it} is the maximum number of iterations the method is allowed to make.

The **convergence** criterion checks if the required reduction of the defect was reached or if the defect is small enough. That is, if

$$\frac{F(x_l)}{F(x_0)} < c_{rel} \quad \text{or} \quad F(x_l) < c_{abs}.$$
(3.40)

If the criterion is satisfied, the solution x_l is accepted. We use $c_{rel} = 10^{-8}$ and $c_{abs} = 10^{-13}$. If the algorithm does not converge in n_{it} iterations it stops and signals the failure to converge¹⁶.

The algorithm has an option of always performing at least one step, even when the initial defect is lower than c_{abs} . We found this option useful in combination with operator splitting. When one part changes at a much slower rate than the other, its

¹⁵The damping parameter can change between iterations.

¹⁶By throwing an exception in DUNE's case.

initial guess is often close to the solution. Forcing the algorithm to make at least one iteration prevents the stagnation of the solution.

The **assembly** of the matrix $\nabla F(x_j)$ and the calculation of the step direction are delegated to the linear solver, although the DUNE's Newton method has the option to skip the matrix assembly and reuse the older matrix if the ratio of the current and previous defect is below a threshold, i.e. if $\frac{F(x_j)}{F(x_{j-1})} < c_{asse}$. We use $c_{asse} = 0$ and always reassemble the matrix. In the case, that the linear solver is matrix-free, the assembly is obviously not performed.

To **calculate** the step direction $d = [\nabla F(x_j)]^{-1}F(x_j)$ the linear problem is solved. Again, the execution is left solely on the linear solver. DUNE's Newton solver can influence it by setting the parameter for minimal linear reduction. This makes the method to be an inexact Newton method if the parameter is weak (accepting less precise solutions). Early iterations can save some work and use less precise directions. The paper [HR89] estimates how the precision of the linear solver must be increased from iteration to iteration to achieve quadratic convergence when the Newton method uses a certain line search method.

Finally, we search for a new solution along the calculated direction. The DUNE offers two **line search** methods. The first is actually not a line search. It simply takes a step in that direction. No search is performed, it always uses $\lambda_j = 1$. The second option is the Hackbusch-Reusken line search method [HR89] that looks for a solution with a lower defect along the direction *d* by gradually decreasing the step length. (Shortening λ_j from 1 by the factor with default value 0.5.) It is more expensive than the previous option because it requires multiple evaluations of the defect but can lead to a smaller defect without performing the matrix assembly and solving the linear system.

For the model in Chapter 4 we implemented a new line search method with the extra projection step. We needed it to deal with negative chemical concentrations caused by aggressive reactions.

3.4.1 Projected Line Search

The projected line search is one part of the projected Newton method [Ber82]. The projected Newton method is designed for constrained problems where the optimum on the boundary might not be a local optimum. Compared to the normal Newton method it has three extra features: the projection step, generalized convergence criterion, and (optionally) a reduced Hessian.

The projection step keeps the solution inside the feasible region. The projection is performed inside the line search method when the new solution candidate is obtained, just before the residuum is calculated and the convergence is checked. Line search methods that try several solutions (like Hackbusch-Reusken line search [HR89]) use the projection on each one.

The generalized convergence criterion is for handling the situation when the (non-local) optimum is on the boundary. The solution is accepted even if the gradient is nonzero as long as the function decreases (in case a minimum is sought) outwards the feasible region.

The last feature of the projected Newton method, the reduced Hessian, is designed to improve the performance and does not affect the results. The reduced Hessian works only with simple constraints like $x_i \ge \pi$ where the constraint affects only a single unknown. This can be generalized to linear constraints by linearly



Fig. 3.6: The projection step.

transforming the system to coordinates in which the constraints are simple (see [Ber82] for details). When an unknown is on the boundary of the feasible region, it often stays on the boundary. The row in the Hessian corresponding to that variable is replaced by the row from the identity matrix. Further optimization is possible if the row assembly can be skipped. Note that the variable can still change – it depends on the residual. Most of the time, it moves outside and is projected back to the feasible region (i.e. to its previous value), so precision is not needed. Occasionally, it moves back inside (inaccurately), and the next iteration will have a proper row in the Hessian.

We use the projection in Chapter 4 to keep concentrations nonnegative. The projected line search method is used only as a means to improve convergence. We want to arrive to the same solution as the normal Newton method so we do not use the generalized convergence criterion. Indeed, if the solution was outside the feasible space (concentration was negative), our method would not converge.

Our use case needs only simple constraints. Degrees of freedom of the finite volume method directly store the values of the primary unknowns. To keep concentrations nonnegative, we take the vector of unknowns and set all negative concentrations to zero. The DUNE design allows us to skip the assembly of the rows corresponding to constrained unknowns on the boundary of the feasible region. Nevertheless, simplifying the Hessian has led to worse convergence, so we keep using the full Hessian.

3.4.2 RASPEN Solver

Raspen solver was implemented into DUNE by Chaiyod Kamthorncharoen as a part of his PhD project [Kam22]. RASPEN stands for restricted additive Schwarz preconditioned exact Newton's method. This method originated as a way to make Newton's method more robust by applying it to a preconditioned system. The preconditioning is nonlinear and involves solving nonlinear problems on subdomains.

The domain is split into *s* subdomains and overlaps are added. Then a fitting partition of unity is defined on the domain. The partition has *s* parts, each corresponding to a unique subdomain. That is, it is equal to 1 inside its subdomain but outside the overlap region, equal to 0 outside the subdomain, and something in between in the overlap region. Our variant is equal to $\frac{1}{m}$ where *m* is the number of subdomains overlapping at that point.

Subdomain problems use the Dirichlet boundary condition on parts of their boundary that are not the domain's boundary. The Dirichlet comes from the previous iteration's the global solution. Notably, the subdomain problems are nonlinear, which is a big difference to Newton's method. Subdomain problems often need a different number of iterations to converge. Solutions calculated by subdomain problems are multiplied by the partition of unity and added up. The value in the overlap region is thus a weighted average of values from several subdomains.

To be more precise, let's define the nonlinear algebraic problem arising from the discretization as $F : \mathbb{R}^n \to \mathbb{R}^n$, F(x) = 0, where *n* is the number of degrees of freedom (DoFs). Subdomain problem operators will be denoted as $G_k : \mathbb{R}^{n_k} \to \mathbb{R}^{n_k}$ with n_k being the number of DoFs of the *k*-th subdomain. Restriction operators $R_k : \mathbb{R}^n \to \mathbb{R}^{n_k}$ will serve to map the DoFs of the global solution to the DoFs of the subdomain problem solution. The matrix R_k is wide rectangular and has exactly one 1 per row, its other elements are zero¹⁷. The partition of unity operators $\hat{R}_k : \mathbb{R}^n \to \mathbb{R}^{n_k}$ are similar to the restriction matrices. They have the same pattern of nonzero elements, but have $\frac{1}{m}$ instead of 1 in the overlap region. The partition of unity property means that

$$\sum_{k=1}^{s} R_k^T \hat{R}_k = I = I^T = \sum_{k=1}^{s} \hat{R}_k^T R_k.$$
(3.41)

The addition of subdomain solutions can be expressed as

$$x_{j+1} = \sum_{k=1}^{s} \hat{R}_k^T G_k(R_k x_j)$$
(3.42)

 $^{^{17}}$ In practice, the restriction matrix is never constructed. The operation can be done without the matrix and more efficiently. The same holds for the matrix \hat{R}_k

and the preconditioned nonlinear system \mathcal{F} is then defined as

$$\mathcal{F}(x) = x - \sum_{k=1}^{s} \hat{R}_{k}^{T} G_{k}(R_{k}x) = 0.$$
(3.43)

 \mathcal{F} has the same solution as the original problem F. Now, we can solve this system by Newton's method

$$x_{j+1} = x_j - \lambda_j [\nabla \mathcal{F}(x_j)]^{-1} \mathcal{F}(x_j)$$
(3.44)

where λ_j is a damping parameter. This is the RASPEN method. It is impossible to construct the matrix $[\nabla \mathcal{F}(x_j)]^{-1}$, the system is solved in the matrix-free fashion.

Returning to the RASPEN abbreviation, the word Restricted is related to the partition of unity property, and the fact that the subdomain corrections are weighted before they are added up¹⁸. It is possible to omit the partition of unity, although a relaxation parameter must be used to help with convergence in the overlap region.

Additive Schwarz is a way of splitting the domain. Each subdomain problem uses the value of the previous iteration for its boundary conditions. As a result, subdomain problems are independent and can be solved in parallel.

Preconditioned (Newton) refers to redefining the problem into the form defined in the equation 3.43 and solving it by the Newton method. Finally, Exact Newton means that it is solved exactly like the Newton solver would solve it, albeit in a matrix-free fashion. Some inexact alternatives to RASPEN approximate the matrix $[\nabla \mathcal{F}(x^j)]^{-1}$ and thus solve a slightly different problem.

The projection step for the line search works with RASPEN too. It is placed into the nonlinear subdomain solver, we did not have to touch the global Newton iteration. Otherwise, the implementation is the same as before. The projection is applied right before the residuum is calculated, only this time it is the residuum of the subdomain problem.

3.5 Complementarity Constraints

Complementarity constraints, also called complementarity conditions, are better known in the field of optimization than PDEs. They name a type of constraint on two quantities that requires both to be nonnegative and at least one to be zero.

$$a, b \in \mathbb{R}: \quad a \ge 0, \quad b \ge 0, \quad ab = 0$$
 (3.45)

¹⁸Yes, our explanation (starting with equation 3.42) adds up subdomain solutions to form the global solution, not subdomain corrections to make the global correction. But weighing by the partition of unity makes them equivalent.

We encounter complementarity constraints in Chapter 5. The first use case is mineral equilibrium reactions. The chemical activity of minerals is constant (see section 2.4.3) up to the moment when the mineral is depleted and its chemical activity becomes zero. The two variables in complementarity constraints are the equilibrium law of mass action and the concentration of the mineral. Remembering the simple example of chemical reaction A + 2B = C with mineral C and soluble species Aand B, the complementarity constraint is

$$c_C \ge 0 \quad \land \quad k_e - c_A c_B^2 \ge 0 \quad \land \quad c_C (k_e - c_A c_B^2) = 0.$$
 (3.46)

 c_* are concentrations and k_e is the equilibrium constant.

The second use case is managing the phase disappearance. The two variables are the saturation of one phase and the condition that molar fractions of that phase sum to one. After denoting the molar fraction of *i*-th aqueous species as c_{wi} , the complementarity constraint for the water phase reads

$$S_w \ge 0 \quad \wedge \quad 1 - \sum_i c_{wi} \ge 0 \quad \wedge \quad S_w \left(1 - \sum_i c_{wi} \right) = 0. \tag{3.47}$$

When the phase disappears, the sum of molar fractions can be less than one¹⁹.

It is possible to implement complementarity constraints without inequalities. We need a function with the following properties:

$$f: \mathbb{R}^2 \to \mathbb{R}, \quad f(a,b) = 0 \iff a \ge 0 \land b \ge 0 \land ab = 0.$$
 (3.48)

We will use $f(a, b) = \min(a, b)$. Another popular choice is Fischer-Burmeister function $f(a, b) = a + b - \sqrt{a^2 + b^2}$. These functions are equivalent to complementarity constraints and using them does not require inequalities. This allows us to use solvers that work only with equations. The price is the nonsmoothness, neither function is differentiable at (0, 0).

As a result, we have to use methods suitable for semismooth problems. Luckily, the differences between Newton's and semismooth Newton's method are significant only in the theory. The algorithm is the same, but weaker assumptions on the problem regularity lead to slower convergence rate estimates. Semismooth Newton has a local Q-superlinear convergence rate, although further assumptions on regularity can improve this to Q-quadratic [Hin10].

¹⁹The sign of the second inequality is implementation dependent. When the water phase disappears, its molar fractions become meaningless. It is the degeneracy we can not avoid, especially if some molar fractions are primary unknowns of the system. We algebraically tie molar fractions to other quantities (mostly to molar fractions of the second phase) in order to give them a unique value when the water phase disappears. That determines the sign of the second inequality.

The semismooth Newton has to meet one condition that is satisfied trivially in the smooth case. We have to guarantee that the matrix $\nabla \mathcal{F}(x^j)$ exists and is invertible, i.e. we have to assign it a suitable value at points where \mathcal{F} is not differentiable. It is sufficient when the \mathcal{F} is semismooth [Ulb00; HIK02].

To get to the definition of the semismooth function, we first define the B-subdifferential, the set of all existing limits of gradients at that point.

Definition 3.5.1 The Boulingard (B-)subdifferential is the set

$$\partial_B f(x) = \{ G \in \mathbb{R}^{n \times m} : \exists \{x_k\} \subset \Omega \quad \text{with} \quad x_k \to x, \, \nabla f(x_k) \to G \}.$$
(3.49)

The convex hull of this set is called Clarke's generalized Jacobian, and if m = 1 (f(x) is a scalar) it can also be called the generalized gradient. For example, the B-subdifferential for the minimum function is $\partial_B \min(x, y)|_{(x,y)=(0,0)} = \{(1,0), (0,1)\}$, and the generalized gradient is $\{(x, y) \in \mathbb{R}^2 : x \ge 0, y \ge 0, x + y = 1\}$.

Definition 3.5.2 The function f is semismooth at x when it is locally Lipschitz continuous, directionally differentiable at x, and

$$\max_{M \in \partial f(x+h)} \|f(x+h) - f(x) - Mh\| = \mathcal{O}(\|h\|) \quad \text{as} \quad h \to 0.$$
(3.50)

If f is semismooth at all $x \in \Omega$, we call f semismooth (on Ω).

This definition resembles a normal Fréchet derivative, but the set $\partial f(x + h)$ is Clarke's generalized Jacobian. The set of points where a semismooth function is not differentiable has Lebesgue measure zero. This is a result of Rademacher's theorem and follows from the assumption that the function is locally Lipschitz continuous. Semismoothness is a property closed under scalar multiplication, summation, and composition [Hin10].

The minimum function and Fischer-Burmeister function are semismooth. They are even semismooth of order 1, i.e. the right-hand side of the equation 3.50 is $\mathcal{O}(||h||^2)$. The convergence rate of the semismooth Newton is then Q-quadratic.

Finally, we remark that using the minimum function makes the semismooth Newton method equivalent to the primal-dual active set strategy. Choosing the rows of the Jacobian according to the branch of the minimum function is equivalent to picking the condition of the active set [HIK02].

3.6 Implementation in DUNE

In this section, we discuss the implementation of our numerical scheme. We use the software package *Distributed and Unified Numerical Environment*, shortly DUNE [Bas+21; Bas+08a; Bas+08b; BB07; BB08; San20]. It is a C++ modular library for solving partial differential equations. Our work uses the discretization module PDELab that builds on other basic/fundamental modules. Besides dune-pdelab eight other modules are utilized: dune-common, dune-istl, dune-localfunctions, dune-geometry, dune-grid, dune-typetree, dune-functions, and dune-raspen.

The core module dune-common contains the build system and the infrastructure and foundation classes for other modules. It is required by all other modules.

The abbreviation in dune-istl stands for Iterative Solver Template Library. It offers matrix and vector classes, and a collection of solvers that operate on them. The module is built for matrices arising from finite element discretization and can use their block structure for optimization. The type of the finite element is then supplied at the compile time as the template parameter.

The module dune-geometry implements reference elements. A reference element is the simplest individual from a category of finite element shapes. For example, the unit cube $(0,1)^n \subset \mathbb{R}^n$ is the reference element for cuboids, and for $n \in$ $\{2,3\}$ the set $\{x \in \mathbb{R}^n | x_i \ge 0, \sum_i x_i \le 1\}$ is the reference element for triangles or tetrahedrons. The module also contains the mappings within the reference element (e.g. the mapping from a cube to any of its faces), and quadrature rules.

The core module dune-grid sets the interface for grids and also implements some simple grids like YASP grid²⁰. Dune-grid requires dune-geometry and uses it to give mesh cells attributes of finite elements, e.g., where and how many degrees of freedom the cell has.

The dune-localfunctions module is closely connected to dune-geometry and builds on reference elements. It implements shape functions, interpolation operators, etc. Since we use the finite volume method, these things are trivial. We do not use dune-localfunctions, but it is required by dune-pdelab.

The dune-functions module works with the concept of a callable function and provides an abstraction to make it easy to evaluate the object that stems from finite elements. It wraps the vector of degrees of freedom together with the grid and the type of finite element space. Similarly to dune-localfunctions, using the finite volume scheme greatly simplifies this task and we do not directly use the dunefunctions module.

The module dune-typetree manages trees of the types. It originated as an abstraction of function spaces, but the generic implementation is useful on other tree structures too. It is a template library and types are required to be known at compile time, which enables it to use various optimizations through template metaprogramming. We use it to load model parameters and set up the settings of

²⁰YASP grid stands for Yet Another Structured Parallel grid, and it is the grid we use in our models.

solvers. Dune-typetree was developed as a part of the dune-pdelab module, but after it caught the attention of other projects it was separated into its own module.

PDELab is a dicretization module. It requires all previous modules. PDELab is our main toolbox and provides grid operators that iterate over the grid, assemblers that collect contributions from reference elements to the residual and matrix, nonlinear solvers, and many other tools. After correctly setting PDELab classes, all we have to do is write equations in the localized form, i.e. what operations are executed on the reference element. Plus write boundary conditions (localized form now uses reference elements corresponding to faces of mesh cells) and the initial condition.

Dune-raspen is a small module developed by Chaiyod Kamthorncharoen as part of his dissertation [Kam22]. It contains the nonlinear solver RASPEN (see Section 3.4).

Honorable mentions go to modules dune-uggrid, dune-vtk, and dune-pdelabtutorials, which were not directly used in this work. The grid module dune-uggrid implements an unstructured grid which would be used if the problems we solve had more complicated geometry and the structured grid would not suffice.

The extension module dune-vtk provides file readers and writers for the VTK XML file formats that are native to the application Paraview that is used to visualize results. We used the functionality provided by the PDELab module, but it is planned for PDELab to stop supporting these features and leave them to the dune-vtk.

The last honorable mention goes to the dune-pdelab-tutorials. As the name suggests, the module is a teaching material that helps with learning to use the PDELab. Tutorials cover a range of problems with increasing difficulty and include nice documentation and even some exercises with solutions. Besides being a teaching material tutorials serve as a checkup of PDELab. Unit tests are nice and quick, but solving problems that were not designed to catch specific bugs adds another layer of ensuring correctness. Tutorials do not cover the whole functionality of PDELab. Some features (like operator splitting) are described in PDELab-recipes – minimal but well-commented code snippets, each focusing on one feature.

Our program follows the structure of tutorial examples. The code is split into five parts. The first part is the *.cc* file that is the target for the compilation. It lists included files (along with files from other modules) and has the function *main* where it prepares the mesh and calls the function named *driver* located in the file with the same name. The driver defines all the types and is responsible for the flow of the program. Most of the work is done here, whether it is defining the finite element space, time-stepping scheme, solvers, or looping over time steps.

The third part is the *local operator* that implements problem equations. The word local means that at any time it has access to only one or two mesh cells and their degrees of freedom. It is intended to operate on the reference element with its basis

functions and transform the result to the mesh cell. Since we use the finite volume method, basis functions and quadratures are trivial. We operate directly on the mesh element and skip the transformation step.

The local operator has nine different functions that accumulate local contributions to the residual or jacobian of the problem. They are combinations of *alpha*, *jacobian*, or *lambda* with *volume*, *skeleton*, or *boundary*. Alpha and lambda methods accumulate to the residual. The difference is that lambda does not have access to the solution vector. Therefore, lambda functions can be skipped during numerical differentiation, those terms would cancel out anyway. Jacobian methods accumulate to the Jacobian matrix. DUNE offers classes that calculate the Jacobian via numerical differentiation from values provided by the corresponding alpha method.

Volume, skeleton, and boundary methods differ by which element they operate on. Volume is designed to evaluate the contributions from integrals over cells, skeleton evaluates integrals over faces between two cells, and boundary evaluates integrals over boundary faces. The grid operator (implemented in PDELab) is tasked with iterating over the grid and accumulating the local contributions from the above methods to correct entries of the residual vector and the Jacobian matrix. For timedependent problems, it is recommended to separate the spatial and the temporal local operator. It enables to automate the creation of the time stepping scheme, which can then be chosen by picking the right time-stepping operator.

The last two files define the problem details. The first provides functions and constants specific to the problem. This includes boundary conditions, initial conditions, functions like the water retention curve, and any constants required by the local operator. The second file is loaded at runtime and by custom has type *.ini*. It contains only constants, especially those that are changed often (e.g. the name of the program output), because changing other files requires recompilation to take effect.

First Model: Leaching and Accelerated Carbonation of Concrete

4

The first model represents a laboratory experiment of accelerated carbonation of concrete. A small sample of concrete is exposed to air with 50% carbon dioxide content for 175 days. This setup was inspired by [BS04] and we use the same functions and constants where applicable, most importantly source functions describing the effects of leaching. The key differences between our and their model are the addition of the air phase and the effects of mechanical dispersion in the water phase. We use a two-dimensional domain, consider gravity, and use a different numerical scheme.

The main result of this chapter is of numerical character. We use this model to study the behavior of two nonlinear solvers, in particular, how they are impacted by the projected line search (introduced in Subsection 3.4.1).

4.1 Governing Equations

$$\partial_{t} (\phi S_{w} \rho_{w}) + \nabla \cdot \left[-\rho_{w} \underbrace{\frac{K_{w} k_{w}}{\mu_{w}} (\nabla p_{w} - \vec{g} \rho_{w})}_{q_{a}} \right] = F_{l} L_{w}$$

$$\partial_{t} (\phi S_{a} \rho_{a}) + \nabla \cdot \left[-\rho_{a} \underbrace{\frac{K_{a} k_{a}}{\mu_{a}} (\nabla p_{a} - \vec{g} \rho_{a})}_{\vec{q}_{a}} \right] = -F_{c} M_{\text{CO}_{2}}$$

$$\partial_{t} (\phi S_{w} c_{w}) + \nabla \cdot \left[c_{w} \vec{q}_{w} - D_{w} \nabla c_{w} \right] = F_{l} L_{\text{Ca}} - F_{c}$$

$$\partial_{t} (\phi S_{a} p_{a} c_{a}) + \nabla \cdot \left[p_{a} c_{a} \vec{q}_{a} - D_{a} \frac{M_{a}}{M_{\text{CO}_{2}}} p_{a} \nabla c_{a} \right] = -F_{c} RT$$

$$\partial_{t} \phi = F_{l} L_{\phi} - F_{c} V_{\text{CaCO}_{3}}. \quad (4.1)$$

The model captures three phenomena: the flow of water and air in a porous medium, transport of calcium in the water phase, transport of carbon dioxide in the air phase, and chemical reactions leaching and carbonation which also affect the porosity. The system has five equations. The first four are mass balance equations for water, air, calcium, and carbon dioxide, and the last equation tracks the porosity change.

Primary unknowns highlighted in red are water pressure, air phase saturation, calcium molar concentration in water, carbon dioxide molar fraction in the air phase, and porosity. The initial and boundary conditions influenced our choice of water pressure and air saturation as primary variables. Water saturation is high throughout the simulation. These primary variables remain well-defined if the air phase disappears.

Names of all terms are summarized in Table 4.1. Values of constants are listed in Table 4.3. Bottom index w is used for water and species therein, and index a for air. We use * as an index in places where either w or a apply (and also c in case of the pressure). When * appears multiple times in a formula, all its occurrences represent the same symbol.

The following subsections provide a more detailed explanation. To improve legibility, we split the explanation of the system into three parts: two-phase flow of water and air, transport of chemical agents, and chemical reactions.

Variable	Short description
ϕ	porosity
p_*	pressure (water, air, capillary)
S_*	saturation
ρ_*	density
c_w	calcium molar concentration
c_a	O_2 molar fraction in air
$\vec{q_*}$	volumetric flux
K_*	intrinsic permeability
<i>k</i> *	relative permeability
μ_*	viscosity
\vec{g}	gravity
D_w	dispersivity of calcium in water
D_a	diffusion coef. of CO_2 in air
M_a	air molar mass
$M_{\rm CO_2}$	carbon dioxide molar mass
F_l	leaching rate
F_c	carbonation rate
L_w	mass of leached water
L _{Ca}	nr. of moles of leached calcium
L_{ϕ}	volume freed by leaching
R	ideal gas constant
Т	temperature (constant)
V _{CaCO3}	calcite molar volume

4.1.1 Two-Phase Flow in Porous Medium

We start with mass balance equations for water and air.

$$\partial_t \left(\phi S_w \rho_w \right) + \nabla \cdot \left(\rho_w \vec{q}_w \right) = F_l L_w \tag{4.2}$$

$$\partial_t \left(\phi S_a \rho_a \right) + \nabla \cdot \left(\rho_a \vec{q}_a \right) = -F_c M_{\text{CO}_2} \tag{4.3}$$
Pores are filled with water and air, there is no third liquid phase. We assume residual saturations are zero, therefore

$$S_w + S_a = 1.$$
 (4.4)

Water density is constant, but air density depends on the air pressure and carbon dioxide concentration. The formula for the density is mentioned in Subsection 4.1.2 that discusses the transport of carbon dioxide.

The terms on the right-hand side of the equations correspond to leaching and carbonation. They will be covered in Subsection 4.1.3. We note that the presence of calcium has a negligible effect on water phase properties¹.

The flux is described via Darcy's law.

$$\vec{q}_w = -\frac{K_w k_w}{\mu_w} \left(\nabla p_w - \vec{g} \rho_w\right) \tag{4.5}$$

$$\vec{q}_a = -\frac{K_a k_a}{\mu_a} \left(\nabla p_a - \vec{g} \rho_a\right) \tag{4.6}$$

The connection between water and air pressure is defined via van Genuchten's formula for capillary pressure [Gen80].

$$p_c = p_a - p_w = \frac{1}{\alpha} \left(S_a^{-\frac{1}{m}} - 1 \right)^{1/n}$$
(4.7)

This formula normally uses an effective saturation, but in this model, they have the same value as the residual saturation is zero for both the water and the air phase. Values of constants α , n, and m can be found in Table 4.3.

Relative permeability follows the van Genuchten-Mualem model. Like the previous formula, this law would use effective saturation if residual saturation was not zero.

$$k_w = \sqrt{S_w} \left(1 - \left(1 - S_w^{\frac{1}{m}} \right)^m \right)^2 \tag{4.8}$$

$$k_a = \sqrt[3]{S_a} \left(1 - (1 - S_a)^{\frac{1}{m}} \right)^{2m}$$
(4.9)

Intrinsic permeability follows Kozeny-Carman law.

$$K_* = K_{*0} \frac{\phi_0^3}{\phi_0^3} \frac{(1-\phi_0)^2}{(1-\phi)^2}$$
(4.10)

¹Calcium hydroxide solubility is low, about 1.72 g per liter at $20^{\circ}C$. Or 23.35 mol/m³ in units of our primary variable c_w . Other calcium ions are present at much lower quantities than calcium hydroxide.

The initial porosity ϕ_0 is constant and can be found in Table 4.3 together with water and air intrinsic permeability coefficients K_{w0} and K_{a0} . Having different water and air intrinsic permeability is specific to cement-based materials that have atypically lower water permeability (see Section 2.2).

4.1.2 Transport of Chemical Agents

Mass balance equations for calcium and carbon dioxide are

$$\partial_t (\phi S_w c_w) + \nabla \cdot [c_w \vec{q}_w - D_w \nabla c_w] = F_l L_{\mathsf{Ca}} - F_c \qquad (4.11)$$

$$\partial_t (\phi S_a p_a c_a) + \nabla \cdot \left[p_a c_a \vec{q}_a - D_a \frac{M_a}{M_{\mathsf{CO}_2}} p_a \nabla c_a \right] = -F_c RT. \qquad (4.12)$$

The calcium mass balance equation is in moles, the primary variable c_w is calcium concentration and has unit mol/m³. Multiplying the equation 4.11 by calcium hydroxide molar weight, $M_{Ca(OH)_2}$, would turn it into mass preserving form. The equation 4.12 was also transformed. Its mass-preserving form was divided by the factor $\frac{M_{CO_2}}{RT}$. It comes from the ideal gas law that is used to describe the relation between the pressure and density.

$$\rho_{c_a} = \frac{M_{\rm CO_2}}{RT} p_a c_a \tag{4.13}$$

The primary variable c_a represents the mole fraction of the carbon dioxide in the air. Following Dalton's law, the product p_ac_a represents the carbon dioxide partial pressure.

The calcium dispersivity represents the combined effects of diffusion and dispersion.

$$D_w = D_{wf} + S_w^{-2} D_{ws} (4.14)$$

The diffusion is the same as in [BS04].

$$D_{wf} = F_l \, e^{9.95\phi} \, 2.3 \cdot 10^{-13} \, \mathrm{m/s^2} \tag{4.15}$$

Coincidentally, the dependence of the diffusion on the saturation follows the same rule as the leaching rate coefficient F_l , which will be defined in the equation 4.26.

The dispersion has the form of Scheidegger tensor [BC10].

$$D_{ws} = \left(a_T |\vec{v}_w| I_n + \frac{(\alpha_L - \alpha_T)}{|\vec{v}_w|} \vec{v}_w \vec{v}_w^T\right)$$
(4.16)

Since the dispersion gets stronger with decreasing saturation (see Section 2.3.3) we weight it by negative power of the saturation. Longitudinal and transversal

dispersion coefficients can be found in the table 4.3. Water velocity can be calculated from the flux as

$$\vec{v}_w = \frac{1}{\phi S_w} \vec{q}_w. \tag{4.17}$$

We can thus express the dispersion tensor via flux as

$$D_{ws} = \frac{1}{\phi S_w} \left(\alpha_T |\vec{q}_w| I_n + \frac{(\alpha_L - \alpha_T)}{|\vec{q}_w|} \vec{q}_w^T \right).$$
(4.18)

The carbon dioxide diffusion in the air phase has the form

$$D_a = \phi^{\frac{4}{3}} S_a^{\frac{10}{3}} 8 \cdot 10^{-7} \,\mathrm{m/s^2}. \tag{4.19}$$

The dispersion in the air phase is not considered (Section 2.3.3 explains why).

The air phase is treated as a mixture of two gases: carbon dioxide, and atmospheric air without no carbon dioxide. The concentration of carbon dioxide affects the molar weight and the viscosity of the air phase. The molar weight of the air phase is a weighted average of the molar weight of carbon dioxide M_{CO_2} and the molar weight of air without carbon dioxide $M_{\bar{a}}$ that has other gasses of the same proportions as in the atmosphere.

$$M_a = (1 - c_a)M_{\bar{a}} + c_a M_{\rm CO_2} \tag{4.20}$$

The air density is calculated via ideal gas law.

$$\rho_a = \frac{M_a}{RT} p_a \tag{4.21}$$

Air phase viscosity is determined by Wilke's equation

$$\mu_{a} = \frac{(1-c_{a})\mu_{\bar{a}}}{(1-c_{a})+c_{a}\varphi_{\bar{a}\bar{c}}} + \frac{c_{a}\mu_{\bar{c}}}{c_{a}+(1-c_{a})\varphi_{\bar{c}\bar{a}}}, \quad \varphi_{xy} = \frac{\left(1+\left(\frac{\mu_{x}}{\mu_{y}}\right)^{\frac{1}{2}}\left(\frac{M_{y}}{M_{x}}\right)^{\frac{1}{4}}\right)^{2}}{2\sqrt{2}\left(1+\frac{M_{x}}{M_{y}}\right)^{\frac{1}{2}}}, \quad (4.22)$$

where $\mu_{\bar{a}}$ and $\mu_{\bar{c}}$ are viscosity of air and viscosity of carbon dioxide. The pressure dependence of the viscosity is neglected because it is low [SZF05].

4.1.3 Chemical Reactions

The last equation of our system tracks the porosity change.

$$\partial_t \phi = F_l L_\phi - F_c V_{\text{CaCO}_3} \tag{4.23}$$

 L_{ϕ} describes the volume of the leached minerals, and V_{CaCO_3} is the molar volume of the calcite.

The carbonation rate coefficient is

$$F_c = 0.492 \,\phi \, S_w^{\frac{3}{2}} (1 - S_w^{\frac{3}{2}}) \, c_w \, p_a c_a. \tag{4.24}$$

It represents the number of moles of reactants participating in the carbonation. The coefficient has several parts. The reaction is kinetic and chemical activities are proportional to concentrations. The calcium concentration appears directly in the equation and the carbon dioxide concentration in water is proportional to its partial pressure in air.

Compared to the air, water can hold relatively little carbon dioxide and its mobility is lower too. The amount of reacting carbon dioxide is thus dominated by the supply from the air phase. On the other hand, calcium is present only in the water phase. Carbonation is the most efficient in the range of 60 - 80% water saturation [ŠL16] and the part $4S_w^{\frac{3}{2}}(1 - S_w^{\frac{3}{2}})$ accounts for this.

Finally, ϕS_w represent the fraction of the space where carbonation takes place. The constant 0.123 appearing in the formula 4.24 is purely empirical [BS04].

$$F_{c} = \overbrace{0.123c_{w} p_{a}c_{a}}^{\text{reaction speed}} \overbrace{\phi S_{w}}^{\text{volume fraction}} \overbrace{4S_{w}^{\frac{3}{2}}(1-S_{w}^{\frac{3}{2}})}^{\text{carbonation efficiency}}$$
(4.25)

The leaching rate coefficient is the Bažant-Najjar formula [BN72]

$$F_l = \frac{1}{1 + (5S_a)^4}.$$
(4.26)

It describes the fraction of the minerals that are in contact with water and can dissolve. This model is based on a study [Ade92] that dealt with situations in which the water saturation is decreasing².

Functions L_w , L_{Ca} , and L_{ϕ} represent the mass of leached water, number of moles of leached calcium, and volume of dissolved minerals. They take calcium concentra-

²If saturation rises after leaching, formula 4.26 introduces a hysteresis effect – out of two otherwise identical control volumes, the one with lower water saturation during leaching will experience less mineral dissolution. Evening out the saturation would not equalize the amount of leached minerals. It goes against the assumption that minerals are in equilibrium with pore solution. Nevertheless, we find this approximation less disruptive than neglecting the hysteresis in the water retention curve.



Fig. 4.1: Leaching functions determining the amount of leached water, calcium, and volume of leached minerals. \bar{M}_w , \bar{M}_{Ca} , and \bar{V}_{ϕ} appear in equations under the time derivative, because the effect of leaching depends on the change of the calcium concentration.

tion as an input and appear in equations under the time derivative. They capture the dependency of the leaching effects on the change of the calcium concentration.

$$L_w = M_{\text{H}_2\text{O}} \,\partial_t \bar{M}_w(c_w) = M_{\text{H}_2\text{O}} \left(\partial_{c_w} \bar{M}_w(c_w) \right) \left(\partial_t c_w \right) \tag{4.27}$$

$$L_{\rm Ca} = \partial_t \bar{M}_{\rm Ca} \tag{4.28}$$

$$L_{\phi} = \partial_t \bar{V}_{\phi} \tag{4.29}$$

Functions \overline{M}_w , \overline{M}_{Ca} , and \overline{V}_{ϕ} are plotted in Figure 4.1, and their derivatives (by c_w) are listed in Table 4.2. Functions are piecewise linear with six stages based on the calcium concentration. Stages come from idealized cement paste composition that consists of four minerals³ [BS04]. Minerals are leached at different ranges of calcium concentrations, so each stage has a different group of chemically active minerals. C-S-H goes through several stages and its Ca/Si ratio is progressively lowered.

Tab. 4.2: Leached minerals at given calcium concentration. Follows the model of [BS04].The number range in C-S-H (...) indicates its Ca/Si ratio.

c_w range	$\partial_{c_w} \bar{M}_w$	$\partial_{c_w} \bar{M}_{Ca}$	$\partial_{c_w} ar{V}_\phi$	Leached minerals
(21, 22)	1100	1300	$36.25\cdot10^{-3}$	Portlandite
(14.8, 21)	50	125	$7.5 \cdot 10^{-3}$	Aft, Afm, C-S-H (1.45–1.65)
(9.3, 14.8)	425	100	$0.5 \cdot 10^{-3}$	Aft, C-S-H (0.9–1.45)
(1.6, 9.3)	2.5	2.5	$2.5 \cdot 10^{-3}$	C-S-H (0.85–0.9)
(0.4, 1.6)	1825	750	$15.75 \cdot 10^{-3}$	C-S-H (0.85)
(0, 0.4)	0	0	0	no dissolution

4.2 Model Setting

We model a laboratory experiment where a small sample of concrete is exposed to a high concentration of aerial carbon dioxide. Initial and boundary conditions follow [BS04]. Water pressure is calculated from saturation and air pressure using equation 4.7 for capillary pressure.

The sample has a cubic shape with a side of length 5 cm. Left and right side are in contact with air and carbon dioxide. The front and the back faces are insulated so the problem has a symmetry that can be rendered with 2D mesh. The bottom is insulated too, and the top can be either insulated or exposed to air.

The domain is rectangular and represents a vertical cross-section left-to-right through the center of the sample. The problem has a mirror symmetry left-to-right

³The sample is made of concrete and 3/4 of its mass is inert aggregate. Leached functions look better for the plain cement paste but we prioritized the clarity over round numbers.

	Value	Unit	Short description
ρ_w	998.205	kg/m ³	Water density
$M_{Ca(OH)_2}$	56.0774	g/mol	$Ca(OH)_2$ molar weight
$M_{\rm CO_2}$	44.01	g/mol	O_2 molar weight
$M_{\bar{a}}$	28.9566	g/mol	Air molar weight
K_{w0}	$1.0038 \cdot 10^{-21}$	m^2	Water intrinsic permeability ^a
K _{a0}	$50K_w$	m^2	Air phase intrinsic permeability
μ_w	$1.002 \cdot 10^{-3}$	Pa∙s	Water dynamic viscosity
$\mu_{ar{c}}$	$1.48 \cdot 10^{-5}$	Pa∙s	CO ₂ viscosity
$\mu_{ar{a}}$	$1.8369 \cdot 10^{-5}$	Pa∙s	Air viscosity
\vec{g}	$(0, -9.81)^T$	m/s ²	Gravity
α	$2 \cdot 10^{-8}$	Pa^{-1}	Flow constant
n	1.65	—	Flow constant
m	$1 - 1/n \approx 0.39$	_	Flow constant
R	8.3144598	J∕(K·mol)	Ideal gas constant
T	293.15	K	Temperature
α_L	1		Longitudinal dispersion
α_T	$\alpha_L/8$		Transversal dispersion
V_{CaCO_3}	$36.93 \cdot 10^{-6}$	m ³ /mol	Calcite molar volume
V_{Ca}	$31.7 \cdot 10^{-6}$	m ³ /mol	Calcium hydroxide molar volume

Tab. 4.3: Values of constants. The air phase is treated as a mixture of carbon dioxide and atmospheric air without carbon dioxide.

^{*a*}The article [BS04] we used as a baseline for the model parameters lists $\frac{K_{w0}\rho_w}{\mu_w} = 10^{-15}$ m/s, but units do not add up. Those terms directly appear in the model equations so we assume that the number is correct and the error is in units: missing m/s² coming from the gravity factor in hydraulic conductivity which would then be $9.81 \cdot 10^{-15}$ m/s. Hydraulic properties are in agreement with measurements in [SMJ12].

S_{a0}	= 0.051
p_{a0}	= 101325 Pa
c_{w0}	$= 22 \text{ mol/m}^3$
c_{a0}	= 0
ϕ_0	= 0.094

Tab. 4.4: Initial conditions. Tab. 4.5: Dirichlet boundary conditions.

S_{aD}	= 0.128
	= 101325 Pa
	-0 mol/m^3
	=0 mol/m
c_{aD}	= 0.5

and we use the mesh of the size 2.5×5 cm with the homogeneous Neumann boundary on the right side. The left side of the domain is in contact with air and has the Dirichlet boundary condition for all unknowns except porosity (which does not need a boundary condition). Boundary values are listed in Table 4.5. Water pressure is calculated from air pressure and saturation using the capillary pressure formula 4.7. The boundary condition for calcium plays a role only briefly in the beginning of the simulation. Carbonation quickly consumes the calcium so homogeneous Dirichlet, Neumann, or free outflow conditions soon have the same effect. We settled on the Dirichlet condition because it has the weakest initial shock.

The bottom boundary condition is homogeneous Neumann. The sample lies on a plate with the bottom side insulated. The top boundary is either insulated and uses homogeneous Neumann condition, or, if it is exposed to air, uses the same Dirichlet condition as the left side.

The initial condition is an equilibrium. There is no carbon dioxide, calcium has a constant concentration, and initial pressure and saturation result in no flow across the sample (with the exception of the jump at the Dirichlet boundary). Table 4.4 lists the initial conditions on the bottom of the sample and to account for the effect of the gravity, the air and water pressure decrease with the elevation so that

$$(\nabla p_w - \vec{g}\rho_w) = 0$$
 and $(\nabla p_a - \vec{g}\rho_a) = 0$, at $t = 0$



Fig. 4.2: Initial and boundary conditions.

Pressure and saturation of the Dirichlet boundary condition are also listed for the bottom of the sample and change with the elevation⁴. The water pressure decreases linearly, but the air density depends on pressure so the pressure drop has an exponential nature although the difference is small on

⁴The motivation behind the adjusting for the elevation is the setting with the insulated top side of the sample. The gravity is the only effect that differentiates the setting from a 1D problem and we wanted to remove the effect of the unstable initial condition on the shape of the carbonation front.

a 5 cm sample. Water pressure decreases faster than air pressure so the air saturation on the top of the sample is slightly higher than on the bottom.

We present three test cases that differ in the type of the top boundary and gravity. The first, called 1D, neglects gravity and the top boundary is insulated. That gives it symmetry and all degrees of freedom in one mesh column hold the same value. We will nevertheless use the same mesh as for other settings, just to be able to compare computational effort. The second scenario considers gravity and the top boundary is insulated too. The initial condition thus has the symmetry like 1D case, but gravity will skew the carbonation front. The third test case considers gravity and the top boundary is in contact with air. It uses the Dirichlet condition with the same value the left boundary has at this height.

4.3 Discretization

We are using the vertex-centered finite volume scheme with backward Euler time stepping. Let $u = (p_w, S_a, c_w, c_a, \phi)$ be the trial function, and $w = (w_1, w_2, w_3, w_4, w_5)$ the test function. Using the notation from Section 3.1.3 the residual formulation of the discretized system reads

$$\begin{aligned} r(u,w) &= \sum_{T\in\mathcal{T}} \partial_t \left(\phi S_w \rho_w\right) w_1 |T| &- \sum_{T\in\mathcal{T}} F_l L_w w_1 |T| \\ &+ \sum_{F\in\mathcal{F}_i} \left(-\rho_w \frac{\{K_w\}_h \{k_w\}_u}{\mu_w} \left(\nabla p_w - \vec{g} \rho_w \right) \cdot \vec{v}_F \right) [w_{1^-} - w_{1^+}] |F| \\ &+ \sum_{T\in\mathcal{T}} \partial_t \left(\phi S_a \rho_a \right) w_2 |T| &- \sum_{T\in\mathcal{T}} -F_c M_{\text{CO}_2} w_2 |T| \\ &+ \sum_{F\in\mathcal{F}_i} \left(-\{\rho_a\}_u \frac{\{K_a\}_h \{k_a\}_u}{\{\mu_a\}_u} \left(\nabla p_a - \vec{g}\{\rho_a\}_a \right) \cdot \vec{v}_F \right) [w_{2^-} - w_{2^+}] |F| \\ &+ \sum_{T\in\mathcal{T}} \partial_t \left(\phi S_w c_w \right) w_3 |T| &- \sum_{T\in\mathcal{T}} \left(F_l L_{\text{Ca}} - F_c \right) w_3 |T| \\ &+ \sum_{F\in\mathcal{F}_i} \left(\{c_w\}_u \vec{q}_w \cdot \vec{v}_F - \left(\{D_{wf}\}_h + \{S_w^{-2}\}_u D_{ws} \right) \nabla c_w \cdot \vec{v}_F \right) [w_{3^-} - w_{3^+}] |F| \\ &+ \sum_{T\in\mathcal{T}} \partial_t \left(\phi S_a p_a c_a \right) w_4 |T| &- \sum_{T\in\mathcal{T}} -F_c R T w_4 |T| \\ &+ \sum_{F\in\mathcal{F}_i} \left(\{p_a c_a\}_u \vec{q}_a \cdot \vec{v}_F - \{D_a\}_h \frac{\{M_a\}_u}{M_{\text{CO}_2}} \{p_a\}_u \nabla c_a \cdot \vec{v}_F \right) [w_{4^-} - w_{4^+}] |F| \\ &+ \sum_{T\in\mathcal{T}} \partial_t \phi w_5 |T| &- \sum_{T\in\mathcal{T}} (F_l L_\phi - F_c V_{\text{CaCO}_3}) w_5 |T|. \end{aligned}$$

$$\tag{4.30}$$

The system is already long enough, so we decided not to express all terms fully. Since we use backward Euler time stepping, partial derivation for time step at t_n

is $\partial_t f = \frac{f(t_n) - f(t_{n-1})}{\Delta t_n}$ and all spatial terms are evaluated at the time t_n . The time derivative in the leaching function L_w is evaluated as

$$L_w = M_{\rm H_2O} \, \frac{\bar{M}_w \left(c_w(t_n) \right) - \bar{M}_w \left(c_w(t_{n-1}) \right)}{\Delta t}.$$
(4.31)

 L_{Ca} and L_{ϕ} follow the same pattern. The product of the gradient and the unit outer normal vector is the directional derivative $\nabla f \cdot \vec{\nu} = \frac{f(x_+) - f(x_-)}{\|x_+ - x_-\|}$.

The dispersion is more complicated, the full explanation is in Section 3.1.2. D_{ws} is defined in equation 4.18 and discretized as

$$D_{ws} = \frac{1}{\{\phi S_w\}_a} \left(\alpha_T |\vec{q}_w| I_n + \frac{(\alpha_L - \alpha_T)}{|\vec{q}_w|} \vec{q}_w \vec{q}_w^T \right).$$
(4.32)

The flux is different from the advection flux, it is constant throughout the mesh cell. It uses the same formula but input coefficients are averaged.

4.4 Numerical Setup

4.4.1 Operator-splitting

We use operator-splitting to split the system into two smaller parts called the flow part and the contaminant part. The flow part has two equations: mass balance equations of water and air. The primary variables are water pressure and air saturation. The contaminant part has three equations: mass balance equations of calcium and carbon dioxide and the equation of porosity change. The primary variables are calcium molar concentration, carbon dioxide concentration, and porosity.

$$Flow part \begin{cases} \partial_t \left(\phi S_w \rho_w\right) + \nabla \cdot \left[-\rho_w \underbrace{\frac{-\vec{q_w}}{\mu_w} \left(\nabla p_w - \vec{g} \rho_w\right)}{\mu_w} \right] = F_l L_w \\ \partial_t \left(\phi \underbrace{S_a}{\rho_a}\right) + \nabla \cdot \left[-\rho_a \underbrace{\frac{K_a k_a}{\mu_a} \left(\nabla p_a - \vec{g} \rho_a\right)}{-\vec{q_a}} \right] = -F_c M_{\text{CO}_2} \end{cases}$$

$$Contaminant part \begin{cases} \partial_t \left(\phi S_w c_w\right) + \nabla \cdot \left[c_w \vec{q_w} - D_w \nabla c_w\right] = F_l L_{\text{Ca}} - F_c \\ \partial_t \left(\phi S_a p_a c_a\right) + \nabla \cdot \left[p_a c_a \vec{q_a} - D_a \frac{M_a}{M_{\text{CO}_2}} p_a \nabla c_a\right] = -F_c RT \\ \partial_t \phi = F_l L_\phi - F_c V_{\text{CaCO}_3}. \end{cases}$$

We use an iterative operator-splitting method (see Section 3.2) because parts of the split operator do not commute and we do not have a posteriori estimate of the splitting error. The iteration converges when defects of both the flow and the contaminant part are reduced by 10^6 or lower than 10^{-10} . Typically, the system converges after the contaminant part is solved and the defect of the flow part remains below 10^{-10} . The maximum number of operator-splitting iterations is 10. If the operator-splitting does not converge in 10 iterations, we reduce the time step size and start anew.

To solve each part of the operator-splitting problem we require the defect to be reduced by 10^8 or to become smaller than 10^{-12} . If the partial solution has negative calcium or carbon dioxide concentration, we stop iterating and consider the operator-splitting failed to converge. Calcium concentrations above -10^{-6} and carbon dioxide concentration above -10^{-4} are tolerated. (The problem with negative solutions was eventually resolved by implementing the projection step into the nonlinear solver's line search method.)

4.4.2 Time Stepping

The optimal time step size is the one that makes the best progress in the simulation relative to the computational effort. The ideal size is not known, we use heuristics to adjust it during the simulation. The time step size is increased by 50% if all applications of the nonlinear solver (Newton or Raspen) in the current time step converged in less than 4 iterations and if at most 4 operator-splitting iterations were used. It is decreased by 25% if the nonlinear solver or operator-splitting fails to converge. If the time step size decreases below 10^{-4} s, the program stops.

These criteria were chosen by running the simulation on a coarse mesh and optimizing for the computational time. This calibration used Newton solver although the same criteria are used for Raspen solver⁵.

4.4.3 Solver Parameters

Parameters are the same for Newton and Raspen solver (see Section 3.4). The convergence criterion is to achieve the relative reduction of 10^{-8} or get the norm of the defect below 10^{-13} . The maximum number of iterations is 20. Solvers use the maximum norm.

Additionally, the flow part's solver is forced to perform at least one iteration even if the norm of the initial defect is below the threshold 10^{-13} . It is to prevent stagnation

⁵The rules for time step management (the number of iterations and the increase/decrease) had a noticeable impact on the final computation time. Other parameters like the maximum number of operator-splitting, nonlinear solver, or linear solver iterations had negligible effect unless, of course, they got prohibitively low. We picked round numbers close to (hardly recognizable and possibly setting-dependent) optimum.

when the time step is too small. Sometimes this happens when the chemical part has problems converging and needs small a time step. The flow part is then barely changing and its initial defect is low. Using projected line search leads to longer time steps, which makes this setting unnecessary.

4.5 Results

In this section, we first describe the problem's key characteristics and main challenges. We start with profiles of our primary variables and zoom in on the carbonation front, where all the numerical issues come from.

In short, the problem is reaction-dominant but diffusion-controlled. The main challenge is to resolve reactions, in particular the carbonation that often leads to negative concentration due to undershoots in the nonlinear solver. The reaction speed is limited by the amount of reactants, i.e. how quickly reactants can reach the reaction site.

We will compare Newton and Raspen solvers on 1D and 2D examples⁶. Our main tools for measuring solver efficiency will be the length of the time step and the computation time.

4.5.1 Graphs of Primary Variables

Concrete Zonation

The concrete can be split into four zones: untouched zone, leaching zone, carbonation front, and carbonated zone. The zones always appear present in this order. Concrete starts pristine, gets leached, and eventually carbonated. The best way to distinguish the zones is to observe concentrations of calcium and carbon dioxide⁷. Profiles of calcium molar concentration, carbon dioxide concentration, and porosity are depicted in Figure 4.3, concentrations were rescaled to fit the graph. Zones as listed appear from right to left.

The untouched zone has no carbon dioxide and calcium concentration is close to the initial condition (nearly saturated). Porosity is in its initial state too.

Calcium molar concentration in the leaching zone covers a wide range. It is close to its maximum next to the untouched zone and decreases almost to zero as we get closer to the carbonation front. Porosity increases with decreasing calcium

⁶Actually, 1D case is computed on a 2D mesh with a symmetry. The top and the bottom are insulated and gravity is ignored. The purpose is to make the 1D and the 2D case comparable. Their system sizes and matrix patterns are identical.

⁷Either can be used as a proxy to the pH of the pore solution. More calcium means alkaline and more carbon dioxide makes it acidic.



Fig. 4.3: Profiles of calcium molar concentration, carbon dioxide concentration, and porosity at 213 days.

molar concentration – minerals get leached from the medium. Carbon dioxide concentration is zero in this zone.

The carbonation front is the only zone where concentrations of both calcium and carbon dioxide are positive. The carbonation reaction takes place in this zone causing the deposition of the calcite which decreases porosity. The carbonation is a quick reaction which makes this zone rather thin. In Figure 4.3 it is only one cell wide.

The carbonated zone has no calcium but carbon dioxide concentration is positive. It is almost zero next to the carbonation front and rises to its maximum at the boundary. Porosity does not change in time, but its profile can be complicated. It depends on how much calcite had been deposited when the carbonation front crossed that place.

In our setting, shown in Figure 4.3, porosity inside the carbonated zone is roughly constant and lower than the initial porosity. Carbon dioxide infiltrates in high amounts. The carbonation front advances quickly and the calcium does not have time to move towards it. The leached zone is thus relatively thin and the amount of deposited calcite is unchanging – it corresponds to the amount of calcium released by leaching.

If we reduce the carbon dioxide content in the air, the carbonation front becomes slower. Porosity of the carbonated zone becomes lower, because more calcium diffuses towards the carbonation front. This has a self-reinforcing effect: Lower porosity further slows down the intrusion of carbon dioxide allowing even more



Fig. 4.4: Profiles of calcium molar concentration, carbon dioxide concentration, and porosity at 213 days. Carbon dioxide concentration at the boundary is 11%.



Fig. 4.5: Profiles of calcium molar concentration, carbon dioxide concentration, and porosity at 232 days. Carbon dioxide concentration at the boundary is 11%.



Fig. 4.6: Comparison of the porosity, calcium molar concentration, and carbon dioxide concentration profiles calculated at 175 days on graphs with 512 and 1024 cells.

calcium to reach the carbonation front. The porosity might even reach zero⁸, which happens in the example with 11% carbon dioxide concentration too. Graphs 4.4 and 4.5 depict the porosity, calcium molar concentration, and carbon dioxide concentration of this setting.

Unfortunately, the self-reinforcing effect is also sensitive to the mesh size. The finer the mesh is, the sharper the profiles become. On the grid with 512 elements in x direction, (the size used for graphs shown in this section,) the carbon dioxide concentration of 12% leads to minimum porosity of 0.039 after two years. When $c_a = 11.5\%$ zero porosity is reached after 412 days, and when $c_a = 11\%$ or $c_a =$ 10.5% after 232 days. On the coarser mesh with 256 elements, porosity remains above 0.02 after two years when carbon dioxide concentration is 11%. At $c_a = 10\%$, porosity reaches zero after 620 days.

The setting with $c_a = 0.5$ at the boundary is converged on the grid with 512 cells in the x direction. The comparison of the porosity, calcium molar concentration, and carbon dioxide concentration on grids with 512 and 1024 cells in x direction is plotted in Figure 4.6. Refining the mesh does not significantly change profiles. They become sharper⁹, but their position does not change. In particular, porosity has a sharper and narrower peak, and the carbonation front is still only one cell wide. The carbonation front on the finer mesh lags one (finer) cell behind the carbonation

⁸Our model can not reach exactly zero porosity, the system would not be well defined. The minimum porosity in Figure 4.5 is $2.5 \cdot 10^{-4}$.

⁹This is the property of the finite volume scheme. Each degree of freedom represents the average of the measured quantity inside the control volume. Refining the mesh and splitting the control volume allows its parts to attain higher and lower values and still have the same average.



Fig. 4.7: Time evolution of carbon dioxide concentration.

front on the coarser mesh. The position does not change throughout the computation, it is one cell behind in the initial condition, as well as in the final frame. The speed of the carbonation front is identical.

Evolution in Time

Graphs 4.7-4.11 capture the evolution in time of all primary variables. This setting used boundary condition $c_a = 0.5$.

Figure 4.7 depicts the concentration of carbon dioxide at preselected times. It reaches its maximum value at the boundary where it is prescribed and decreases to zero at the place where the carbonation front ends and leached zone starts.

The profiles of calcium concentration are shown in Figure 4.8. The carbonation front is rather fast due to the high concentration of carbon dioxide. Calcium does not diffuse much towards the front before the front reaches the location from where the calcium was leached. As a result, the leached zone is thin and the slope of the calcium concentration is steep. The leached zone becomes wider later in the simulation. Carbon dioxide needs to travel further to reach the carbonation front and it progresses slower.

Figure 4.9 shows porosity. We can observe, that porosity does not change in the carbonated zone and that the leached zone becomes progressively wider with higher porosity, which is caused by the deceleration the carbonation front and widening of the leached zone. Calcium concentration gets lower before the carbonation



Fig. 4.8: Time evolution of calcium molar concentration.



Fig. 4.9: Time evolution of porosity.



Fig. 4.10: Time evolution of air saturation.



Fig. 4.11: Time evolution of water pressure in kPa.



Fig. 4.12: Time evolution of carbon dioxide concentration and porosity. Setting with 11% carbon dioxide concentration.

front reaches the place, so more minerals can dissolve before calcite starts being deposited.

Finally, Figures 4.10 and 4.11 describe profiles of air saturation and water pressure. They do not include the profile on the day 16. Like profiles from other times, they resemble a piecewise linear function with two parts that break at x = 0.0006 m (the position of the carbonation front), but their bigger ranges make other profiles unreadable. Air saturation continues to decrease to 0.12 and water pressure rises to -28703 kPa within the shown cutout of the domain¹⁰. The saturation and pressure equilibrate at around 60 days. Afterward, the transport of carbon dioxide and calcium is dominated by molecular diffusion. The graph at 213 days seems to be an outlier, but such changes in saturation and water pressure are present at other times. One could say that the other three profiles are deceptively uniform. We ascribe these changes to the effects of chemical reactions. Carbonation and leaching alter the porosity and leaching also releases water. This affects saturation and in turn water pressure via capillary pressure.

We also prepared time evolutions of carbon dioxide and porosity in the setting with lower carbon dioxide content. They can be found in Figure 4.12.

Other Test Cases

We use three test cases which differ in the type of boundary condition on the top and in the strength of the gravity. 1D test case has no gravity and top boundary is

 $^{^{10}}$ Initial air saturation is 0.051 and initial water pressure (equal to atmospheric minus capillary pressure) is $-15325.5~{\rm kPa}.$



Fig. 4.13: Saturation at the top and bottom boundary of the domain on day 175. Test case with gravity and insulated top boundary.

insulated. The problem has thus a symmetry and all DoFs with the same x coordinate have the same value. We keep using the rectangular mesh with 512×128 cells to make the numerical effort comparable to the second test case. Graphs in the previous section come from this setting.

The second test case has an insulated top boundary and gravity is not neglected. Boundary and initial conditions are marginally different from the 1D case. Porosity, calcium molar concentration, and carbon dioxide concentration are the same. The pressure decreases with height so that flow in the y direction is zero, i.e.

$$\partial_y p_i - g\rho_i = 0, \quad i \in \{a, w\}.$$

$$(4.33)$$

Water pressure decreases faster than air pressure due to the density difference¹¹. As a result, capillary pressure increases with height and air saturation too. The size of the domain makes the effect minuscule, less than 0.005% (about 500 Pa of pressure and $2.5 \cdot 10^{-6}$ of saturation). The value on the bottom of the domain has the same value as in the 1D case. The same holds for the value of the Dirichlet boundary on the left side.

Even during the simulation, gravity does not have a significant effect on the values of primary unknowns, they look like in the 1D test case¹². Figure 4.13 depicts the saturation at the top and bottom boundary at 175 days, where is the biggest

¹¹Water pressure decreases linearly, but air density depends on its pressure so air pressure decreases exponentially.

¹²Mostly because the water level evens out relatively quickly and afterward the transport of chemical agents is dominated by molecular diffusion, which is not influenced by gravity.



Fig. 4.14: The third test case. Water pressure and air saturation on day 175.

relative difference from among primary variables. Due to the similarity to the 1D case, we do not present graphs of all primary variables for this test case.

The third test case has the top boundary in contact with air. It has the Dirichlet boundary condition of the same value as the left boundary at this height. With the exception of the top and the left boundary (where the Dirichlet boundary value is imposed), the initial condition is identical to the previous case with gravity. Values of primary variables at the end of the simulation are shown in figures 4.14, 4.15, and 4.16. Since the final time is 175 days, the leaching zone is still only one cell wide. The mesh size is 256×512 cells. The carbonated zone has the highest porosity in the top left corner, where the boundary is in contact with air from two sides.

Placing the Dirichlet boundary condition representing the contact with air on top of the sample is greatly different from placing it on the bottom of the sample.



Fig. 4.15: The third test case. Calcium molar concentration and carbon dioxide concentration on day 175.



Fig. 4.16: The third test case. Porosity on day 175.

The latter leads to the instability associated with the gravity fingering phenomenon [DiC13]. Our model is not equipped to resolve such instability¹³.

4.5.2 The Effect of the Projected Line Search

The projection step in the line search method projects all negative calcium and carbon dioxide concentrations to zero. This is done before the defect is calculated and the solver's convergence criterion remains unchanged. Therefore, barring numerical errors, the solver converges to the same solution, although if the true solution had a negative concentration, the solver would be unable to converge. However, we expect the concentrations to be nonnegative. The projection step is used to improve the convergence speed in situations when negative concentrations were spotted in intermediate solutions.

We explain the impact of the projection step by observing the length of the time step. The time step size is adjusted to the problem's difficulty. If the solver fails to converge, the time step size is shortened. This makes the problem easier because the bigger weight on the mass term makes the matrix better conditioned. Moreover, the solution will get closer to the previous time step's solution that is used as an initial guess. On the other hand, if the problem is too easy, i.e. the solver converges in just a few iterations, the time step size is lengthened.

Figure 4.17 depicts the length of the time step used throughout the computation. We look at the time step size used in the particular moment of the simulation. This makes it easier to compare graphs of time steps because different solvers use a different number of time steps. The 1D test case and the test case with gravity were chosen to emphasize the dependence of the time step size on the position of the carbonation front relative to the mesh. The problem is actually solved on 2D mesh in both cases, so they have the same number of DoFs. The carbonation front in the 1D test case is vertical, but gravity disrupts the symmetry in the other test case.

Using a traditional solver makes the time step size follow an oscillating pattern. Each time the solver encounters a tough spot the time step size gets drastically reduced because the solver repeatedly fails to converge. Once the time step is small enough, the computation can advance past the critical point. The time step size then quickly grows back.

The dips in the time step size correspond to the carbonation front crossing into a new cell. It is accompanied by calcium concentration reaching zero, which is the cause of the convergence problems. Solutions sometimes also have negative calcium concentration, which means they are discarded in the operator-splitting

¹³The most common approach is to add a dynamic component (time derivative of the saturation) to the capillary pressure formula [RS14].



(d) The flow and the chemical part use Raspen solver. Test case with gravity

Fig. 4.17: The logarithm of the time step size throughout the computation with and without the projected line search. y axis is the decadic logarithm of the time step size, and x axis is the accumulation of all previous time steps (the time in the simulation).



(d) Raspen solver without projection on the test case with gravity. 6895 time steps.

Fig. 4.18: The logarithm of the time step size throughout the computation. Neither solver uses projected line search. Unlike graphs in Figure 4.17 where x axis is scaled to the simulation time, here it is the time step number.

Solver	Line search	Test case	Time steps	Runtime [h]
Newton	none	no gravity	1510	1.27
Newton	projected	no gravity	501	0.51
Raspen	none	no gravity	1434	3.88
Raspen	projected	no gravity	344	1.79
Newton	none	with gravity	13272	6.20
Newton	projected	with gravity	602	0.65
Raspen	none	with gravity	6895	9.86
Raspen	projected	with gravity	352	2.08

Tab. 4.6: Effect of the projection step on Newton and Raspen solver's runtime.

convergence check (if they are past the threshold). The convergence problems were eventually resolved by adding the projection step to the line search method of the nonlinear solver, which prevents the calcium concentration from becoming negative. The time step size remains close to its maximum, some smaller dips are still present. The source of these dips is unknown to us, they are not caused by calcium undershoots nor by adjusting time step size to hit the output frame moment¹⁴.

Raspen solver benefited from the projection step more than the Newton solver. Even though it is able to take longer time steps, its iterations are more expensive, and overall computation time is longer. Table 4.6 contains the number of time steps and the computational time solvers used in the setting with and without gravity.

When gravity is considered, the symmetry is disrupted. The initial condition is the same as before, so the carbonation front starts aligned with mesh and gets more skewed over time. As a result, the carbonation front traverses fewer DoFs at once compared to the 1D example, but more often. The effect is small, the carbonation front is not even visibly shifted¹⁵. The graphs of time steps resemble the case without gravity.

For this reason, we added Figure 4.18. It again shows the logarithms of the time step sizes of Newton and Raspen solver without the projection on tests with and without gravity. However, this time the x axis is not scaled to the simulation time. We can see that solvers need only a couple of short steps in each dip in the case without gravity. On the other hand, in the test case with gravity, most time steps are short, and the number of short time steps in dips grows during the simulation. We conclude that the convergence problems associated with the carbonation front

¹⁴Output is created at predetermined moments and time step size is lowered to match it. The time step size is adjusted two steps in advance, so the worst case scenario is the halving of the time step size.

 $^{^{15}}$ The transport of the calcium and carbon dioxide is dominated by diffusion which is not directly affected by gravity. For the sample of height 5 cm the difference in water saturation on top and bottom in steady state is small: 0.005%.

Overall	Time	Failed		Flow			Contaminan	t
time	steps	opsp.	Solver	Appl.	Time	Solver	Appl.	Time [h]
			Test cas	se witho	ut gravi	ty (1D):		
0.51	501	13	Newton	3521	0.16	Newton	3271(3521)	0.18(0.28)
0.63	333	22	Newton	2687	0.12	Raspen	2511(2687)	0.37(0.47)
1.95	489	7	Raspen	3484	1.62	Newton	3250(3484)	0.17(0.27)
1.79	344	17	Raspen	2760	1.28	Raspen	2587(2760)	0.37(0.47)
			Test c	ase with	ı gravity	(2D):		
0.61	602	11	Newton	4057	0.19	Newton	3689(4057)	0.20(0.35)
0.65	344	29	Newton	2769	0.12	Raspen	2314(2769)	0.38(0.49)
2.19	568	9	Raspen	3794	1.82	Newton	3489(3794)	0.19(0.31)
2.08	352	19	Raspen	2783	1.35	Raspen	2597(2783)	0.57(0.68)

Tab. 4.7: Performance of different solver combinations for flow and contaminant. Numbers in parenthesis include attempts that failed to converge. Time is in hours.

entering a new cell are present already when only a few DoFs are affected, and they were not enhanced in the test case without gravity by having the whole column of cells undergoing the critical phase simultaneously.

Table 4.6 and graphs of time steps sizes in this subsection were generated on a mesh with 256×64 cells (width \times height). The domain was split into 32 subdomains with overlap region 8 cells wide. The reader shall be reminded that the 1D test case uses the same mesh and neglecting the gravity gives the problem symmetry. Due to its effectiveness, all results in other sections use the projected line search method.

4.5.3 Performance of Newton and Raspen solver

In Subsection 4.5.2, we described the impact of the projected line search on Newton and Raspen solver performance. In this section, we take a more in-depth look at Newton and Raspen solver. Both solvers use the projected line search.

Before we start comparing the solvers, we have to add a disclaimer. The relative effectiveness of the Newton and Raspen solver depends on the problem in question. Newton proved to be a more efficient solver than Raspen for our problem, but even the same model with different parameters can lead to different outcomes¹⁶.

The main results of this subsection are summarized in Table 4.7. It shows the number of Newton and Raspen solver applications and the computation time separately for the flow and the contaminant part. Data are presented for all four solver com-

¹⁶We even worked with such parameters for some time. Raspen solver was still more expensive but was able to make so much longer (and thus fewer) time steps that the total computation time was shorter. The parameters proved to be incorrect (misnaming of hydraulic conductivity with permeability in [BS04] confused us), and had a more difficult flow part.

binations, i.e. operator-splitting parts use different solver types. From the data in the table, we conclude that:

- The contaminant part is more difficult to solve. There are no solver failures in the flow part, whereas depending on the solver and test case, the contaminant part has a convergence failure rate of 4 10%. These failures are the main factor limiting the time step size.
- Newton solver is faster than Raspen. The comparison is indirect at best but the difference is quite profound: It is twice as fast in the contaminant part and six to seven times as fast in the flow part.
- Raspen solver is more robust in the contaminant part than Newton solver. Using Raspen results in 30-40% fewer time steps. (Which decreases the computation time of the solver in the flow part.) Data from the flow part are less conclusive. Newton solver in the flow part is used by the run with the highest and also the run with the lowest number of time steps.

Solver applications in the flow and contaminant part include uses from operatorsplitting iterations that failed to converge (third column of Table 4.7). Thanks to the projected line search, the operator-splitting does not fail due to negative concentrations. All failures can be attributed to exceeding the maximum limit of iterations, which is 10.

The overall time is more than the sum of flow and contaminant computation times, the extra time is around 0.4 seconds per time step. The time outside of solvers is spent predominantly on the defect calculation for the operator-splitting. Smaller contributions come from writing the output (30 snapshots at predetermined times; and for each time step its size, number of operator-splitting iterations, number of projections, and the largest projection), and keeping a copy of the previous time step's solution to be able to recover when a solver fails to converge.

4.5.4 Parallel Scaling

We parallelize the code with the additive Schwarz domain decomposition method. Unfortunately, it is impossible to spread the workload evenly across subdomains. The untouched zone and the carbonated zone are easy to resolve since they do not experience chemical reactions. Leaching zone and more so the carbonation front are more demanding. They are also thin, the area that requires the most computational effort is localized. Therefore, subdomains that contain the carbonation front are the bottleneck of the computation. Reducing the size of the subdomain that holds the carbonation front still helps though. The computer has two Intel(R) Xeon(R) CPU E5-2698 v3 @ 2.30GHz processors in one socket. Together they have 32 physical cores. Processors have the hyper-threading feature, but the bottleneck is the arithmetic unit and each physical core has only one. Turbo boost was disabled.

Results presented in this section use Newton solver with projected line search. The added overlap width of four cells¹⁷ was used for the mesh 256×64 . The number of cells in the overlap doubles with refinement, while its width in meters stays constant. We also tried a smaller overlap of two cells. It was sufficient for the computation on two subdomains, but on four subdomains, the 1D test case took 15.33 h. Increasing the overlap width to four cells solved the issue.

Tables 4.8 and 4.9 show performance statistics for the 1D test case and the test case with gravity respectively. The tables list the computation time, number of time steps, and, separately for each operator-splitting part, the average number of nonlinear iterations used to converge, the average number of linear iterations per nonlinear iteration, the time spent on assembling the residual vector and the Jacobian matrix, and the time spent on solving the system.

Due to the iterative operator-splitting, each time step uses solvers several times. 1D test case used on average 6.7, 5.2, and 3.9 operator-splitting iterations per time step on meshes 256×64 , 512×128 , and 1024×256 respectively. The test case with gravity used on average 6.4, 5.1, and 3.8 operator-splitting iterations per time step. Runs with more (and shorter) time steps tend to need fewer operator-splitting iterations. Runs on bigger meshes have also fewer convergence failures (0-2, whereas the smallest mesh has 9-14) due to exceeding the maximum number of operator-splitting iterations.

The number of iterations in tables 4.8 and 4.9 is calculated only from the solver applications that converged, although the times also include failed attempts. The nonlinear solver of the flow part has never failed to converge, with the exception of runs on the 1024×256 mesh on the test case with gravity, where it failed twice out of circa 4000 applications. The nonlinear solver of the contaminant part not only needed more nonlinear solver applications, but it also failed in circa 7% of them in the 1D case and in 8% in the case with the gravity. Always due to exceeding the maximum number of iterations (20). As a result, around 40% of the contaminant part's nonlinear iterations were wasted.

Each linear iteration consists of solving the linearized subdomain problem and communicating the correction from the overlap region. A higher number of linear iterations thus indicates a bigger need to communicate. The sequential run does not split the domain, the linearized problem is solved directly, which is counted as

¹⁷Four cells of overlap are added to each subdomain. The overlap region is eight cells wide.

	Comp.	Time		Flow p	art		Contaminant part			
Ranks	time	stops	N-l. it.	L. it. per	Asse.	Solve	N-l. it.	L. it. per	Asse.	Solve
	[h]	steps	per use	n-l. it.	time	time	per use	n-l. it.	time	time
1	5.87	414	1.83	1	0.38	0.58	2.37	1	0.81	3.21
2	3.23	414	1.86	3.94	0.20	0.43	2.37	1	0.43	1.71
4	1.80	423	1.93	9.20	0.12	0.34	2.48	1.07	0.25	0.82
8	1.16	456	1.91	13.50	0.07	0.24	2.45	1.08	0.15	0.53
16	0.72	453	1.89	13.48	0.04	0.17	2.45	1.11	0.08	0.32
32	0.51	501	1.87	20.42	0.02	0.14	2.52	1.12	0.05	0.23
64 ht	0.52	493	1.87	18.79	0.02	0.15	2.56	1.18	0.05	0.23

Tab. 4.8: Scaling results on the test case without gravity (called 1D because of the symmetry). Mesh size is 256×64 cells.

ht stands for hyper-threading

On the mesh of size 512×128 .

2	17.30	501	1.85	3.09	0.76	1.90	2.57	1	1.72	11.12
4	9.83	547	1.92	8.51	0.48	1.75	2.67	1.04	1.09	5.40
8	5.86	574	1.90	12.63	0.26	1.26	2.65	1.04	0.62	3.11
16	3.80	608	1.89	12.67	0.15	0.91	2.71	1.05	0.39	1.90
32	2.69	620	1.88	19.00	0.08	0.84	2.75	1.06	0.21	1.30

On the mesh of size 1024×256 .

16	19.47	807	1.83	11.20	0.60	4.10	3.02	1.01	1.59	11.36
32	13.68	882	1.82	16.54	0.34	3.95	3.00	1.02	0.92	7.38

	Comp.	Time		Flow p	art		Contaminant part			
Ranks	time	Time	N-l. it.	L. it. per	Asse.	Solve	N-l. it.	L. it. per	Asse.	Solve
	[h]	steps	per use	n-l. it.	time	time	per use	n-l. it.	time	time
1	6.67	465	1.86	1	0.42	0.63	2.39	1	0.94	3.70
2	3.66	465	1.89	3.89	0.22	0.47	2.39	1	0.50	1.96
4	1.99	496	1.95	8.83	0.14	0.37	2.49	1.05	0.31	0.98
8	1.08	441	1.90	13.63	0.06	0.23	2.46	1.07	0.14	0.49
16	0.83	543	1.91	13.31	0.04	0.19	2.47	1.08	0.10	0.38
32	0.61	602	1.89	21.02	0.02	0.16	2.52	1.13	0.06	0.29

Tab. 4.9: Scaling results on the test case with gravity. Mesh size is 256×64 cells.

On the mesh of size 512×128 .

2	17.54	507	1.87	3.39	0.76	1.98	2.58	1	1.76	11.24
4	10.97	597	1.94	8.32	0.52	1.88	2.68	1.04	1.25	6.14
8	6.00	576	1.91	12.62	0.26	1.27	2.67	1.04	0.64	3.20
16	3.71	597	1.89	12.62	0.15	0.89	2.70	1.06	0.38	1.85
32	2.78	655	1.88	17.71	0.08	0.86	2.73	1.08	0.22	1.35

On the mesh of size 1024×256 .

16	21.95	888	1.84	10.75	0.65	4.42	2.97	1.005	1.83	13.05
32	16.34	1018	1.84	15.27	0.39	4.31	2.94	1.014	1.17	9.20

92

one iteration. The average number of linear iterations is the same for the nonlinear iterations that led to a converged state and those that did not.

Data from tables 4.8 and 4.9 show that increasing the number of subdomains is met with an increase in the number of time steps and nonlinear iterations, i.e. the problem is getting more difficult. Since moving from one to two ranks does not make a difference, it might be related to the presence of the carbonation front in the overlap region.

The contaminant part needs very few communications to converge, the number stays close to one per nonlinear iteration. This signifies that the contaminant part's problem is localized, which is not surprising after seeing the shape of the carbonation front. The workload is not spread evenly, but the toughest subdomain does get smaller. As a result, the assemble and solve time scales better for the contaminant than the flow part.

On the other hand, the flow part is not localized. Increasing the number of subdomains also increases the number of overlap communications. The increase could be prevented by using a solver with a coarse correction, as was demonstrated by Chaiyod Kamthorncharoen in his dissertation [Kam22].

The third test case, the one with the left and the top boundary in contact with air, uses a different mesh because the previous mesh was too coarse in the y direction and the leached zone spreading from the top boundary had much lower porosity. The results in Table 4.10 use the mesh with 128×256 cells, which is one refinement level less than what the figures 4.15 - 4.16 use¹⁸.

The robustness of the nonlinear solver proved crucial. Table 4.10 uses Raspen solver for the contaminant part. For Raspen, the nonlinear iteration is the preconditioned Newton iteration (as written in the equation 3.44), and the number of linear iterations reflects how many GMRES iterations were used to solve the linearized (preconditioned) system. (Raspen subdomain solves are independent, and each rank can have a different number of nonlinear iterations. Linearized subdomain problems are solved exactly with UMFPack.) The assemble time comes from the rank 0 process, which has the bottom left subdomain. The total time also includes the assemble time and other parts that are not accounted for in the Newton solver's solve time like the calculation of the initial defect.

We tried to make strong scaling results with the Newton solver in the contaminant part, but the time step size was too small and the computation progressed too slowly. The computation on 32 ranks was fine, but computations on 8 and 16 ranks

 $^{^{18}}$ For these meshes the maximum porosity of the leached zone is also smaller. On the 256×512 mesh, the porosity of the leached zone reaches 0.12 and up to 0.14 next to the top left corner. On the 128×256 mesh, the porosity reaches 0.10 with a few spikes of 0.12 in the corner.

Ranks	Comp.	Time steps	Flow part				Contaminant part			
	time		N-l. it.	L. it. per	Asse.	Solve	N-l. it.	L. it. per	Asse.	Total
	[h]		per use	n-l. it.	time	time	per use	n-l. it.	time	time
4	60.92	1933	2.22	4.95	0.92	2.24	2.59	2.07	2.42	56.16
8	27.44	1975	2.22	6.65	0.49	1.53	2.61	2.15	1.25	24.45
16	12.99	1976	2.20	7.38	0.25	0.90	2.59	2.21	0.55	11.30
32	7.85	2120	2.19	10.09	0.15	0.71	2.75	2.45	0.32	6.65
64 ht	6.31	2110	2.19	10.78	0.12	0.72	2.77	2.48	0.27	5.17

Tab. 4.10: Scaling results on the test case with top boundary in contact with air. Mesh size is 128×256 cells.Contaminant part uses Raspen solver.

On the mesh of size 256×512 . The contaminant part uses Raspen solver.

							1	1		
32	100.77	6179	2.25	7.28	1.30	7.52	3.01	2.50	2.52	88.97

On the mesh of size 128×256 . The contaminant part uses Newton solver.	Solve							
Computations on 8 and 16 ranks did not finish in 40 hours.								

32	4.92	3801	2.05	9.84	0.18	0.88	2.98	1.88	0.43	2.88
64 ht	4.31	4027	2.03	10.40	0.15	0.92	3.09	1.82	0.39	2.37

had to be stopped for taking too long¹⁹. We do not have an explanation for why the computation on more ranks worked.

Raspen solver can finish the computation on fewer ranks, but the computation time is more than twice as long as when the number of ranks doubles. This might be a sign of a bad workload balance. The top left subdomain, where the carbonation front curves, solves a harder problem than others. Making the toughest subdomain smaller targets the computation bottleneck, and other ranks spend less time idling. The run on 64 ranks confirms this theory – the computation time was decreased even though the computer has only 32 physical cores. However, Newton solver uses the same amount of iterations on all subdomains, so the cause for the time decrease going from 32 to 64 ranks remains unclear. We note that the Newton solver experiences a smaller time decrease than the Raspen solver.

4.6 Remarks on Unfruitful Attempts

Before the projected line search was tried, we experimented with several other methods to address the problem of the time step reduction (see Section 4.5.2). Neither worked, although in hindsight we got the explanation for some. The list is not exhaustive...

The first was the L-scheme [Bor+18]. A linearization scheme similar to Newton's method with one extra term – the difference between the current and the previous iteration. It has a better convergence radius than Newton's method but is only linearly convergent. The L-scheme indeed converged at a slower rate than Newton's method, but the time step size showed a similar pattern. Finding optimal damping parameter L that multiplies the added difference is tough for such a nonlinear system, so we can not dismiss the option that the parameters we tried were not close enough. (L has to be large enough to ensure convergence, but smaller L has a better convergence rate.) In hindsight, enlarging the convergence radius could not solve the issue. The problem was not getting close to the solution but to control the undershoots into negative concentrations.

Another attempt was smoothing the leaching functions L_w , L_{Ca} , and L_ϕ . The hypothesis was that the convergence problems were caused by the lack of the regularity of the problem and these functions were the only nonsmooth terms. Two types of regularization were used, both local. The first was a cubic interpolation in the neighborhood of points where two linear parts meet²⁰, and the interpolation matched values and the first derivation of linear functions at its ends. This, however, did

 $^{^{19}\}mbox{After}$ computing for more than 40 hours the simulations reached only 110 days out of 175.

²⁰We also tinkered with the width of the interpolated region.

not conserve the total amount of leached material. The most noticeable difference was that the speed of the carbonation front changed, it got faster. The computation time improved a little too, but we were solving a different problem. The second regularization attempt respected the total amount of leached material. The resulting profiles looked unaffected but computation time improvement was only about 3-5%. Since the gains were so small we decided to proceed without regularization and use leaching functions as presented.

Another unsuccessful attempt was substepping the contaminant part. The idea was to let the flow part make two or three times longer time steps and save some work. We tried two approaches: linearly interpolating flow data, and using previous step's flow data in all but the last substep. The latter worked better. We suppose the issue was that simple interpolation does not respect the nonlinear relation between pressure and saturation. Calculated fluxes did not match well the saturation changes, and as a result, mass-balance equations had problems converging. Overall, it was not worth it. Worse flow data made the contaminant part more difficult, which caused the time step size to be reduced.

We also tried to use the operator-splitting to split the advection, diffusion, and reaction of the contaminant part further. Several splitting variants²¹ were tried but none was better than the presented option. The monolithic approach (not splitting the system into the flow and the contaminant part) works well. It is just slower and has the same problem with the time step size reduction.

²¹All combinations of splitting contaminant into two or three subparts were tried, both in Lie-Trotter splitting and Strang splitting with any ordering of subparts.
5

Second Model: Multicomponent Multiphase Flow with Phase Disappearance

The second model is a benchmark proposed by SiTraM community¹ [Hoo+24]. The benchmark features four increasingly complex scenarios. We will present the first three, as our approach was too inefficient for the last scenario.

The benchmark was inspired by geological storage of CO_2 . The benchmark considers three phases: solid, liquid, and gas. The solid phase is the simplest as it is immovable and has only one component, the calcite mineral participating in chemical reactions. The gas phase has two components: water vapor and carbon dioxide. In the first three benchmark scenarios, the liquid phase holds water, carbon dioxide, and ions Ca^{+2} and CO_3^{-2} . There is one chemical reaction: carbonation. The fourth scenario also includes ions H^+ , OH^- , and HCO_3^- . There are four chemical reactions. Carbonation gets adjusted and the three remaining reactions describe chemical equilibria between ions.

The first two scenarios are one-dimensional. They are very similar and differ only in the type of chemical reaction. The first uses kinetic reaction rate and the second maintains (dynamical) chemical equilibrium. The third and the fourth scenarios are two-dimensional. They add the effects of gravity, have nonuniform initial porosity, and feature more complicated flow fields.

In all scenarios the domain starts fully saturated with the liquid phase that contains water, ions in chemical equilibrium, but no carbon dioxide. Carbon dioxide is injected into the domain on the left side, and fluids can outflow freely on the right side. The injection first creates an unsaturated and then a gas-saturated region. Throughout most of the simulation, the domain contains a liquid-saturated, an unsaturated, and a gas-saturated region.

Handling the phase appearance and disappearance is the main challenge of the benchmark. To a great extent, it is simplified by the fact that capillary pressure is neglected. All phases are under the same pressure, so the pressure is welldefined under all conditions. However, molar fractions can become ill-defined, especially those belonging to components that are present in only one phase. Our

¹SiTraM is an abbreviation derived from advances in SImulation of reactive flow and TRAnsport in porous Media.

key contribution lies in using complementarity constraints to regularize the system by assigning a unique value to molar fractions when their phase disappears.

5.1 Governing Equations

We will follow the benchmark notation [Hoo+24] with some simplifications. The benchmark definition is rather general and can be made clearer by using the fact that there are two liquid phases, and the solid phase contains only one mineral.

The porous medium contains one mineral that participates in chemical reactions, and solids that do not. Nonreacting solids will never enter our equations and only take up some space. Variable ϕ^T denotes the total porosity, the space populated by liquid, gas, and reactive solid species. The benchmark uses $\phi^T = 1$, so there are no nonreactive solid species.

The mass balance equation for the mineral m (calcite) is

$$\partial_t c_m = R,\tag{5.1}$$

where c_m is its molar concentration (mol/m³), and R the reaction rate. Depending on the scenario, the benchmark uses different formulas for the reaction rate. We will describe them later.

The mineral saturation \hat{s}_m is calculated from its molar concentration

$$\hat{s}_m = \frac{M_m}{\tilde{\rho}_m} c_m \tag{5.2}$$

using its molecular weight M_m and the density of the solid phase $\tilde{\rho}_m$. The fraction of the total volume occupied by the mineral m is $\phi^T \hat{s}_m$ thus the porosity (the volume fraction filled by liquid and gas phase) is

$$\phi = \phi^T - \phi^T \hat{s}_m. \tag{5.3}$$

If more minerals were present, their saturation would be calculated the same way and the equation for porosity 5.3 would get additional terms to be subtracted.

The general mass balance equation for a component c present in fluid phases has the form

$$\partial_t n_c + l_c + q_c = \sum_{k=1}^K v_{ck} r_k^K + \sum_{q=1}^Q v_{cq} r_q^Q, \quad c = 1, \dots, C,$$
(5.4)

where *C* is the number of components in fluid phases², n_c represents the mass of the component *c*, l_c is the total flux of the component *c*, and q_c is the source/sink term that is nonzero only at the location of the well. The benchmark scenarios have inflow and outflow situated at the boundaries and we implement them via boundary conditions so q_c is zero everywhere. The right-hand side describes chemical reactions. v_{ck} are stoichiometric coefficients and r_k^K are reaction rates for kinetic reactions, whereas v_{cq} and r_q^Q are the stoichiometric coefficients and reaction rates for equilibrium reactions³.

Denoting the liquid phase with the index l, and the gas phase with the index g, the mass term n_c is

$$n_c = \phi \sum_{j \in \{l,g\}} (\rho_j S_j x_{cj}), \quad c = 1, \dots, C.$$
 (5.5)

 ϕ is porosity. S_j is the saturation of the phase j. Phases fill this space completely, so $S_l + S_g = 1$. ρ_j is the molar density of the phase j, and x_{cj} is the molar fraction of the component c in the phase j.

Molar fractions x_{cj} are the main difference to our previous model, and it is what makes this problem *multicomponent*. x_{cj} represents what fraction of all moles forming the phase j belongs to the component c. They are dimensionless variables with values from zero to one. Each phase contains multiple components and each phase is completely comprised of its components, therefore

$$\sum_{c=1}^{C} x_{cj} = 1, \quad j \in \{l, g\}.$$
(5.6)

The benchmark has two components that are present in both the liquid and the gas phases: water and carbon dioxide. The remaining components are present only in the liquid phase, so their molar fraction of the gas phase is always zero⁴.

The flux of the component c is defined as

$$l_c = \nabla \cdot \sum_{j \in \{l,g\}} \left(\rho_j x_{cj} \vec{v}_j - \rho_j \phi S_j d \nabla x_{cj} \right), \quad c = 1, \dots, C.$$
(5.7)

²For convenience, we will later use the name of the component instead of its number. E.g., $n_{\rm H_{2}O}$ instead of n_1 .

³Here we deviated from the benchmark's notation but kept the equation 5.4 unchanged. We will index kinetic and equilibrium reactions together, i.e., instead of kinetic reaction 1 and equilibrium reaction 1 we will have reactions 1 and 2.

⁴Adding a small volume to the gas phase could be done in order to regularize the problem. It alleviates the issue with all terms disappearing when liquid saturation becomes zero. We did not use this regularization.

The diffusion coefficient d is the same in both phases in all directions for all components. Another simplification is that capillary pressure is neglected and all phases (even the solid phase) are under the same pressure p. The advection uses Darcy velocity $\vec{v_i}$ defined as

$$\vec{v}_j = -k \frac{k_{rj}}{\mu_j} \left(\nabla p - \tilde{\rho}_j g \nabla h \right), \quad j \in \{l, g\}.$$
(5.8)

Here, k is the intrinsic permeability, k_{rj} is the relative permeability, μ_j is the viscosity, $\tilde{\rho}_k$ is the mass density (kg/m³), g is the strength of the gravity, and h represents the elevation so $-\nabla h$ is the unit vector pointing downwards. The intrinsic permeability k depends on the current porosity, the initial porosity ϕ_0 , and (possibly spatially varying) initial intrinsic permeability k_0 .

$$k = k_0 \left(\frac{\phi}{\phi_0}\right)^3 \tag{5.9}$$

Relative permeability k_{rj} is the function of the saturation. The benchmark uses the Brooks-Corey formula. The formula is the same for both fluids,

$$k_{rj} = k_{rj}^e (S_j)^2, \quad j \in \{l, g\}.$$
 (5.10)

The constant k_{rj}^e is the relative permeability of a medium fully saturated with phase j. This formula enjoys two simplifications. The residual saturations are considered to be zero, and the exponent is the same in both phases.

The mass density depends on the pressure. The relation is linearized around the initial state.

$$\tilde{\rho}_j = \rho_{j,0} \left(1 + C_j (p - p_0) \right), \quad j \in \{l, g, m\}$$
(5.11)

 $\rho_{j,0}$ is the density at initial pressure p_0 , and C_j is the compressibility constant. As Table 5.1 shows, the compressibility of the liquid and the solid phase is low.

The molar density is calculated from the density and molar fractions of the phase. The density of the component c in the phase j is $\rho_j x_{cj} M_c$ and summing over the components yields the density of the phase. Therefore,

$$\rho_j = \frac{\tilde{\rho}_j}{\sum_{c=1}^C x_{cj} M_c}, \quad j \in \{l, g\}.$$
(5.12)

Since the mineral has its own phase, its molar density is simply

$$\rho_m = \frac{\tilde{\rho}_m}{M_m}.$$
(5.13)

Property	Value	Unit
Permeability k_0	100	[mD]
Total porosity ϕ^T	1	[-]
Initial porosity ϕ	0.3	[-]
Density at p_0 : ρ_{l0} , ρ_{g0} , ρ_{m0}	[1000, 100, 2000]	[kg/m ³]
Compressibility C_l , C_g , C_m	$[10^{-6}, 10^{-4}, 10^{-7}]$	[1/bar]
Viscosity μ_l, μ_g	[1, 0.1]	[cP]
Fully-sat. relat. perm. k_{rl}^e , k_{rl}^e	[1, 1]	[-]
Diffusion coef. d	10^{-9}	[m ² /s]
Molecular weight M_m , M_{H_2O} , M_{CO_2}	[100.09, 18.01528, 44.01]	[g/mol]
Molecular weight $M_{Ca^{+2}}$, $M_{CO_3^{-2}}$	[40.078, 60.009]	[g/mol]
Partition coef. $K_{H_2O,l}, K_{CO_2,l}$	[0.1, 10]	[-]
Kinetic reaction const. K_k	1000	[mol/m ³ /day]
Chemical equilibrium const. K_{sp}	0.0625	[-]

Tab. 5.1: Values of constants. The nontraditional units are millidarcy $1 \text{ mD} = 9.869233 \cdot 10^{-16} \text{ m}^2$, and centipoise $1 \text{ cP} = 10^{-3} \text{ Pa s.}$

5.1.1 Chemical reactions

The fugacity of the component c in the phase j is given by

$$f_{cj} = \phi_{cj} x_{cj} p, \quad c = 1, \dots, C, \quad j \in \{l, g\},$$
(5.14)

where ϕ_{cj} is the fugacity coefficient of an ideal mixture. Phases are in thermodynamic equilibrium, i.e. chemical potentials of the components in both phases are equal. Therefore, the fugacity of the same component in both phases is the same, $f_{cl} = f_{cg}$. The simplification that both phases have the same pressure allows us to transform this relation into the form that uses only molar concentrations and the partition coefficients $K_{cl} = \phi_{cg}/\phi_{cl}$:

$$K_{cl}x_{cl} - x_{cg} = 0, \quad c = 1, \dots, C.$$
 (5.15)

For components that are present only in the liquid phase, we simply use $K_{cl} = 0$.

The first three problems of the benchmark have simple chemistry. The only chemical reaction is the carbonation.

$$CaCO_3 \rightleftharpoons Ca^{+2} + CO_3^{-2} \tag{5.16}$$

 Ca^{+2} and CO_3^{-2} are present in the liquid phase but not in the gas phase. The calcite, $CaCO_3$, is the single reactive mineral of the solid phase. In total, there are four components (C = 4) in fluid phases: H₂O, CO₂, Ca⁺², and CO₃⁻². The last,

fourth scenario of the benchmark has more complicated chemistry with three extra components and will be described in Section 5.5.

The reaction is either treated as a kinetic reaction or an equilibrium reaction. The chemical activities of aqueous species are equal to their concentrations, whereas the chemical activity of the solid mineral is constant. The forward reaction rate is thus constant and the backward reaction rate depends on molar concentrations of Ca^{+2} and CO_3^{-2} .

For the kinetic reaction, the reaction rate is

$$r_k = AK_k \left(1 - \frac{x_{\text{Ca}}, l x_{\text{CO}}, l}{K_{sp}} \right),$$
 (5.17)

where A is the reactive surface area, K_k is the kinetic reaction constant, and K_{sp} is the equilibrium constant. The reactive surface area linearly depends on the mineral saturation.

$$A = A_0 \hat{s}_m = (1 - \phi_0) \hat{s}_m \tag{5.18}$$

The calcite saturation in the reactive surface area coefficient A makes the kinetic reaction rate linearly dependent on the calcium concentration. The equilibrium constant is chosen such that molar fractions of Ca^{+2} and CO_3^{-2} are in equilibrium when their molar fractions are 0.25, i.e. $K_{sp} = 0.25^2 = 0.0625$.

In the case of the chemical equilibrium, the reaction rate is unknown. More precisely, the reaction rate is as fast as is necessary to maintain the equilibrium condition

$$1 - \frac{x_{\text{Ca},l} x_{\text{CO}_3,l}}{K_{sp}} = 0 \quad \text{if } c_m > 0.$$
(5.19)

If the calcite concentration is zero, the solution may become undersaturated. The chemical potential would lead to more dissolution, there is just no calcite left. The potential can never be negative because the extra components can always precipitate.

$$1 - \frac{x_{\text{Ca},l} x_{\text{CO}_3,l}}{K_{sp}} \ge 0 \quad \text{if } c_m = 0 \tag{5.20}$$

The chemical activity of calcite is constant, but unlike in the kinetic case, its concentration is not directly present in the formula. We have to treat the case when the concentration is zero carefully, otherwise it eagerly becomes negative. The chemical equilibrium condition (eq. 5.19 and 5.20) has the structure of the complementarity constraint. We require calcite concentration and chemical potential to

be nonnegative, and at least one of them must be zero. In the system, we implement this constraint in the form of the minimum function.

$$\min\left(c_m, 1 - \frac{x_{\text{Ca},l} x_{\text{CO}_3,l}}{K_{sp}}\right) = 0$$
(5.21)

5.1.2 Summary

For the first three benchmark scenarios, the system consists of a mass balance equation for each component,

$$\sum_{j \in \{l,g\}} \partial_t \left(\phi S_j \rho_j x_{\mathrm{H}_2\mathrm{O},j} \right) + \nabla \cdot \left(\vec{v}_j \rho_j x_{\mathrm{H}_2\mathrm{O},j} - \phi S_j \rho_j D \nabla x_{\mathrm{H}_2\mathrm{O},j} \right) = 0$$
(5.22)

$$\sum_{j \in \{l,g\}} \partial_t \left(\phi S_j \rho_j x_{\text{co}_2, j} \right) + \nabla \cdot \left(\vec{v}_j \rho_j x_{\text{co}_2, j} - \phi S_j \rho_j D \nabla x_{\text{co}_2, j} \right) = 0$$
(5.23)

$$\partial_t \left(\phi S_l \rho_l x_{\mathrm{Ca},l} \right) + \nabla \cdot \left(\vec{v_l} \rho_l x_{\mathrm{Ca},l} - \phi S_l \rho_l D \nabla x_{\mathrm{Ca},l} \right) = -R \tag{5.24}$$

$$\partial_t \left(\phi S_l \rho_l x_{\cos_3, l} \right) + \nabla \cdot \left(\vec{v}_l \rho_l x_{\cos_3, l} - \phi S_l \rho_l D \nabla x_{\cos_3, l} \right) = -R \tag{5.25}$$

$$\partial_t c_m = R$$
 (5.26)

and an extra equation if the reaction is in equilibrium and R is unknown.

$$\min\left(c_m, 1 - \frac{x_{\text{Ca},l} x_{\text{CO}_3,l}}{K_{sp}}\right) = 0$$
(5.27)

If the reaction is kinetic, the reaction rate can be directly evaluated as

$$R = -(1 - \phi_0) \frac{M_m}{\tilde{\rho}_m} c_m \left(1 - \frac{x_{\text{Ca},l} x_{\text{Co}_3,l}}{K_{sp}} \right).$$
(5.28)

The system is constrained by fugacity conditions

$$x_{\rm H_2O,l} = 10 \, x_{\rm H_2O,g} \tag{5.29}$$

$$x_{\rm co_2,g} = 10 \, x_{\rm co_2,l} \tag{5.30}$$

and the fact that molar fractions and saturations add to one.

$$S_l + S_g = 1$$
 (5.31)

$$x_{\rm H_2O,l} + x_{\rm CO_2,l} + x_{\rm Ca,l} + x_{\rm CO_3,l} = 1$$
(5.32)

$$x_{\rm H_2O,g} + x_{\rm CO_2,g} = 1$$
 (5.33)

Molar fraction $x_{ca,g}$ and $x_{co_3,g}$ are always zero, because Ca^{+2} and CO_3^{-2} are not present in the gas phase. When one phase is not present, molar fractions of that phase lose their meaning. Fugacity conditions 5.29 and 5.30 are then meaningless too, as well as the molar fraction condition associated with that phase, 5.32 or 5.33.

The Darcy velocity $\vec{v_j}$ is defined as

$$\vec{v}_j = -k \frac{k_{rj}}{\mu_j} \left(\nabla p - \tilde{\rho}_j g \nabla h \right), \quad j \in \{l, g\}.$$
(5.34)

The mass density depends on the pressure

$$\tilde{\rho}_j = \rho_{j,0}(1 + C_j(p - p_0)), \qquad j \in \{l, g, m\}$$
(5.35)

and the molar density also on molar fractions.

$$\rho_j = \frac{\tilde{\rho}_j}{\sum_{c=1}^C x_{cj} M_c}, \qquad j \in \{l.g\}$$
(5.36)

The porosity is

$$\phi = 1 - \frac{M_m}{\tilde{\rho}_m} c_m. \tag{5.37}$$

The intrinsic permeability is

$$k = k_0 \left(\frac{\phi}{\phi_0}\right)^3 \tag{5.38}$$

and relative permeability is

$$k_{rj} = S_j^2, \qquad j \in \{l, g\}.$$
 (5.39)

If the reaction rate is kinetic, we can use formulas 5.34 - 5.39 and 5.28 to directly evaluate terms and be left with four PDEs 5.22 - 5.25, one ODE 5.39, and five algebraic equations 5.29 - 5.33 for ten unknowns p, S_l , S_g , $x_{H_20,l}$, $x_{H_20,g}$, $x_{C0_2,l}$, $x_{C0_2,g}$, $x_{Ca,l}$, $x_{C0_3,l}$, and c_m . If the reaction is in equilibrium, the formula for the reaction rate 5.28 is no longer valid. R is eleventh unknown and the equation 5.27 is added to the system.

Constraints 5.29 – 5.33 can be directly substituted to the system to simplify it (most notably $S_l + S_g = 1$), but not all at once. When the liquid or the gas phase disappears, some constraint has to be relaxed, otherwise certain states are unreachable. For example, the state $S_l = 1$, $x_{H_{20},l} = 0.5$, $x_{CO_2,l} = 0$, which is the initial condition. Our approach is always to enforce (and directly substitute) the fugacity conditions. When one phase disappears, we relax the condition on the sum of molar fractions of that phase. A more detailed explanation, together with the discussion on the degenerate states, can be found in Section 5.2.

5.1.3 Initial and Boundary Conditions

1D Scenarios

The first two benchmark scenarios are one-dimensional. They have an identical domain, initial conditions, and boundary conditions. The only difference is how the chemical reaction is treated. One scenario uses kinetic reaction rate and the other has chemical equilibrium.

The domain is a strip of length 1000 m, width 1 m, and height 1 m. On the left side (x = 0) carbon dioxide is injected at the constant rate $0.2 \text{ m}^3/\text{day}$. The flux of other components through this boundary is zero. The boundary condition on the right side (x = 1000) is a free outflow for all components at constant pressure p_0 . The top, the bottom, the front, and the back side are insulated. It is treated as a one-dimensional problem and gravity is neglected.



Fig. 5.1: The domain of the 1D scenarios. Carbon dioxide is injected on the left side at rate $0.2 \text{ m}^3/\text{day}$, and the right side is kept at constant pressure allowing free outflow.

Initial conditions can be found in Table 5.2. The initial calcite molar concentration can be calculated from porosity⁵ using the equations 5.2 and 5.3 as

$$c_m = \frac{\rho_{m0}}{M_m} \left(1 - \frac{\phi}{\phi^T} \right).$$
(5.40)

Its initial value is then $c_m = 13987.4 \text{ mol/m}^3$.

Tab. 5.2: Initial conditions.

Variable:	ϕ_0	p_0	S_l	$x_{\mathrm{H_2O},l}$	$x_{\mathrm{co}_2,l}$	$x_{\mathrm{Ca},l}$	$x_{\text{CO}_3,l}$
Initial Value:	0.3	95 bar	1	0.5	0	0.25	0.25

Components Ca^{+2} and CO_3^{-2} have identical initial values, boundary conditions, and transport operators, and the reaction consumes them at the equal rate. Therefore, their concentrations are the same throughout the simulation and we can remove one variable from the system. This symmetry is present in all scenarios we present.

The experiment lasts for 1000 days. During this time the carbon dioxide manages to penetrate just over 620 m.

⁵The calculation of the mineral volume also depends on the pressure to evaluate the mass density (see equation 5.11) but the initial density is given.

2D Scenarios

The third and the fourth scenarios use a domain that is 600 m long, 10 m wide, and 240 m tall. Boundary conditions are similar to previous scenarios. The top, the bottom, the front, and the back side are insulated, and the boundary condition on the right side is free outflow at pressure p_0 too. The injection on the left side (x = 0) is more complicated. We inject $200 \text{ m}^3/\text{day}$ of water through the top half of the boundary ($z \in (120, 240)$), and $1000 \text{ m}^3/\text{day}$ of carbon dioxide through the bottom half of the boundary⁶. The gravity makes the carbon dioxide rise and water sink, so the two injected streams cross each other.

The third scenario uses the same initial conditions as the first two scenarios, with the exception of porosity which is not constant. Most of the domain keeps porosity at 0.3 but there is a channel with porosity of 0.8 in the middle of the domain. The fourth scenario has the same initial porosity, pressure, and saturation. It has more ions, and their molar fractions are close to what we could observe in the real world – they are significantly lower than in other scenarios. Their initial values are summarized in Section 5.5.



Fig. 5.2: Initial porosity and boundary conditions of the 2D scenario. $\Omega_1 = \{(x, y, z) \in \mathbb{R}^3 \mid x \in (120, 480), y \in (0, 10), z \in (80, 160)\}$

The simulated time is the same as in the previous two scenarios: 1000 days. This time the unsaturated region reaches the right boundary. After all, the domain is shorter, has a permeable channel, and the injection rate is higher.

5.2 Numerical Setup

The system is solved in a fully-coupled fashion and the time stepping scheme is backward Euler method – fully implicit. The time step length follows the same

 $^{^{6}}$ The pressure at the left boundary changes throughout the simulation which makes the exact amounts of CO₂ hard to determine.

heuristics as our previous model (see Section 4.4). The maximum time step length is 0.1 day in 1D scenarios to suppress the pressure oscillation. In the 2D scenario, the maximum time step length is 1 day.

The nonlinear solver is the Newton solver, which uses a direct solver UMFPack [Dav04] for linearized systems. The matrix is calculated from the residual via numerical differentiation. The process is parallelized with the additive Schwarz domain decomposition method.

Our choice of primary variables is pressure, liquid saturation, molar fraction of water in the liquid phase, molar fraction of carbon dioxide in the gas phase, molar fraction of calcium ions in the liquid phase, and molar concentration of calcite (in the solid phase). We will mark them $p, S_l, x_{H_2O,l}, x_{CO_2,g}, x_{Ca,l}$, and c_m respectively.

The system has six equations:

$$\sum_{j \in \{l,g\}} \partial_t \left(\phi S_j \rho_j x_{\mathbf{H}_2 \mathbf{O}, j} \right) + \nabla \cdot \left(\vec{v}_j \rho_j x_{\mathbf{H}_2 \mathbf{O}, j} - \phi S_j \rho_j D \nabla x_{\mathbf{H}_2 \mathbf{O}, j} \right) = 0$$
(5.41)

$$\sum_{j \in \{l,g\}} \partial_t \left(\phi S_j \rho_j x_{\text{co}_2, j} \right) + \nabla \cdot \left(\vec{v}_j \rho_j x_{\text{co}_2, j} - \phi S_j \rho_j D \nabla x_{\text{co}_2, j} \right) = 0$$
(5.42)

$$\partial_t \left(\phi S_l \rho_l x_{\mathsf{ca},l} + c_m \right) + \nabla \cdot \left(\vec{v}_l \rho_l x_{\mathsf{ca},l} - \phi S_l \rho_l D \nabla x_{\mathsf{ca},l} \right) = 0 \tag{5.43}$$

$$\partial_t c_m = -A\left(1 - \frac{x_{\operatorname{Ca}}, l x_{\operatorname{CO}_3}, l}{K_{sp}}\right) \quad \text{or} \quad \min\left(c_m, 1 - \frac{x_{\operatorname{Ca}}, l x_{\operatorname{CO}_3}, l}{K_{sp}}\right) = 0 \tag{5.44}$$

$$\min\left(S_l, 1 - \sum_{c=1}^{C} x_{c,l}\right) = 0$$
 (5.45)

$$\min\left(S_g, 1 - \sum_{c=1}^C x_{c,g}\right) = 0.$$
 (5.46)

The first two are mass balance equations for H_2O and CO_2 . The third equation is the sum of the mass balance equations for Ca^{+2} and calcite. This way the reaction term cancels out which allows skipping the extra equation in the chemical equilibrium case where the reaction rate is unknown.

The mass balance equation for CO_3^{-2} is omitted. The initial and boundary conditions of CO_3^{-2} are identical to those of Ca^{+2} , the chemical reaction consumes them at the same rate, and their diffusion coefficient is the same too. Therefore, their concentrations remain the same throughout the whole computation. The variable $x_{\text{co}_3,l}$ is not a primary variable, it is set equal to $x_{\text{ca},l}$.

The fourth equation (eq. 5.44) is different depending on the type of the chemical reaction. If the kinetic reaction rate is used, the reaction rate can be evaluated

directly, and we use the left equation: the mass balance equation for the calcite. If the equilibrium reaction is imposed, we instead use the complementarity constraint⁷.

The last two equations are complementarity constraints managing the phase disappearance. They are related to the fact that we enforce the fugacity conditions 5.29 and 5.30 even when one phase disappears. The condition that the sum of molar fractions of the disappeared phase is one must be relaxed in order to allow for all possible states to be reachable, for example, water-saturated conditions with no carbon dioxide, or gas-saturated conditions with no water⁸.

The secondary variables are calculated from the primary by application of equations from Section 5.1. In short, the molar fraction $x_{co_3,l}$ is equal to $x_{ca,l}$, and other non-primary molar fractions are calculated through the fugacity conditions. Porosity is calculated from c_m and p. S_g from S_l . Molar densities from the pressure and all molar fractions of the corresponding phase. Darcy velocity \vec{v}_j from the pressure and saturation⁹. The reactive surface area of the mineral A from p and c_m . Notice that the fugacity relation is used even when one phase disappears and molar fractions of that phase lose significance.

5.2.1 Degenerate States

The main challenge of the benchmark is the phase disappearance. Three types of degeneracy are present: the state without the liquid phase, without the gas phase, and without calcite¹⁰. It is possible to have a doubly degenerate state without the calcite and one of the fluid phases.

The interaction between the saturation and molar fractions is shown in Figure 5.3. It depicts the 1D problem with only two components: water and carbon dioxide. Besides having no calcium it is identical to 1D benchmark scenarios. We inject carbon dioxide at x = 0 into the water-saturated domain and capture profiles at $x \in (0, 100)$ at time 150 days.

Water-saturated area (x > 83) has molar fractions identical to the initial condition $(x_{\text{H}_2\text{O},l}, x_{\text{CO}_2,l}, x_{\text{H}_2\text{O},g}, x_{\text{CO}_2,g}) = (1, 0, 0.1, 0)$. The sum of molar fractions of the gas

⁷In the equilibrium case, the reaction rate R is unknown and the system gains the complementarity constraint (the right-side option of the equation 5.44) to constrain it. The mass balance equation for the calcite is just $\partial_t c_m = R$, and it is the only place where R is present after it was eliminated from the equation 5.43. Both R and calcite mass balance equation can be removed from the system and we are left with the complementarity constraint.

⁸The fact that there is no gas phase without carbon dioxide and no liquid phase without water is what determines the sign of the molar fraction conditions in the complementarity constraints.
⁹The gravity term uses the mass density which does not depend on molar fractions.

¹⁰The fact that the total porosity is one means that calcite is the only mineral (reactive or nonreactive) of the solid phase. Having no solid phase is the quirk of the model and is treated as a porous medium with $\phi = 1$. The last benchmark scenario has more realistic chemistry with much smaller molar concentrations and calcite dissolution is minimal at this time scale.



Fig. 5.3: Depiction of the interaction between the saturation and molar fractions during the phase appearance and disappearance. This is the benchmark scenario 1 with no calcium ions, i.e. carbon dioxide is injected from the left into the water-saturated sample. Captured at time 150 days and zoomed to the first 100 meters of the domain.

phase does not add to one, but the complementarity constraint for the gas phase 5.46 is fulfilled because $S_g = 0$. Before the gas phase can appear, the molar fractions need to change, which is happening at $x \in (74, 83)$. At the breaking point where the gas phase appears, both parts of the complementarity constraint for the gas phase 5.46 are equal to zero. In the unsaturated area, all four molar fractions are constant because they are constrained by two fugacity relations $(x_{\rm H_20,l} = 10 x_{\rm H_20,g})$ and $x_{\rm co_2,l} = 0.1 x_{\rm co_2,g}$ and $x_{\rm co_2,l} = 0.1 x_{\rm co_2,l} = 0$ and $1 - x_{\rm H_20,g} - x_{\rm co_2,g} = 0$). When the water saturation is zero (at x < 3), the complementarity constraint for the liquid phase 5.45 is fulfilled even when $1 - x_{\rm H_20,l} - x_{\rm co_2,l} > 0$, which allows $x_{\rm H_20,l}$ to decrease and $x_{\rm co_2,l}$ to increase.

The following subsections discuss degeneracies caused by the phase disappearance specific to our implementation of equations. Another approach would still have to deal with the degeneracy but particular issues and solutions might differ. The main factor influencing the behavior of our implementation is our choice of the primary variables and how the fugacity relation is enforced even when one phase disappears and only the condition on the sum of molar fractions is relaxed.

Liquid-saturated State

The state without the gas phase is the easiest degenerate state to handle. Both components of the gas phase are present in the liquid phase too, so their mass balance equations stay well defined.

The complementarity constraint for the gas phase 5.46 ensures that $1 - x_{\text{H}_2\text{O},g} - x_{\text{CO}_2,g} \ge 0$. The positive value of the condition is reached when the carbon dioxide content is low. Such is the state of the initial condition, the domain is water-saturated with no carbon dioxide.

It is not possible to create the gas phase without the carbon dioxide. The fugacity ensures that $x_{\text{H}_2\text{O},g} = 0.1 x_{\text{H}_2\text{O},l} \leq 0.1$ and the sum of molar fractions must be 1 when the phase is present. The appearance of the gas phase must be preceded by the increase of the carbon dioxide molar fraction in the liquid phase up to the point when the sum of the molar fractions of the gas phase becomes 1 and that branch of the complementarity constraint reaches the equality. At this point, the other branch of the complementarity condition can become greater and the gas phase emerges.

Gas-saturated State

The state without the liquid phase is the hardest one to handle. The components that are also present in the gas phase pose no problems. However, two components, namely Ca^{+2} and CO_3^{-2} , are present only in the liquid phase.

Nevertheless, in the particular setting of the benchmark the system stays wellposed. CO_3^{-2} is present in the same amount as Ca^{+2} , so $x_{\text{co}_3,l}$ can be substituted by $x_{\text{ca},l}$. Almost all terms from their mass balance equations depend on saturation and vanish when the liquid phase disappears. The only remaining term is the reaction term. Since we added the calcite mass balance equation to the equation 5.43, what remains is $\partial_t c_m = 0$. If we had mass balance equations for both Ca^{+2} and CO_3^{-2} , they would be linearly dependent in a gas-saturated state and the system would be ill-posed.

In the gas-saturated state, the mass balance equation for Ca^{+2} no longer contains $x_{Ca,l}$, only c_m . In the kinetic case, the molar fraction $x_{Ca,l}$ is coupled to the mass balance equation for c_m , and in the chemical equilibrium it is coupled to the complementarity condition. In both cases the calcite concentration can not change $(\partial_t c_m = 0)$, so the molar fraction is forced to be at equilibrium, where $x_{Ca,l} = 0.25$. However, one minor regularization was necessary. The equation 5.44 had to be adjusted to

$$\partial_t c_m = -A\left(1 - \frac{x_{\operatorname{Ca}}, l|x_{\operatorname{Ca}}, l|}{K}\right) \quad \text{or} \quad \min\left(c_m, 1 - \frac{x_{\operatorname{Ca}}, l|x_{\operatorname{Ca}}, l|}{K}\right) = 0.$$
(5.47)

110 Chapter 5 Second Model: Multicomponent Multiphase Flow with Phase Disappearance

Setting $x_{\text{CO}_3,l}$ equal to $x_{\text{Ca},l}$ means the reaction rate contains square of $x_{\text{Ca},l}$. When the liquid saturation is zero, this square is the only place of the system where $x_{\text{Ca},l}$ appears. The addition of the dependence on the sign of $x_{\text{Ca},l}$ removed the negative solution¹¹.

The situation gets more dire when the calcite concentration becomes zero too. $c_m = 0$ in the equation 5.47 makes A = 0 and satisfies the complementarity constraint for any $x_{\text{H}_2\text{O},l} \leq 0.25$. Then $x_{\text{H}_2\text{O},l}$ is not present in the system at all. Such a doubly degenerate situation does not happen in any benchmark scenario, although the 2D case with chemical equilibrium comes close. Our implementation is not able to handle this situation.

The benchmark scenario with extended chemistry has more components in the liquid phase. When the liquid phase disappears the system becomes ill-posed. All our attempts at regularization failed, so we have no results for this benchmark scenario.

No calcite

This degeneracy affects only the chemical reaction. In the kinetic case, the reaction rate can be still directly evaluated. However, the dependence on the mineral saturation means that if there is no calcite the reaction rate is zero. This state is an unstable equilibrium. Zero reaction rate means, that the calcite can not precipitate regardless of the amounts of Ca^{+2} and CO_3^{-2} . On the other hand, any small precipitation leads to a positive reaction rate and more precipitation. In practice, the reaction slows as calcite dissolves and its concentration never actually reaches zero.

In the equilibrium case, the reaction rate no longer depends on the calcite concentration and the concentration can easily become zero. The complementarity condition 5.44 ensures that it does not become negative and allows the pore solution to become undersaturated by letting molar fractions of Ca^{+2} and CO_3^{-2} be less than 0.25.

As mentioned before, the gas-saturated state with no calcite is a doubly degenerate state that our model can not handle. The scenario with extended chemistry uses realistic concentrations of ions and does not even come close to dissolving all calcite.

5.2.2 Alternative Ways of Handling Degeneracies

Our approach can easily handle the liquid-saturated state, and the state with no calcite. The gas-saturated state is problematic because calcium is present solely in

¹¹This addition was not proactive, it was implemented after the negative solution was produced. Convergence problems were not observed, just a random degree of freedom flipped its sign.

the liquid phase and its molar fraction becomes undefined when the liquid phase disappears. It remains algebraically defined through the reaction term, but that works only when the calcite concentration is positive.

An alternative way to manage the degeneracy is to use persistent variables. Instead of the molar fraction, the primary variable represents the total mass or amount of moles of the component. In our notation, such variable would be $c_c = \phi \sum_{j \in \{l,g\}} S_j \rho_j x_{cj}$, although ϕ could be omitted. It remains well-defined even when either saturation is zero. However, the degeneracy is still present. Computing the molar fraction becomes ill-defined when the saturation approaches zero. Often the persistent variables are used for all components. Then, in order to obtain the distribution of components across phases (i.e. molar fractions), we have to solve a system of equations. This is called the flash computation and is done locally each time molar fractions in some control volume are needed.

Another approach is variable switching. When one quantity becomes undefined another one is picked to be represented by the primary variable so the system is well-posed. This approach is not dissimilar to ours. After all, using complementarity constraints is equivalent to the primal-dual active set strategy [HIK02]. Instead of branching in the type of the variable, we have branching inside the complementarity constraint.

Regularizing the system is also a possibility. This involves changing the equations or coefficients a little, so technically it is solving a different problem. Our attempts with the regularization did not lead to better results, quite the opposite – the point when the system could not converge even with the minimal time step size was reached earlier. To mention a few regularization attempts, we first tried to prevent the disappearance of the calcium molar fraction variable from the system by bounding the liquid saturation away from zero in the mass term of the calcium mass balance equation. However, dabbling in the delicate mass balance around the zero liquid saturation worsened the convergence.

Another attempt was to use a nonzero calcium fugacity coefficient and add some Ca^{+2} to the gas phase. The equation would then retain some terms in the gassaturated state like water and carbon dioxide¹². As in the previous case, the resulting scheme failed sooner, not later, in our simulation.

¹²This regularization was suggested by Etienne Ahusborde, who participated in the benchmark in the $DuMu^x$ team. For them, this regularization worked but the fugacity coefficient could not be as small as desired, otherwise the regularization was too ill-conditioned.

5.2.3 Finite Volume Discretization

Keeping the previous equation numbering, the strong formulation is

$$\sum_{j \in \{l,g\}} \partial_t \left(\phi S_j \rho_j x_{\mathbf{H}_2 \mathbf{0}, j} \right) + \nabla \cdot \left(\vec{v}_j \rho_j x_{\mathbf{H}_2 \mathbf{0}, j} - \phi S_j \rho_j D \nabla x_{\mathbf{H}_2 \mathbf{0}, j} \right) = 0 \quad (5.41)$$

$$\sum_{j \in \{l,g\}} \partial_t \left(\phi S_j \rho_j x_{\text{co}_2,j} \right) + \nabla \cdot \left(\vec{v}_j \rho_j x_{\text{co}_2,j} - \phi S_j \rho_j D \nabla x_{\text{co}_2,j} \right) = 0 \quad (5.42)$$

$$\partial_t \left(\phi S_l \rho_l x_{\mathsf{ca},l} + c_m \right) + \nabla \cdot \left(\vec{v}_l \rho_l x_{\mathsf{ca},l} - \phi S_l \rho_l D \nabla x_{\mathsf{ca},l} \right) = 0 \quad (5.43)$$

$$\partial_t c_m = -A\left(1 - \frac{x_{\operatorname{Ca}}, lx_{\operatorname{CO}_3}, l}{K_{sp}}\right) \quad \text{or} \quad \min\left(c_m, 1 - \frac{x_{\operatorname{Ca}}, lx_{\operatorname{CO}_3}, l}{K_{sp}}\right) = 0 \quad (5.44)$$

$$\min\left(S_l, 1 - \sum_{c=1}^{C} x_{c,l}\right) = 0 \quad (5.45)$$

$$\min\left(S_g, 1 - \sum_{c=1}^{n} x_{c,g}\right) = 0. \quad (5.46)$$

To get the variational formulation and consequently finite volume discretization, we multiply equations with test functions and use Green's formula to transfer the divergence to the test function. Choosing test functions to be piecewise constant on the mesh \mathcal{T} and plugging basis functions of this space one by one into the time-discretized system grants us the system of algebraic equations. The discretization details can be found in Section 3.1.

Let \mathcal{F}_i be the set of inner faces of mesh elements

$$\forall f \in \mathcal{F}_i \quad \exists \tau_1, \tau_2 \in \mathcal{T} : \quad f = \overline{\tau}_1 \cap \overline{\tau}_2,$$

and \mathcal{F}_b be the set of boundary faces of mesh elements

$$\forall f \in \mathcal{F}_b \quad \exists \tau \in \mathcal{T} : \quad f = \overline{\tau} \cap \overline{\Omega}.$$

We further split \mathcal{F}_b into four parts based on the type of the boundary condition.

$$\mathcal{F}_b = \mathcal{F}_{bo} \cup \mathcal{F}_{bn} \cup \mathcal{F}_{bw} \cup \mathcal{F}_{bq}$$

 \mathcal{F}_{bo} – Free outflow

- \mathcal{F}_{bn} Insulated, homogeneous Neumann condition
- \mathcal{F}_{bw} Water injection
- \mathcal{F}_{bq} Carbon dioxide injection

Functions are discontinuous over the inner faces. Recalling the notation from Section 3.1, we denote the inner and outer element relative to the inner face $f \in \mathcal{F}_i$ as τ_- and τ_+ .

When we integrate over the surface of some element τ , the unit outer normal vector points outwards. Accordingly, the outer element τ_+ is on the side of the face f to which the unit outer normal vector $\vec{\nu}_f$ points, and the inner element τ_- lies in the opposite direction of the normal vector.

We use this notation also for traces of functions. For example, s_- is the trace of the function s from the inside element. There is no outer element for boundary faces, so for them only τ_- and s_- are defined. For convenience, we extend this notation to boundary faces and use s_+ to mean the boundary value if the Dirichlet condition is used. The last notation we recall are symbols for the harmonic mean $\{\cdot\}_h$, arithmetic mean $\{\cdot\}_a$, and upwind $\{\cdot\}_u$.

Finally, the residual formulation of the system is

$$\begin{split} r(u,w) &= \sum_{\tau \in \mathcal{T}} \sum_{j \in \{l,g\}} \partial_{t} \left(\phi S_{j} \rho_{j} x_{\mathrm{H}_{2}0,j} \right) w_{1} |\tau| \\ &+ \sum_{f \in \mathcal{F}_{b}} \sum_{j \in \{l,g\}} \left(\vec{v}_{j} \vec{v}_{f} \{ \rho_{j} x_{\mathrm{H}_{2}0,j} \}_{u} - \{ \phi S_{j} D \}_{h} \{ \rho_{j} \}_{u} \nabla x_{\mathrm{H}_{2}0,j} \vec{v}_{f} \right) [w_{1-} - w_{1+}] |f| \\ &+ \sum_{f \in \mathcal{F}_{b}} \sum_{j \in \{l,g\}} \left(\vec{v}_{j} \vec{v}_{f} \{ \rho_{j} x_{\mathrm{H}_{2}0,j} \}_{u} \right) w_{1-} |f| + \sum_{f \in \mathcal{F}_{bw}} q_{\mathrm{H}_{2}0} w_{1-} |f| \\ &+ \sum_{\tau \in \mathcal{T}} \sum_{j \in \{l,g\}} \partial_{t} \left(\phi S_{j} \rho_{j} x_{\mathrm{co}_{2},j} \right) w_{2} |\tau| \\ &+ \sum_{f \in \mathcal{F}_{b}} \sum_{j \in \{l,g\}} \left(\vec{v}_{j} \vec{v}_{f} \{ \rho_{j} x_{\mathrm{co}_{2},j} \}_{u} - \{ \phi S_{j} D \}_{h} \{ \rho_{j} \}_{u} \nabla x_{\mathrm{co}_{2},j} \vec{v}_{f} \right) [w_{2-} - w_{2+}] |f| \\ &+ \sum_{f \in \mathcal{F}_{b}} \sum_{j \in \{l,g\}} \left(\vec{v}_{j} \vec{v}_{f} \{ \rho_{j} x_{\mathrm{co}_{2},j} \}_{u} \right) w_{2-} |f| + \sum_{f \in \mathcal{F}_{b}} q_{\mathrm{co}_{2}} w_{2-} |f| \\ &+ \sum_{f \in \mathcal{F}_{b}} \partial_{t} \left(\phi S_{l} \rho_{l} x_{\mathrm{ca},l} + c_{m} \right) w_{3} |\tau| \\ &+ \sum_{\tau \in \mathcal{T}} \partial_{t} \left(\phi S_{l} \rho_{l} x_{\mathrm{ca},l} \}_{u} - \{ \phi S_{l} D \}_{h} \{ \rho_{l} \}_{u} \nabla x_{\mathrm{ca},l} \vec{v}_{f} \} \right) [w_{3-} - w_{3+}] |f| \\ &+ \sum_{f \in \mathcal{F}_{b}} \left(\vec{v}_{l} \vec{v}_{f} \{ \rho_{l} x_{\mathrm{ca},l} \}_{u} w_{3-} |f| \\ &+ \sum_{f \in \mathcal{F}_{b}} \left(\vec{v}_{l} \vec{v}_{f} \{ \rho_{l} x_{\mathrm{ca},l} \}_{u} w_{3-} |f| \right) \\ &+ \sum_{f \in \mathcal{F}_{b}} \min \left(C_{m}, 1 - \frac{x_{\mathrm{ca},l} x_{\mathrm{co}_{3},l}}{K_{sp}} \right) w_{4} |\tau| \right] \\ &+ \sum_{\tau \in \mathcal{T}} \min \left(S_{l}, 1 - \sum_{c=1}^{C} x_{c,l} \right) w_{5} |\tau| \\ &+ \sum_{\tau \in \mathcal{T}} \min \left(S_{g}, 1 - \sum_{c=1}^{C} x_{c,g} \right) w_{6} |\tau| \end{aligned}$$
(5.48)

114 Chapter 5 Second Model: Multicomponent Multiphase Flow with Phase Disappearance

with the normal flux evaluated as

$$\vec{v}_j \vec{\nu}_f = -\{k\}_h \frac{\{k_{rj}\}_u}{\mu_j} \left(\frac{p_+ - p_-}{\|x_+ - x_-\|} - \{\tilde{\rho}_j\}_a g \nabla h \cdot \vec{\nu}_f\right)$$
(5.49)

and the partial derivation in time as

$$\partial_t s(t)\Big|_{t=t_n} = \frac{s(t_n) - s(t_{n-1})}{t_n - t_{n-1}}.$$
 (5.50)

All spatial terms are evaluated at time $t = t_n$, i.e., we use backward Euler discretization in time.

Since our grid is rectangular and axiparallel, $\nabla h \cdot \vec{\nu}_f$ is either 0, 1, or -1. On the free outflow boundary ($f \in \mathcal{F}_{bo}$) the pressure p_+ is set to the outflow pressure p_0 . Coefficients are no longer averaged but use values from the inner element (as if we used upwind instead of harmonic mean). x_- and x_+ are centers of the inner and the outer element, or, in case the face lies at the boundary, x_+ is the face's center.

The algebraic system is then obtained by setting test functions $w_i, i \in \{1, 2, \dots 6\}$ to

$$w_i = \begin{cases} 1, & x \in \tau, \\ 0, & x \in \Omega \setminus \tau \end{cases}$$

one by one while keeping other test functions zero for all mesh elements $\tau \in \mathcal{T}$. The size of the system is $6|\mathcal{T}|$.

The matrix of the system has columns $\partial r(u, w)/\partial u_j$. Rows correspond to the ordering of the test functions w. We order degrees of freedom in what DUNE calls *entity blocked* fashion. The system is block-wise 3-diagonal in a one-dimensional setting, and 5-diagonal in a two-dimensional setting. The blocks are of size 6×6 with the patterns depicted in Figure 5.4.

,					,	,						,	Filled when:
$(\times$	×	×	\times	l	$\times)$	()	<	\times	\times	\times	l	$\times)$	\times always
X	\times	\times	\times	l	\times	>	<	\times	\times	\times	l	×	. never
l	×	l	l	l	×		ļ	\times	l	l	l	l	k kinotia ango
k				×	×								
	ī	1	1	1									<i>i</i> liquid phase present
	-	U	ı	U									g gas phase present
(.	g	g	g	•	• /		•	•	·	·	•	•)	\overline{l} gas-saturated
	4:-	~~~	-1 h 1	a a1-			~f	с		1 1	1.	_1-	\bar{q} liquid-saturated
	diagonal block off-diagonal block												



As we can see, the system is tightly coupled. In the two-dimensional case, matrix rows can have up to 30 entries¹³. The system is significantly reduced in the gassaturated state due to the majority of terms with $x_{\text{Ca},l}$ disappearing. The calcite mass balance equation is the only row depending on $x_{\text{Ca},l}$, at least as long as $c_m > 0$. (This is the unsolved doubly-degenerate state.) In the gas-saturated state, we thus have three equations that determine three variables. Complementarity constraint for the liquid phase determines S_l , and calcium and calcite mass balance equations determine $x_{\text{Ca},l}$ with c_m . There is dependence on the saturation of neighboring block(s) in the calcium mass balance equation, but if those control volumes are gas-saturated too, the complementarity constraint will tie that degree of freedom to itself. The remaining three equations (mass balance of water and carbon dioxide plus the complementarity condition of the gas phase) are determining p, $x_{\text{H}_2\text{O},l}$, and $x_{\text{CO}_2,g}$.

5.2.4 Regularization

Our discretization contains nearly no regularization. Formulas are programmed as they are written in equations in Section 5.1, but some had to be regularized for values that would not be reachable in the real world. Otherwise, unphysical solutions were possible.

Many changes are natural, after all, following the formula blindly outside the domain of its validity is futile. Here is the list:

- Saturation is bounded to interval [0, 1] for the calculation of relative permeability and diffusion.
- Pressure can not be lower than zero bars for the calculation of mass density.
- Porosity is at minimum zero and at maximum one.
- Chemical equilibrium condition and analogical term in the kinetic reaction rate was adjusted to $1 \frac{x_{Ca}, l | x_{Ca}, l |}{K_{sn}}$.
- the denominator in the harmonic average calculation is at minimum 10^{-30} .

All of these deviations from blindly following formulas are necessary. Unsurprisingly, the biggest problem was saturation. By design, it attains its extreme values, and we often see it becoming negative but within a numerical error margin close to zero¹⁴. However, bounding it strongly from below by the zero (for example by em-

¹³Upwinding somewhat reduces this number though the worst-case scenario is possible if the liquid flows into some inner cell from all directions.

 $^{^{14}}$ In the 2D scenario with gravity the lowest and highest values of the liquid saturation are $-2.5\cdot10^{-36}$ and $1+1.4\cdot10^{-11}$. The lowest values in 1D scenarios are $-1.9\cdot10^{-34}$ in kinetic and $-2.5\cdot10^{-32}$ in equilibrium case, and their highest are 1 and 1.

ploying the projected line search introduced in Section 3.4.1) leads to convergence issues.

5.3 One-dimensional Scenarios Results

The first two scenarios are fairly similar as they differ only in the rate of the chemical reaction. However, chemical reactions play a little role in these scenarios, the calcite concentration (tied to porosity) barely changes. Chemical reactions happen in two places, both are close to where a phase appears or disappears. The first place is where the carbon dioxide molar fraction increases from zero. This reduces molar fractions of other components, and decreased calcium concentration leads to calcite dissolution which translates into porosity increase. The second place is where pure carbon dioxide gas pushes into unsaturated area¹⁵. Carbon dioxide entering the unsaturated region forces the water to evaporate so that molar fractions stay balanced. The evaporation reduces water molar fraction in the liquid phase and other fractions are then increased. The increase in the calcium concentration leads to calcite precipitation and, subsequently, the decrease in porosity.

The difference between the kinetic and the equilibrium case is therefore slight, the graphs 5.5 - 5.8 differ only in the calcium concentration. In the chemical equilibrium case, the calcium concentration is kept constant – at the equilibrium point. In the kinetic reaction case, the reaction happens only when the calcium concentration is out of equilibrium, so it is lower where the gas phase appears, and higher where the liquid phase disappears. The latter is barely noticeable though. We will see bigger differences in the 2D scenarios where the chemical reactions are stronger.



Fig. 5.5: Concentrations at 10 days. Kinetic and equilibrium case.

Figures 5.5 – 5.8 show the profiles of S_l , $x_{H_2O,l}$, $x_{CO_2,g}$, $x_{Ca,l}$, and porosity at 10, 100, 500, and 1000 days. To highlight the self-similarity of the graphs, we show only

¹⁵The zone with no liquid phase and some water vapor is so thin it is practically nonexistent. The gassaturated region with pure carbon dioxide is right next to the unsaturated area.



Fig. 5.6: Concentrations at 100 days. Kinetic and equilibrium case.



Fig. 5.7: Concentrations at 500 days. Kinetic and equilibrium case.



Fig. 5.8: Concentrations at 1000 days. Kinetic and equilibrium case.



Fig. 5.9: Concentrations at 1000 days. Kinetic and equilibrium case. Zoomed to $x \in (614, 624)$.

118 Chapter 5 Second Model: Multicomponent Multiphase Flow with Phase Disappearance



Fig. 5.10: Pressure at several times. Kinetic and equilibrium case.



Fig. 5.11: Pressure on the left boundary over time. Kinetic and equilibrium case.

the left part of the domain of the length proportional to the elapsed time. The linear dependence of the front position on the time comes from the constant injection rate.

The self-similarity applies to the front position and the saturation. The width of the front stays constant, over time the front becomes sharper relative to the rest of the curve. Figure 5.9 shows the values at the final time for x from 614 m to 624 m. The cut-out was chosen such that the front is in a similar position to the graph at the time t = 10 days.



Fig. 5.12: Pressure on the left boundary over time on meshes with increasing refinement level. Kinetic case.

The only unknown that is not self-similar is the pressure¹⁶. As Figure 5.10 shows, pressure is decreasing to 95 bars on the right boundary. The constant injection of carbon dioxide first builds up the pressure (notice that the pressure at 10 days is slightly lower) until the flow speed matches the injection rate. The gradient is constant in saturated regions and is steeper in the liquid-saturated part than in the unsaturated and gas-saturated part because the gas hydraulic conductivity is higher¹⁷.

Figure 5.11 shows the evolution in time of the pressure on the left boundary. The wiggling effect is heavily dependent on the mesh size. Presumably, the culprit is the permeability calculation. The pressure adjusts to keep the flow constant. The dependence of the pressure oscillation on the mesh size is depicted in Figure 5.12. It shows the pressure evolution of the cell on the left boundary during the first 100 days on meshes with 1000, 2000, and 4000 cells. Finer mesh has higher frequency because the front crosses more cells in the same time frame. The maximum time step size is 0.01 days for all three meshes¹⁸.

5.3.1 Numerical Convergence

Table 5.3 and 5.4 show the strong scaling results for computation with kinetic and equilibrium chemistry on a mesh with 2000 cells. The results using equilibrium chemistry follow the same pattern with marginally higher values, so we decided not to recreate the whole table. All computations took a similar number of time steps and Newton iterations – ranging from 10007 to 10052 and 31239 to 31427 respectively. The Newton solver does not handle parallelization and is affected only indirectly. The parallelization is handled by the linear solver and we list the number of solves it performed, the time spent on assembling the matrix, the time spent solving linear systems depending on the number of subdomains, and the overlap width. The assemble time and the linear solve time do not add up to the total time. The remaining time was spent on creating outputs, and on the defect evaluation in Newton solver¹⁹.

Increasing the number of subdomains increases the number of linear solves. In the sequential case, that number is equal to the number of Newton iterations. In parallel cases, it is necessary to solve subdomain problems (with updated overlap)

¹⁶The pressure also affects the density, and consequently porosity and saturation. However, this effect is negligible.

¹⁷The permeability is the same, the difference comes from the viscosity.

¹⁸Using longer time steps does not affect the amplitude. But it reduces the sampling rate and graphs become discordant.

¹⁹One frame per time step is created. Newton evaluates the initial defect plus one defect per iteration to check the convergence.

Subdomains	Overlap	Lin. solves	Assemble t.	Lin. solve t.	Total time
1	—	31239	627	1100	1993
2	2	73492	344	1416	1925
2	4	75026	336	1261	1755
2	8	75523	327	1117	1587
2	16	75780	319	1066	1521
2	32	80453	333	946	1415
2	64	83990	329	987	1454
4	2	218347	164	1955	2220
4	4	199931	164	2009	2288
4	8	197548	166	1935	2197
4	16	198593	169	1939	2208
4	32	198652	172	1725	2000
4	64	189054	176	1060	1326
4	128	159644	197	1060	1359
8	2	602503	86	2587	2730
8	4	482487	88	2193	2336
8	8	426602	88	2047	2191
8	16	388079	89	1971	2122
8	32	351588	94	1817	1974
8	64	285918	104	1496	1670
8	120	212423	122	1371	1579

Tab. 5.3: Strong scaling results of the kinetic case on the mesh with 2000 cells. Time is in seconds.

Tab. 5.4: Strong scaling results of the equilibrium case on the mesh with 2000 cells. Time is in seconds.

Subdomains	Overlap	Lin. solves	Assemble t.	Lin. solve t.	Total time
1	_	31252	625	1097	1997
2	2	73627	343	1409	1921
2	32	80631	324	941	1400
4	4	199685	165	2115	2379
4	64	188773	184	1079	1353
8	2	586767	87	2696	2838
8	64	285590	104	1519	1694

several times until convergence is reached. Increasing the number of subdomains increases this number, and increasing the overlap width decreases it, with the exception of 2 subdomains which is puzzling.

The matrix is assembled at the beginning of each nonlinear iteration. It is thus not affected by the number of linear solves. More subdomains mean smaller matrices and shorter assembly time. Increasing the overlap enlarges the matrix and lengthens the time²⁰.

The time spent solving the linearized problems is the dominant part. After some point, increasing the number of subdomains leads to a longer total time despite the computation using more cores. We attribute the inefficiency of the domain decomposition approach to the lack of load balancing. The most challenging part is to resolve the interface between the unsaturated and the gas- or the liquid-saturated region. This area is small and moving, the workload can not be evenly split among the processes.

5.4 Two-dimensional Scenario Results

The two-dimensional scenario is much more challenging than one-dimensional scenarios. Besides the extra dimension, this scenario features nonhomogeneous initial porosity and calcite concentration, more complicated boundary conditions, and gravity. Moreover, the conditions lead to the doubly degenerate state with zero liquid saturation and calcite concentration. The kinetic reaction rate does not consume the calcite completely, the zero concentration is approached asymptotically. The problem does get progressively harder but our approach is sufficient for the time span required by the benchmark. If the benchmark used equilibrium chemistry, the calcite would be depleted completely. The computation would then fail when the calcite-less cell is about to become fully gas-saturated, which is, depending on the mesh refinement²¹, at the latest at day 550.

Let's recall the problem setting. Figure 5.13 shows the initial porosity and boundary conditions of the setup. We have one highly porous channel in the middle of the domain, water is injected on the top half and carbon dioxide on the bottom half of the left boundary. The top and the bottom boundary are insulated, and liquids can outflow freely through the right boundary.

 $^{^{20}}$ Again, except 2 subdomains. There the number of time steps is 10007 and the number of Newton iterations *increases* with the overlap width from 31299 to 31312. Times can vary up to 10\% between identical runs but repeated computations confirmed the trend.

²¹Finer meshes reach this point sooner. Smaller control volumes have sharper interfaces and more extreme values.



Fig. 5.13: Initial porosity and boundary conditions of the 2D scenarios.



(a) Pressure at 100 days.

The dominant flow direction is from left to right due to the injection of water and carbon dioxide. The flow is faster in the center of the domain due to the highly permeable channel. Gravity induces weaker flow in the vertical direction because the gas phase formed by the injected carbon dioxide has a lesser density than water. Gas rises, so injected water and carbon dioxide streams cross each other.

Figures 5.14 (a)-(f) show values of primary variables at 100 days, with porosity instead of calcite concentration. Injected fluids did not get far, most of the domain is still liquid-saturated with initial molar fractions. Only the pressure changed everywhere, it is highest at the injection side and the gravity makes it higher on the bottom too. The permeable channel can be traced on the pressure graph too.

Injected water dilutes the pore solution, which triggers the calcite dissolution to replenish calcium. Porosity is increased. The calcite is not dissolved completely but it is close. At day 100, the maximum porosity is 0.97. The porosity increase is more pronounced at the bottom part of the water injection zone, as the flow is stronger due to the gravity and the vicinity of the highly porous channel.

The injection of carbon dioxide affects the molar fractions less. After all, the molar fraction of the carbon dioxide in the liquid phase is less than 0.1. In the unsaturated



(b) Liquid saturation at 100 days.



(c) Carbon dioxide molar fraction in the gas phase at 100 days.



(d) Water molar fraction in the liquid phase at 100 days.



(e) Calcium molar fraction in the liquid phase at 100 days.



(f) Porosity at 100 days.





(a) Pressure at 500 days.



(b) Liquid saturation at 500 days.

region, we observe a small increase in porosity by 0.01 and almost constant molar fractions in ranges $x_{\text{H}_2\text{O},l} \in (0.40404, 0.404041)$, $x_{\text{CO}_2,l} \in (0.09597, 0.09598)$, and $x_{\text{Ca},l} \in (0.24994, 0.25)$. The gas-saturated region is already present at day 100. The transition from unsaturated state to gas-saturated is accompanied by a slight change in molar fractions and porosity decreases back to 0.3. This is what we observed in the 1D scenario with kinetic reaction rate.

At 500 days, the pressure profile looks similar to the profile at 100 days with two important differences. The maximum pressure decreased to 160 bar and the left side of the permeable channel is no longer outlined in the pressure graph. Both can be attributed (at least partially) to the dissolution of the calcite and the creation of an even more permeable channel with $\phi \approx 1$ spanning from the left boundary to the channel.

Injected water washes the calcium away and a leached channel forms. Very little calcite is left. The minimum molar concentration is $6.3 \cdot 10^{-4}$ mol/m³ (down from 14474.6, which makes it less than $5 \cdot 10^{-10}$ %). The leached channel connects to the wider permeable channel in the middle where the injected water spreads.



(c) Carbon dioxide molar fraction in the gas phase at $500~{\rm days}.$



(d) Water molar fraction in the liquid phase at 500 days.



(e) Calcium molar fraction in the liquid phase at $500~\mathrm{days}.$



(f) Porosity at 500 days.

Fig. 5.15: Graphs of primary variables and porosity at 500 days.

Injected carbon dioxide spreads mainly in two directions: It is pushed right and floats up. After entering the permeable region, it spreads faster along its upper edge. Carbon dioxide diffuses from the permeable channel alongside that edge and, since it can not spread as quickly in the less permeable region, it accumulates there. This is the origin of the gas phase strip above the permeable channel. Some carbon dioxide rises straight up. It crosses the water stream and the leached channel it created. Water injected from the left prevents it from getting closer to the boundary.



(a) Pressure at 1000 days.

At 1000 days, the maximum pressure is down to 140 bar but qualitatively nothing changed compared to 500 days. The porosity continues the trend from the previous graph – the leached channel is wider, and the permeable channel got more leached too. The minimum calcite molar concentration is down to $1.9634 \cdot 10^{-11}$.

The basis of the fan structure above the channel is already present at the day 500 and it steadily grows throughout the simulation. The more porous lines are at the edges of smaller flow streams of carbon dioxide.



(**b**) Liquid saturation at 1000 days.



(c) Carbon dioxide molar fraction in the gas phase at $1\,000$ days.



(d) Water molar fraction in the liquid phase at 1000 days.



(e) Calcium molar fraction in the liquid phase at $1\,000$ days.



(f) Calcite molar concentration at 1000 days.



(g) Porosity at 1000 days.

Fig. 5.16: Graphs of primary variables and porosity at 1000 days. This time we also included calcite molar concentration which is proportional to $1 - \phi$.

The major stream of carbon dioxide that went upward connected with the boundary and spread in both directions. On the right, the upward stream connected with the other mass of carbon dioxide which diffused slowly upwards from the permeable channel. The connection left a liquid-saturated bubble between the two major streams of carbon dioxide.

The upward stream is also reaching the left boundary. This is one of two critical places in the domain because the fluid rich in carbon dioxide invades a calcite-depleted region. The conditions are very close to the doubly-degenerate state. Liquid saturation is $9.94704 \cdot 10^{-5}$, and calcite molar concentration is $2.11487 \cdot 10^{-11}$.

The second critical place is close to the left boundary at the interface between streams of injected water and carbon dioxide. Although it does not appear to lead to double degeneracy, changes in the properties of neighboring cells dominated by different flows are stark. It is the interface between the gas-saturated and the unsaturated region. It contains the least porous cells in the domain ($\phi = 0.21$) right next to cells with $\phi \approx 1$. The gas-saturated region has no water vapor ($x_{\text{H}_2\text{O},g} = 0$) even though the main and the only water stream is right next to it ($x_{\text{H}_2\text{O},g} = 1/11$). Other variables do not change so rapidly. Carbon dioxide is fixed in the unsaturated part to $x_{\text{CO}_2,g} = 10/11$ so the jump is not big. Moreover, calcium is washed away in the unsaturated region and not present in the gas-saturated region.

5.4.1 Parallel Scaling

Table 5.5 shows the number of linear iterations and computation times of solving the problem on the mesh with 60×24 cells²². The number of subdomains corresponds to the number of processes.

The initial time step size is 0.01 days. All computations on this mesh quickly reach the maximal time step length of one day and keep it till the end. The number of time steps for runs in Table 5.5 is 1028 with three exceptions. The sequential run has 1036, the run on two subdomains with overlap 6 has 1029, and the run on 8 subdomains with overlap 4 has 1021.

The total number of Newton iterations is close too. Sequential uses 3936, it needs one linear solve per iteration. Parallel cases range from 4103 to 4142. Table 5.5 shows the dependence of the number of linear solves, assemble time, and linear solve time on the number of subdomains and overlap. They follow the same pattern as in 1D scenarios. Note that this mesh is much coarser than the 1D one and the relative overlap width is larger.

²²This is the mesh size originally proposed by the benchmark, but it is quite coarse. Shown graphs use once-refined mesh, which is the mesh the benchmark ended up using.

Subdomains	Overlap	Lin. solves	Assemble t.	Lin. solve t.	Total time
1	_	3936	99.8	404.2	560
2	1	37117	54.6	491.5	570
2	2	23829	57.2	401.4	484
2	4	17627	59.3	374.1	458
2	6	15436	64.5	375.6	466
4	1	72047	28.2	377.1	420
4	2	43595	30.0	318.5	365
4	4	27538	33.3	319.9	371
4	6	21620	37.4	357.0	415
8	1	106740	15.6	249.8	276
8	2	72239	18.1	237.2	267
8	4	46445	22.5	267.9	303

Tab. 5.5: Additive Schwarz parallelization results. Overlap is in the number of cells and time in seconds. The mesh has 60×24 cells.

Tab. 5.6: Additive Schwarz parallelization results. Mesh size is 120×48 cells. Wide overlap is necessary, and it does not outperform sequential code.

Subdomains	Overlap	Time steps	Assemble t.	Lin. solve t.	Total time
1	—	2376	1177	7350	9449
2	8	4634	1769	14433	17172
2	12	3928	1332	11075	13216
4	12	4446	845	8831	10357

Table 5.6 tells the story of the once-refined mesh. We decided to replace the field with the number of linear solves for the number of time steps because the differences in the number of time steps make the number of iterations inconclusive. The overlap is given in the number of cells so the same width is achieved with twice as many cells as in Table 5.5.

This table is much smaller because not all combinations of the number of subdomains and the overlap width were able to finish. The computation on two subdomains with the overlap of one, two, four, and (surprisingly) 16 cells gets stuck at 938 days – exactly at the same spot. Four subdomains meet the critical point sooner. With the overlap of two cells, it stops at 470 days, and with the overlap of eight cells at 832 days. Data from these computations did not give us any proper leads to the cause. After all, the values of all variables are within the error margin from the sequential case.

The major problem with refining is not even the quadrupling of the number of degrees of freedom but the fact that function graphs are getting sharper. Specifically, the stream of carbon dioxide that rises upwards and spreads to the sides after it
hits the upper boundary. It keeps to the top row of cells as it is less dense. Since the refined mesh has smaller cells that need less carbon dioxide to become gassaturated, the gas spreads sideways faster and reaches the area drained of calcite close to the top left boundary sooner. The refined problem is reaching the doubly degenerate situation faster. Porosity will never reach zero with kinetic reaction rate but the problem becomes badly conditioned. Our method is unable to finish the computation at twice refined mesh, it stops at 389.57 days.

5.5 Extended Chemistry

The benchmark features one more scenario with more complex chemistry. Three additional chemical agents are used: H^+ , OH^- , and HCO_3^- . The system has four chemical reactions instead of one.

$$\begin{split} H_2 \mathsf{O} &\leftrightarrows \mathsf{H}^+ + \mathsf{O}\mathsf{H}^- \\ H_2 \mathsf{O} + \mathsf{CO}_2 &\leftrightarrows \mathsf{H}^+ + \mathsf{HCO}_3^- \\ \mathsf{HCO}_3^- &\leftrightarrows \mathsf{H}^+ + \mathsf{CO}_3^{-2} \\ \mathsf{CaCO}_3 + \mathsf{H}^+ &\leftrightarrows \mathsf{Ca}^{+2} + \mathsf{HCO}_3^- \end{split} \tag{5.51}$$

The first three are upheld via chemical equilibrium and the calcite reaction is kinetic. The addition of H^+ ions makes the calcite reaction now depend on the pH of the solution. Moreover, the concentration of CO_2 is coupled to CO_3^{-2} . The injection of carbon dioxide has a significant impact on the chemical balance.

Molar fractions are closer to real values, which means they are much lower. Their initial values are listed in Table 5.17.

Chemical equilibrium reactions are

$K_1 a_{{\rm H}_2{\rm O},l} = a_{{\rm H},l} a_{{\rm OH},l}$	(5.52)
$K_2 a_{\rm H_2O,l} a_{\rm CO_2,l} = a_{\rm H,l} a_{\rm HCO_3,l}$	(5.53)
$K_3 a_{\mathrm{HCO}_3,l} = a_{\mathrm{H},l} a_{\mathrm{CO}_3,l}$	(5.54)

2	$x_{\mathrm{H_2O},l}$	$1 - 5.888 \cdot 10^{-6}$
	$x_{\text{CO}_2,l}$	$3.9624 \cdot 10^{-10}$
	$x_{\mathrm{Ca},l}$	$2.1703 \cdot 10^{-6}$
	$x_{\text{co}_3,l}$	$6.2315 \cdot 10^{-7}$
	$x_{\mathrm{H},l}$	$2.3507 \cdot 10^{-12}$
	$x_{\mathrm{OH},l}$	$1.5475 \cdot 10^{-6}$
	$x_{\text{HCO}_3,l}$	$1.5467 \cdot 10^{-6}$

and the changed kinetic reaction for the calcite dissolution is

$$\partial_t c_m = (1 - \phi_0) \hat{s}_m \left(1 - \frac{a_{\text{Ca},l} a_{\text{HCO}_3,l}}{K_{sp} a_{\text{H},l}} \right).$$
 (5.55)

Fig. 5.17: Initial molar fractions. The equilibrium constants are listed in Table 5.7²³. Chemical equilibria are expressed in terms of chemical activities $a_{c,l}$. The chemical activity of water is one, i.e. $a_{H_2O,l} =$ 1, and others are given by the formula

$$a_{c,l} = \frac{x_{c,l}}{x_{H_{2}O,l}} \frac{1}{M_{H_{2}O}}, \quad c \in \{CO_2, Ca^{+2}, H^+, OH^-, CO_3^{-2}, HCO_3^-\}.$$
 (5.56)

Our approach is not able to finish the computation of this system. Additional species present only in the liquid phase make this system ill-posed when an area becomes gassaturated. (We are not saved by some chemical reaction as in the previous scenario.) The double degeneracy is not a problem in this scenario. Smaller concentrations of chemical agents lead to less calcite dissolution and 1000 days is not enough to completely dissolve it.

Tab. 5.7: Equilibrium constants.

K_1	$1.122 \cdot 10^{-14}$
K_2	$5.093 \cdot 10^{-7}$
K_3	$5.260 \cdot 10^{-11}$
K_{sp}	$1.2618 \cdot 10^{-2}$

5.6 Summary

We successfully solved the first three benchmark scenarios. Our results are in good agreement with other teams [Ahu+24]. We could not solve the extended chemistry case, albeit many other teams did not fare better. Only two teams submitted the extended chemistry problem.

Our approach is unique in using complementarity constraints for handling the phase disappearance. It proved to be a viable strategy, although it seems to be very sensitive to regularizations, especially those touching the mass balance equations. The main advantage of this approach is the minimal regularization. The only deviation of the numerical implementation from the finite volume discretization²⁴ guarantees that the denominator in the calculation of the harmonic average is at least 10^{-30} . Most importantly, the phase disappearance degeneracy is handled purely with complementarity conditions. There is no transition region between unsaturated and liquid- or gas-saturated state.

The main disadvantage of our approach is the difficulty in extending it. We were unable to implement the last benchmark scenario, and could not resolve the doublydegenerate state. All our attempts to regularize the system led to convergence issues. Those attempts include:

²³Benchmark lists decadic logarithms of them because together with initial conditions they are calculated from chemical potential using Gibbs free energy relation.

²⁴Besides functions operating on the intended range of primary variables, e.g., relative permeability using $\min(\max(S_l, 0), 1)$ instead of plain S_l .

- Using small nonzero calcium fugacity to have some calcium in the gas phase. It makes the calcium mass balance equation depend on the calcium molar fraction even in the gas-saturated state.
- Preventing the mass term in the calcium mass balance equation from vanishing by bounding the saturation in that term from zero.
- Changing the calcium mass balance equation in the gas-saturated state into something that defines calcium molar fraction. For example into $x_{\text{Ca},l} = 0$ or $\partial_t x_{\text{Ca},l} = 0$. Smoothly transitioning between the equations (based on saturation) did not work either.

The convergence issues were encountered already when the gas phase was disappearing, long before the doubly-degenerate state was reached.

Conclusion

6

The centerpiece of this work is two strongly nonlinear problems. Both feature twophase flow in a porous medium, transport of contaminants, and chemical reactions that affect the porosity. In both, the main complication is a strong local nonlinearity although their nature differs. The problems are solved numerically using a finite volume scheme.

The first model describes the accelerated carbonation of concrete in the laboratory setting [BS04]. Accelerated conditions with 50% carbon dioxide concentration in air make the system reaction-dominant. The carbonation front is always one mesh cell wide, so refining the mesh makes the profiles of calcium and carbon dioxide concentration sharper. The progression of the carbonation front was met with convergence issues when the calcium concentration was approaching zero. We overcame the issues through the addition of the projection step into the nonlinear solver's line search method.

Furthermore, the first model served to compare the performance of two nonlinear solvers: Newton and RASPEN [Dol+16]. The Newton solver proved to be faster, but RASPEN was able to take longer time steps and even complete the variations of the hardest test case in which the Newton solver failed.

The second model, conceptually more complex, encompasses a collection of the SiTraM benchmark problems [Hoo+24]. Here, the main challenge is to manage the phase appearance and disappearance. Our tool for managing the branching between saturated and unsaturated states was complementarity constraints. Unfortunately, generalizing this approach to the last benchmark scenario proved difficult.

Future Work

There are several avenues one could expand upon our work, and each presents a unique challenge. However, predicting what tools are the best for a new problem is hard. Nonlinearities call for an individual approach. For example, the projected line search method, so crucial in the first model, was useless in the second model even though it has more bounded variables. Projecting the saturation was even detrimental, despite the saturation being already bounded by complementarity constraints. The most obvious avenue is finishing the last scenario of the SiTraM benchmark. Combining complementarity constraints with another regularization technique proved difficult, and our model needed to use too small time steps. We see the biggest strength of complementarity constraints in keeping the branching on the algebraic level, simplifying the discretization and implementation. They could widen the possibilities of generating the code from a Unified Form Language representation.

Increasing the number of contaminants is not the only option for extending our models. One could use a more complicated porous medium with inhomogeneities or fractures, or add features like hysteresis, spreading of cracks, water uptake by roots, etc. However, this would change the model's behavior so much that it could hardly be called its extension.

Yet another avenue is increasing the mesh size and looking into parallel scaling. Our problems were calculated on relatively small meshes. This favors direct linear solvers over iterative solvers, but there was another reason to use the direct solver. We used the direct solver because iterative solvers could not converge. Getting an iterative solver to work would require an amazing preconditioner. Several preconditioners available in DUNE PDELab were tried (Jacobi, Gauss-Seidel, SSOR, and ILU0) but neither was good enough. The direct solver was used only for the subdomain problems, so the first concern could be the load balancing because we deal with problems with the moving front (carbonation front or the interface between saturated and unsaturated zone). Moving to bigger meshes would also increase the importance of the coarse correction. Increasing the number of subdomains comes hand in hand with the rise in the number of communications between subdomains. The first model has shown that some parts of the problem can be affected more strongly than others, but we did not reach the point where it would be significant.

Finally, solving inverse problems could be the easiest way to extend our model. The objective of an inverse problem is finding model parameters that best fit given measurements, and possibly also quantifying the uncertainty of such findings. The apparatus for solving inverse problems is usually independent of the problem and can be applied to a whole suite of models. The direct model can be treated as a black box, so this route poses a different challenge.

Bibliography

- [Ade92] F. Adenot. "Durabilité des Bétons: Caractérisation et Modélisation des Processus Physiques et Chimiques de Dégradation du Ciment". PhD thesis. University of Orléans, 1992 (cit. on p. 64).
- [Ahu+24] E. Ahusborde, B. Amaziane, S. de Hoop, et al. "A Benchmark Study on Reactive Two-Phase Flow in Porous Media: Part II - Results and Discussion". In: *Computational Geosciences* (2024). DOI: 10.1007/s10596-024-10269-y (cit. on pp. 8, 134).
- [Ama13] I. Amato. "Green Cement: Concrete Solutions". In: *Nature* 494 (2013). DOI: 10.1038/494300a (cit. on p. 2).
- [Ang+17] U. M. Angst, M. R. Geiker, A. Michel, et al. "The steel-concrete interface". In: *Materials and Structures* 50 (2017) (cit. on pp. 13, 18).
- [Asp24] J. Aspdin. "An Improvement in the Modes of Producing an Artificial Stone". In: *Patent no. BP 5022* (1824) (cit. on p. 1).
- [Aur+15] M. Auroy, S. Poyet, P. Le Bescop, et al. "Impact of Carbonation on Unsaturated Water Transport Properties of Cement-Based Materials". In: Cement and Concrete Research 74 (2015) (cit. on pp. 4, 14).
- [BH12] M. R. Bangi and T. Horiguchi. "Effect of Fibre Type and Geometry on Maximum Pore Pressures in Fibre-reinforced High Strength Concrete at Elevated Temperatures". In: *Cement and Concrete Research* 42 (2012) (cit. on p. 26).
- [BS04] B. Bary and A. Sellier. "Coupled Moisture—Carbon Dioxide–Calcium Transfer Model for Carbonation of Concrete". In: *Cement and Concrete Research* 34 (2004) (cit. on pp. 14, 16, 22, 59, 62, 64, 66, 67, 88, 137).
- [Bas18] P. Bastian. Lecture Notes on Parallel Solution of Large Sparse Linear Systems. Interdisziplinäres Zentrum für Wissenschaftliches Rechnen Universität Heidelberg. Apr. 2018 (cit. on pp. 47, 48).
- [BB08] P. Bastian and M. Blatt. "On the Generic Parallelisation of Iterative Solvers for the Finite Element Method". In: *Int. J. Computational Science and Engineering* 4(1) (2008), pp. 56–69 (cit. on p. 56).
- [Bas+08a] P. Bastian, M. Blatt, A. Dedner, et al. "A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part I: Abstract Framework". In: *Computing* 82(2-3) (2008), pp. 103–119 (cit. on p. 56).
- [Bas+08b] P. Bastian, M. Blatt, A. Dedner, et al. "Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part II: Implementation and Tests in DUNE". In: *Computing* 82(2-3) (2008), pp. 121–138 (cit. on p. 56).

- [Bas+21] P. Bastian, M. Blatt, A. Dedner, et al. "The DUNE framework: Basic concepts and recent developments". In: *Computers & Mathematics with Applications* 81 (2021). DUNE project website: https://dune-project.org/, pp. 75–112 (cit. on pp. 7, 31, 56).
- [BN72] Z. P. Bažant and L. J. Najjar. "Nonlinear Water Diffusion in Nonsaturated Concrete". In: *Matériaux et Constructions* 5 (1972) (cit. on p. 64).
- [BC10] J. Bear and A. H.-D. Cheng. Modeling Groundwater Flow and Contaminant Transport. Vol. 23. ISBN: 978-1-4020-6681-8. Springer Netherlands, 2010 (cit. on pp. 9, 14, 16, 19, 62).
- [BR18] L. Berlyand and V. Rybalko. *Getting Acquainted with Homogenization and Multiscale*. ISBN: 978-3-030-01776-7. Birkhäuser, 2018.
- [Ber82] D. P. Bertsekas. "Projected Newton Methods for Optimization Problems with Simple Constraints". In: SIAM Journal on Control and Optimization (1982).
 DOI: 10.1137/0320018 (cit. on pp. 50, 51).
- [BB07] M. Blatt and P. Bastian. "The Iterative Solver Template Library". In: Lecture Notes in Scientific Computing 4699 (2007). In B. Kåström, E. Elmroth, J. Dongarra and J. Wasniewski, Applied Parallel Computing. State of the Art in Scientific Computing, pp. 666–675 (cit. on p. 56).
- [Blu01] M. J. Blunt. "Flow in porous media pore-network models and multiphase flow". In: Elsevier Science, Current Opinion in Colloid & Interface Science 6 (2001) (cit. on p. 9).
- [Bor+18] M. Borregales, F. A. Radu, K. Kumar, and J. M. Nordbotten. "Robust Iterative Schemes for Non-linear Poromechanics". In: *Computational Geosciences* (2018) (cit. on p. 95).
- [Bru+12] F. Brue, C. A. Davy, F. Skoczylas, N. Burlion, and X. Bourbon. "Effect of Temperature on the Water Retention Properties of two High Performance Concretes". In: Cement and Concrete Research (2012) (cit. on p. 15).
- [BDR12] H. Bruining, M. Darwish, and A. Rijnks. "Computation of the Longitudinal and Transverse Dispersion Coefficient in an Adsorbing Porous Medium Using Homogenization". In: *Transport in Porous Media* 91 (2012) (cit. on p. 9).
- [CEF97] C. Carde, G. Escadeillas, and R. François. "Use of Ammonium Nitrate Solution to Simulate and Accelerate the Leaching of Cement Pastes due to Deionized Water". In: *Magazine of Concrete Research* (1997) (cit. on p. 4).
- [CKK10] J. Carrayrou, M. Kern, and P. Knabner. "Reactive Transport Benchmark of MoMaS". In: Computational Geosciences 14 (2010). DOI: 10.1007/s10596-009-9157-7 (cit. on p. 6).
- [Cas+09] M. Castellote, L. Fernandez, C. Andrade, and C. Alonso. "Chemical Changes and Phase Analysis of OPC Pastes Carbonated at Different CO₂ Concentrations". In: *Materials and Structures* (2009) (cit. on p. 4).

- [CC06] C.-F. Chang and J.-W. Chen. "The Experimental Investigation of Concrete Carbonation Depth". In: *Cement and Concrete Research* 36 (2006) (cit. on p. 22).
- [CGQ19] T. Chen, X. Gao, and L. Qin. "Mathematical Modeling of Accelerated Carbonation Curing of Portland Cement Paste at Early Age". In: *Cement and Concrete Research* 120 (2019).
- [Chi+97] A. V. Chicagov, D. A. Varlamov, R. A. Dilanyan, et al. "The Local and WWWadapted Versions of the Software System on Crystal Structures of Minerals (MINCRYST-97)". In: 7 (1997). http://database.iem.ac.ru/mincryst/ (cit. on p. 23).
- [Col10] F. Collins. "Inclusion of Carbonation During the Life Cycle of Built and Recycled Concrete: Influence on Their Carbon Footprint". In: *Int J Life Cycle Assess* 15 (2010). DOI: 10.1007/s11367-010-0191-4 (cit. on p. 2).
- [CB02] M. S. Constanza-Robinson and M. L. Brusseau. "Gas Phase Advection and Dispersion in Unsaturated Porous Media". In: *Water Resources Research* 38 (2002) (cit. on pp. 19, 20).
- [CW15] L. Czarnecki and P. Woyciechowski. "Modelling of Concrete Carbonation; is it a Process Unlimited in Time and Restricted in Space?" In: *Bulletin of the Polish Academy of Sciences, Technical Sciences* 63 (2015).
- [Dav04] T. A. Davis. "Algorithm 832: UMFPACK V4.3 An Unsymmetric-Pattern Multifrontal Method". In: AMC Transactions on Mathematical Software 30 (2004) (cit. on p. 107).
- [Del07]J. M. P. Q. Delgado. "Longitudinal and Transverse Dispersion in Porous Media".In: Chemical Engineering Research and Design (2007). DOI: 10.1205/cherd07017.
- [Dia00] S. Diamond. "Mercury Porosimetry; An Inappropriate Method for the Measurement of Pore Size Distributions in Cement-based Materials". In: Cement and Concrete Research 30 (2000) (cit. on pp. 4, 10).
- [DiC13] D. A. DiCarlo. "Stability of Gravity-Driven Multiphase Flow in Porous Media: 40 Years of Advancements". In: *Water Resources Research* 49 (2013). DOI: 10.1002/wrcr.20359 (cit. on p. 84).
- [Dol+16] V. Dolean, M. J. Gander, W. Kheriji, F. Kwok, and R. Masson. "Nonlinear Preconditioning: How to Use a Nonlinear Schwarz Method to Precondition Newton's Method". In: *SIAM Journal on Scientific Computing* 35 (6 2016) (cit. on p. 137).
- [Dui+08] C. J. van Duijn, A. Mikelić, I. S. Pop, and C. Rosier. "Effective Dispersion Equations For Reactive Flows With Dominant Peclet and Damkohler Numbers". In: Advances in Chemical Engineering 34 (2008).
- [Eks01] T. Ekström. "Leaching of Concrete: Experiments and Modelling". PhD thesis. Division of Building Materials, LTH, Lund University, 2001 (cit. on pp. 4, 11, 22, 24).

[Fle+11] B. Flemisch, M. Darcis, K. Erbertseder, et al. "DuMux: DUNE for Multiphase, component, scale, physics, ... Flow and Transport in Porous Media". In: Advances in Water Resources 34 (2011). New Computational Methods and Software Tools, DOI: 10.1016/j.advwatres.2011.03.007 (cit. on p. 5). [FB12] M.V.A. Florea and H.J.H. Brouwers. "Chloride Binding Related to Hydration Products Part I: Ordinary Portland Cement". In: Cement and Concrete Research 42 (2012) (cit. on pp. 23, 26). [FS95] J. Freund and R. Stenberg. "On Weakly Imposed Boundary Conditions for Second Order Problems". In: Proceedings of the International Conference on Finite Elements in Fluids - New Trends and Applications (1995). [Gab+04] E. Gabriel, G. E. Fagg, G. Bosilca, et al. "Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation". In: Proceedings, 11th *European PVM/MPI Users' Group Meeting*. 2004 (cit. on p. 48). [Gen80] M. T. van Genuchten. "A Closed-Form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils". In: Soil Science Society of America Journal 44 (1980) (cit. on pp. 13, 61). [GLS99] W. Gropp, E. Lusk, and A. Skjellum. Using MPI: Portable Parallel Programming with the Message-Passing Interface. 1st ed. MIT Press, 1999. [GG91] H. Gvirtzman and S. M. Gorelick. "Dispersion and Advection in Unsaturated Porous Media Enhanced by Anion Exclusion". In: Letters to Nature 352 (1991) (cit. on p. 42). [HR89] W. Hackbusch and A. Reusken. "Analysis of a Damped Nonlinear Multilevel Method". In: Numerische Mathematik 55, Springer-Verlag (1989) (cit. on pp. 50, 51). [Hal07] C. Hall. "Anomalous Diffusion in Unsaturated Flow: Fact or Fiction?" In: Cement and Concrete Research 37 (2007) (cit. on pp. 4, 13). [Hin10] M. Hintermüller. Semismooth Newton Methods and Applications. Accessed 24.11.2022 at https://www.math.uni-hamburg.de/home/hinze/Psfiles/ Hintermueller OWNotes.pdf. Oberwolfach-Seminar on "Mathematics of PDE-Constrained Optimization". Department of Mathematics Humboldt-University of Berlin, 2010 (cit. on pp. 54, 55). M. Hintermüller, K. Ito, and M. Kunish. "The Primal-Dual Active Set Strategy [HIK02] as a Semismooth Newton Method". In: SIAM Journal on Optimization 13 (2002) (cit. on pp. 55, 112). [HKK17] J. Hoffmann, S. Kräutle, and P. Knabner. "Existence and Uniqueness of a Global Solution for Reactive Transport with Mineral Precipitation-Dissolution and Aquatic Reactions in Porous Media". In: SIAM Journal on Mathematical Analysis 49 (2017).

- [Hoo+24] S. de Hoop, D. Voskov, E. Ahusborde, B. Amaziane, and M. Kern. "A Benchmark Study on Reactive Two-Phase Flow in Porosu Media: Part I - Model Description". In: *Computational Geosciences* (2024). DOI: 10.1007/s10596-024-10268-z (cit. on pp. v, vi, 7, 97, 98, 137).
- [Hro15] P. Hron. "Numerical Simulation of Multi-Phase Multi-Component Reactive Flow in Porous Media". PhD thesis. Heidelberg University, 2015.
- [Hum+02] W. Hummel, U. Berner, E. Curti, F. J. Pearson, and T. Thoenen. Technical report 02-16, Nagra / PSI Chemical Thermodynamic Data Base 01/01. Tech. rep. Nagra (National Cooperative for the Disposal of Radioactive Waste), 2002 (cit. on p. 23).
- [IH08] F. Ilinca and J.-F. Hétu. "A New Stabilized Finite Element Method for Reactiondiffusion Problems: The Source-stabilized Petrov-Galerkin Method". In: *International Journal for Numerical Methods in Engineering* 75 (2008) (cit. on p. 36).
- [Ipp14] O. Ippisch. "Contributions to the Large-scale Simulation of Flow and Transport in Heterogeneous Porous Media". habilitation. Heidelberg University, 2014 (cit. on p. 35).
- [JWG16] U. H. Jakobsen, K. De Weerdt, and M. R. Geiker. "Elemental Zonation in Marine Concrete". In: *Cement and Concrete Research* 85 (2016) (cit. on pp. 3, 22).
- [JN11] V. John and J. Novo. "Error Analysis of the SUPG Finite Element Discretization of Evolutionary Convection-Diffusion-Reaction Equations". In: *SIAM J. Numer. Anal.* 49 (2011) (cit. on p. 36).
- [JS08] V. John and E. Schmeyer. "Finite Element Methods for Time-dependent Convection-diffusion-reaction Equations with Small Diffusion". In: *Computer Methods in Applied Mechanics and Engineering* 198 (2008) (cit. on p. 37).
- [Kam22] C. Kamthorncharoen. "Two-level Restricted Additive Schwarz Preconditioned Exact Newton with Applications". dissertation. Heidelberg University, 2022 (cit. on pp. 52, 57, 93).
- [KMK03] J. F. Kanney, C. T. Miller, and C. T. Kelley. "Convergence of Iterative Split-Operator Approaches for Approximating Nonlinear Reactive Transport Problems". In: Advances in Water Resources 26 (2003).
- [Kei+21] E. Keilegavlen, R. Berge, A. Fumagalli, et al. "PorePy: an Open-Source Software for Simulation of Multiphysics Processes in Fractured Porous Media". In: *Computational Geosciences* 25 (2021). DOI: 10.1007/s10596-020-10002-5 (cit. on p. 5).
- [Kel99] C. T. Kelley. "Iterative Methods for Optimization". In: Society for Industrial and Applied Mathematics (1999).

[KT06]	A. Korpa and R. Tretting. "The Influence of Different Drying Methods on Cement Paste Microstructures as Reflected by Gas Adsorption: Comparison Between Freeze-drying (F-drying), D-drying, P-drying and Oven-drying Meth- ods". In: <i>Cement and Concrete Research</i> 36 (2006) (cit. on p. 11).
[Krä08]	S. Kräutle. "General Multi-Species Reactive Transport Problems in Porous Media: Efficient NUmerical Approaches and Existence of Global Solutions". habilitation. University of Erlagen-Nuremberg, 2008 (cit. on p. 26).
[Kre84]	P. C. Kreijger. "The Skin of Concrete Composition and Properties". In: <i>Matériaux et Construction</i> 17 (1984) (cit. on pp. 13, 18).
[KNR07]	M. Kutílek, D. R. Nielsen, and K. Reichardt. <i>Soil Water Retention Curve, Interpretation.</i> ICTP Trieste College on Soil Physics, lecture notes https: //indico.ictp.it/event/a06222/material/4/49.pdf. accessed on 20.5.2024. 2007 (cit. on p. 15).
[Laf+15]	B. Lafuente, R. T. Downs, H. Yang, and N. Stone. "The Power of Databases: the RRUF Project". In: <i>Highlights in Mineralogical Crystallography, T. Armbruster, R. M. Danisi</i> . 2015, pp. 1–30 (cit. on p. 23).
[LD09]	V. Lagneau and J. van Der Lee. "Operator Splitting-based Reactive Transport Models in Strong Feedback of Porosity Change: the Contribution of Analytical Solutions for Accuracy Validation and Estimator Improvement". In: <i>Journal</i> <i>of Contaminant Hydrology</i> 112 (2009). DOI: 10.1016/j.jconhyd.2009.11.005 (cit. on p. 6).
[Lat13]	C. Latrille. "Effect of Water Content on Dispersion of Transferred Solute in Unsaturated Porous Media". In: <i>Procedia Earth and Planetary Science</i> 7 (2013) (cit. on p. 20).
[LK18]	A. A. Lukassen and M. Kiehl. "Operator Splitting for Chemical Reaction Systems with Fast Chemistry". In: <i>Journal of Computational and Applied Mathematics</i> 344 (2018).
[MK14]	J. Malek and M. Kaouther. "Destructive and Non-destructive Testing of Concrete Structures". In: <i>Journal of Civil Engineering</i> 8 (2014) (cit. on p. 3).
[Mar+17]	J. J. Martínez, T. Le Borgne, H. Tabuteau, and Y. Méheust. "Impact of Sat- uration on Dispersion and Mixing in Porous Media: Photobleaching Pulse Injection Experiments and Shear-enhanced Mixing Model". In: <i>Water Re-</i> <i>sources Research</i> 53 (2017). DOI: 10.1002/ 2016WR019849 (cit. on p. 20).
[Mat+00]	V. Matte, M. Moranville, F. Adenot, C. Richet, and J. M. Torrenti. "Simulated Microstructure and Transport Properties of Ultra-high Performance Cement- based Materials". In: <i>Cement and Concrete Research</i> 30 (2000).
[MR10]	W. Merz and P. Rybka. "Strong Solutions to the Richards Equation in the Unsaturated Zone," in: <i>Journal of Mathematical Analysis and Applications</i> (2010).

- [NMJ02] G. Nützmann, S. Maciejewski, and K. Joswig. "Estimation of Water Saturation Dependence of Dispersion in Unsaturated Porous Media: Experiments and Modelling Analysis". In: Advances in Water Resources 25 (2002) (cit. on p. 20).
- [PNM18] M. Pap, S. G. Nehme, and A. Mahler. "Analysis and Finite Element Modelling of Water Flow in Concrete". In: *Periodica Polytechnica Civil Engineering* (2018) (cit. on p. 15).
- [PW15] D. L. Parkhurst and L. Wissmeier. "PhreeqcRM: A Reaction Module for Transport Simulators Based on the Geochemical Model PHREEQC". In: Advances in Water Resources 83 (2015) (cit. on p. 5).
- [PFB05] H. Q. Pham, D. G. Fredlund, and S. L. Barbour. "A Study of Hysteresis Models for Soil-Water Characteristic Curves". In: *Canadian Geotechnical Journal* 42 (2005). DOI: 10.1139/t05-071 (cit. on p. 15).
- [PLM18] F. de Prenter, C. Lehrenfeld, and A. Massing. "A Note on the Stability Parameter in Nitsche's Method for Unfitted Boundary Value Problems". In: Computers and Mathematics with Appplications 75 (2018).
- [RS14] A. Rätz and B. Schweizer. "Hysteresis Models and Gravity Fingering in Porous Media". In: ZAMM Journal of Applied Mathematics and Mechanics: Zeitschrift für angewandte Mathematik und Mechanik 94 (2014). DOI: 10.1002/zamm.201200052 (cit. on p. 84).
- [Rou+16] N. Roussel, A. Gram, M. Cremonesi, et al. "Numerical Solutions of Concrete Flow: A Benchmark Comparison". In: *Cement and Concrete Research* 79 (2016). DOI: 10.1016/j.cemconres.2015.09.022 (cit. on p. 5).
- [SMB00] E. Samson, J. Marchand, and J. J. Beaudoin. "Modeling the Influence of Chemical Reactions on the Mechanisms of Ionic Transport in Porous Materials. An Overview". In: Cement and Concrete Research 30 (2000).
- [San20] O. Sander. DUNE The Distributed and Unified Numerics Environment. ISBN 978-3-030-59701-6. Springer, 2020 (cit. on pp. 5, 56).
- [ŠL16] B. Šavija and M. Luković. "Carbonation of Cement Paste: Understanding, Challenges, and Opportunities". In: *Construction and Building Materials* 117 (2016) (cit. on pp. 25, 64).
- [Sch+16] P. Schiller, M. Wahab, T. Bier, S. Waida, and H.-J. Mögel. "Sorption Hysteresis of Hardened Cement Pastes". In: *Journal of Materials Science and Chemical Engineering* 4 (2016) (cit. on pp. 4, 15).
- [SS18] A. Schlichting and C. Seis. "Analysis of the Implicit Upwind Finite Volume Scheme with Rough Coefficients". In: *Numerische Mathematik* 139 (2018) (cit. on p. 36).
- [SZF05] J. W. P. Schmelzer, E. D. Zanotto, and V. M. Fokin. "Pressure Dependence of Viscosity". In: *The Journal of Chemical Physics* 122 (2005) (cit. on p. 63).
- [SMJ12] S. Schneider, D. Mallants, and D. Jacques. "Determining Hydraulic Properties of Concrete and Mortar by Inverse Modelling". In: *Material Research Society* (2012). DOI: 10.1557/opl.2012.601 (cit. on p. 67).

- [Sch17] B. Schweizer. "Hysteresis in Porous Media: Modelling and Analysis". In: Interfaces and Free Boundaries 19 (2017). DOI: 10.4171/IFB/388 (cit. on p. 14).
- [Sey+23] L. M. Seymour, J. Maragh, P. Sabatini, et al. "Hot Mixing: Mechanistic Insights into the Durabiility of Ancient Roman Concrete". In: Science Advances 9 (2023) (cit. on p. 1).
- [SW85] D. Shi and D. N. Winslow. "Contact Angle and Damage During Mercury Intrusion into Cement Paste". In: *Cement and Concrete Research* 15 (1985) (cit. on p. 4).
- [Sim+05] M. De Simoni, J. Carrera, X. Sánchez-Vila, and A. Guadagnini. "A Procedure for the Solution of Multicomponent Reactive Transport Problems". In: Water Resources Research 41 (2005).
- [SCK13] K. Sisomphon, O. Copuroglu, and E.A.B. Koenders. "Effect of Exposure Conditions on Self Healing Behavior of Strain Hardening Cementitious Composites Incorporating Various Cementitious Materials". In: Construction and Building Materials 42 (2013) (cit. on pp. 23, 25).
- [SYU15] C. I. Steefel, S. B. Yabusaki, and K. Ulrich Mayer. "Reactive Transport Benchmarks for Subsurface Environmental Simulation". In: *Computational Geosciences* 19 (2015). DOI: 10.1007/s10596-015-9499-2 (cit. on p. 6).
- [Tay97] H. F. W. Taylor. *Cement Chemistry*. 2nd ed. ISBN 978-0-7277-2592-9. Thomas Telford Publishing, 1997 (cit. on pp. 1, 21, 22, 24).
- [TB12] M. Thomas and T. Bremner. "Performance of Lightweight Aggregate Concrete Containing Slag After 25 Years in a Harsh Marine Environment". In: *Cement and Concrete Research* 42 (2012).
- [Tok+17] T. K. Tokunaga, S. Finsterle, Y. Kim, et al. "Ion Diffusion Within Water Films in Unaturated Porous Media". In: *Environmental Science & Technology* 51 (2017) (cit. on p. 42).
- [Ulb00] M. Ulbrich. "Semismooth Newton Methods for Operator Equations in Function Spaces". In: *SIAM Journal on Opritmization* 13 (2000) (cit. on p. 55).
- [Wan+20] Y. Wang, D. Voskov, M. Khait, and D. Bruhn. "And Efficient Numerical Simulator for Geothermal Simulation: A Benchmark Study". In: *Applied Energy* 264 (2020). DOI: 10.1016/j.apenergy.2020.114693 (cit. on p. 5).
- [Wil50] C. R. Wilke. "A Viscosity Equation of Gas Mixtures". In: *The Journal of Chemical Physics* 18 (1950).
- [WZB12] H. S. Wong, R. W. Zimmerman, and N. R. Buenfeld. "Estimating the Permeability of Cement Pastes and Mortars Using Analysis and Effective Medium Theory". In: Cement and Concrete Research 42 (2012) (cit. on p. 4).
- [Xu18] Mingtian Xu. "A Modified Finite Volume Method for Convection-Diffusion-Reaction Problems". In: *Internationa Journal of Heat and Mass Transfer* 117 (2018) (cit. on p. 36).

[Zho+17] C. Zhou, F. Ren, Z. Wang, W. Chen, and W. Wang. "Why Permeability to Water is Anomalously Lower than that to Many Other Fluids for Cement-Based Material?" In: *Cement and Concrete Research* 100 (2017) (cit. on p. 13).

List of Figures

2.1	Variability of REV properties in relation to element's size	10
2.2	Van Genuchten curves for relative permeability.	13
2.3	Illustration of the ink bottle and snap-off effect	15
2.4	Imbibition and drainage retention curve.	15
3.1	Notation for face's inside and outside cell.	33
3.2	Control volume of vertex-centered finite volumes on rectangular mesh.	38
3.3	Three sets of evaluation points for the dispersion coefficients	40
3.4	Three discretization schemes have the same convergence rate	40
3.5	Subdomains and their overlap	47
3.6	The projection step	51
4.1	Leaching functions.	65
4.2	Initial and boundary conditions	68
4.3	Profiles of calcium molar concentration, carbon dioxide concentration,	
	and porosity at 213 days.	73
4.4	Profiles of calcium molar concentration, carbon dioxide concentration,	
	and porosity at 213 days. Lower CO_2 content.	74
4.5	Profiles of calcium molar concentration, carbon dioxide concentration,	
	and porosity at 232 days. Lower CO_2 content	74
4.6	Comparison of the porosity, calcium molar concentration, and carbon	
	dioxide concentration profiles calculated at 175 days on graphs with	
	512 and 1024 cells	75
4.7	Time evolution of carbon dioxide concentration	76
4.8	Time evolution of calcium molar concentration.	77
4.9	Time evolution of porosity.	77
4.10	Time evolution of air saturation	78
4.11	Time evolution of water pressure	78
4.12	Time evolution of carbon dioxide concentration and porosity. Setting	
	with 11% carbon dioxide concentration.	79
4.13	Saturation on top and bottom boundary on day 175. Test case with	
	gravity	80

4.14	The third test case. Water pressure and air saturation on day 175 81
4.15	The third test case. Calcium molar concentration and carbon dioxide
	concentration on day 175
4.16	The third test case. Porosity on day 175
4.17	The logarithm of the time step size throughout the computation with
	and without the projected line search
4.18	Comparison of time step sizes with unscaled x axis
5.1	The domain of the 1D scenarios
5.2	Initial porosity and boundary conditions of the 2D scenario 106
5.3	Interaction between the saturation and molar fractions during the phase
	appearance and disappearance
5.4	Fill-in of the diagonal and off-diagonal blocks of the matrix 115
5.5	Concentrations at 10 days. Kinetic and equilibrium case
5.6	Concentrations at 100 days. Kinetic and equilibrium case
5.7	Concentrations at 500 days. Kinetic and equilibrium case
5.8	Concentrations at 1000 days. Kinetic and equilibrium case
5.9	Concentrations at 1000 days. Kinetic and equilibrium case. Zoomed to
	$x \in (614, 624)$
5.10	Pressure at several times. Kinetic and equilibrium case
5.11	Pressure on the left boundary over time. Kinetic and equilibrium case. 119
5.12	Pressure on the left boundary over time on meshes with increasing
	refinement level. Kinetic case
5.13	Initial porosity and boundary conditions of the 2D scenarios
5.14	Graphs of primary variables and porosity at 100 days
5.15	Graphs of primary variables and porosity at 500 days
5.16	Graphs of primary variables and porosity at 1000 days
5.17	Initial molar fractions

List of Tables

2.1	Composition of ordinary Portland cement without additives, CEM I	22
2.2	Cement chemistry notation.	22
2.3	Calcium carbonate polymorphs	23
4.1	Nomenclature	60
4.2	Leached minerals at given calcium concentration	66
4.3	Values of constants	67
4.4	Initial conditions.	68
4.5	Dirichlet boundary conditions	68
4.6	Effect of the projection step on Newton and Raspen solver's runtime.	87
4.7	Performance of different solver combinations for flow and contaminant.	88
4.8	Scaling results on the test case without gravity.	91
4.9	Scaling results on the test case with gravity.	92
4.10	Scaling results on the test case with top boundary in contact with air	94
5.1	Values of constants	01
5.2	Initial conditions.	05
5.3	Strong scaling results of the kinetic case	21
5.4	Strong scaling results of the equilibrium case	21
5.5	Additive Schwarz parallelization results	32
5.6	Additive Schwarz parallelization results on refined mesh	32
5.7	Equilibrium constants	34